РАДИО-ЕЖЕГОДНИК



ПУТЕВОДИТЕЛЬ по ARDUINO

ВЫПУСК 34

2015

РАДИО - ЕЖЕГОДНИК 2015 выпуск 34

ТЕМАТИЧЕСКИЙ ОБЗОР ПЕЧАТИ И ИНТЕРНЕТ-РЕСУРСОВ

ТЕМА НОМЕРА:

Путеводитель по *ARDUINO*

Чем больше читаю про Arduino,
тем сильнее хочется попробовать :)
(из комментария на Хабре)

Сборник поможет начинающим сориентироваться в безбрежном море *Arduino* и сделать свои первые конструкции даже не имея навыков программирования. Опытные «ардуинщики» найдут в нём новые идеи и пути для своего дальнейшего совершенствования.

Выпускающий редактор: *С. Степанов*

Над выпуском работали: *С. Муратчаев*, *С. Скворцов*, *В. Смирнов*, *В. Накоряков*, *В. Гололобов*, *С. Глушенко*, *B. Davidov*, *М. Шустов*

Художник: *О. Агафонов*

E-mail: ***radioyearbook@gmail.com***

Январь 2015

Информационная поддержка: *Портал "РадиоЛоцман"* ***www.rlocman.ru***

Официальные версии журнала доступны для свободной загрузки:
www.rlocman.ru/radioyearbook

Путеводитель по ARDUINO

СОДЕРЖАНИЕ

7 **Амперка - официальный дистрибьютор Arduino в России**

8 Готовые наборы от Амперки

9 Arduino Uno

12 Arduino Leonardo

14 Arduino Mega 2560

16 Arduino Due

19 Arduino Yun

22 Arduino Ethernet

23 Arduino Mini

24 Arduino Micro

25 Strela

30 LilyPad Arduino 328

33 STM32 Nucleo F401RE

36 Baby Orangutan B-328

38 Teensy 3.1

40 Netduino 2

42 GPRS Shield v2

43 Motor Shield (2 канала, 2 А)

45 Multiservo Shield

47 Music Shield v2

48 NFC Shield v2

49 Relay Shield (4 канала по 5 А)

51 Wi-Fi Shield

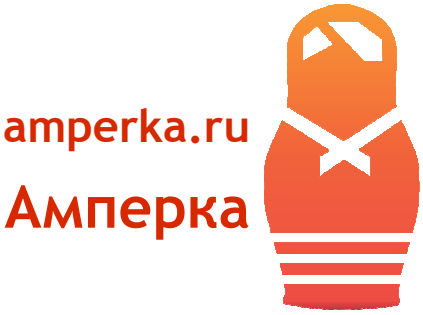
53 XBee

54 Bluetooth Bee

55 Wi-Fi Bee v2

57 CAN-BUS Shield

58 RS-485 Shield



60 **РадиоЛоцман**

61 Как создавали и продвигали Arduino

68 Открытая платформа Arduino высвобождает творческий потенциал

72 Arduino Shields - расширение возможностей аппаратной платформы

76 Аналоговый функционал Arduino: как использовать его в своих проектах

80 Сетевые возможности Arduino-приложений и систем

85 Arduino Due - официальная плата Arduino на микроконтроллере ARM

Статьи, идеи и проекты – интернет-обзор

89 Arduino.ru

92 RoboCraft

94 Все проекты с Arduino

96 Паяльник

98 Хабрахабр

102 KAZUS

103 Digitrode

104 Elektor Labs

106 Hack n Mod

Обзор публикаций по Arduino (с подробным содержанием)

112 Микроконтроллеры DUINO

113 Конструкции с Arduino

114 Знакомство с Arduino

115 Блокнот программиста Arduino

116 Arduino For Dummies / Arduino для «чайников»

119 Arduino Projects For Dummies / Arduino проекты для «чайников»

121 30 Arduino Projects for the Evil Genius / 30 проектов на Arduino для Злого Гения

122 Arduino + Android Projects for the Evil Genius / Arduino + Android проекты для Злого Гения

123 Programming Arduino Next Steps / Следующие шаги в программировании Arduino

125 LEGO and Arduino Projects / LEGO и проекты Arduino

126 Environmental Monitoring with Arduino / Экологический мониторинг с Arduino

127 Atmospheric Monitoring with Arduino / Атмосферный контроль с Arduino

128 Making Things Talk / Устанавливаем связь

130 Arduino Bots and Gadgets / Arduino роботы и гаджеты

131 Make a Mind-Controlled Arduino Robot / Создайте разумного робота на Arduino

132 MintDuino / Создание Arduino-совместимой макетной платы микроконтроллера

133 Make an Arduino–Controlled Robot / Создаём Arduino–управляемого робота

135 Basic Arduino Projects / Простые проекты на Arduino

137 Pro Arduino / Arduino профи

139 Beginning Sensor Networks with Arduino and Raspberry Pi /
Введение в сенсорные сети с Arduino и Raspberry Pi
141 Beginning Android ADK with Arduino / Введение в Android ADK с Arduino
142 Beginning Arduino Programming / Введение в программирование Arduino
144 Arduino Projects to Save the World / Проекты Arduino для сохранения окружающей среды
146 Arduino and LEGO Projects / Проекты Arduino и LEGO
147 Beginning Arduino / Arduino для начинающих
150 Arduino and Kinect Projects / Проекты Arduino и Kinect
152 Arduino Wearables / «Носимый» Arduino
154 Practical Arduino / Arduino на практике
156 Beginning C for Arduino / Знакомство с Си для Arduino
158 Practical Arduino Engineering / Практическое конструирование с Arduino
161 Arduino Adventures / Приключения с Arduino
163 Practical AVR Microcontrollers / Микроконтроллеры AVR на практике
165 Arduino Robotics / Роботы Arduino
168 Arduino Internals / Arduino изнутри
170 Learn Electronics with Arduino / Учитесь электронике с Arduino
172 iOS Sensor Apps with Arduino / Подключение датчиков к устройствам на iOS при помощи Arduino
173 Distributed Network Data / Распределённая сеть передачи данных
174 Building Wireless Sensor Networks / Создание беспроводных сенсорных сетей
176 Beginning NFC / Введение в NFC
178 Arduino Cookbook / Справочник по Arduino
182 Programming Interactivity /
Руководство разработчика по проектированию Arduino и openFrameworks
185 C Programming for Arduino / Программирование на C для Arduino
188 Raspberry Pi Home Automation with Arduino / Raspberry Pi и домашняя автоматизация с Arduino
190 Internet of Things with the Arduino Yun / Интернет-сервисы и Arduino Yun
191 Exploring Arduino / Изучение Arduino: инструменты и методы инженерного искусства
193 Professional Android Open Accessory Programming with Arduino /
Профессиональное программирование на Android Open Accessory для Arduino
195 Arduino Workshop / Мастерская Arduino: практическое знакомство с 65 проектами
199 Arduino in Action / Arduino в действии
202 Arduino Starters Kit Manual / Полное руководство для начинающих в Arduino
203 Programming Your Home / Атоматизируйте с Arduino, Android и компьютером
204 Arduino: A Quick-Start Guide / Arduino: быстрый старт
205 Introduction to Arduino / Введение в Arduino: пара пустяков
206 Arduino Microcontroller Processing for Everyone! / Обработка данных в Arduino для всех!
209 20 Unbelievable Arduino Projects / 20 невероятных проектов на Arduino

Ознакомительные версии книг

211 Программирование микроконтроллерных плат Arduino/Freeduino. Улли Соммер

220 Глава 1. Общие сведения о микроконтроллерах

226 Глава 2. Программирование микроконтроллеров

228 Проекты с использованием контроллера Arduino. Петин В.А.

232 Проекты с использованием контроллера Arduino, 2-е изд. Петин В.А.

245 Предисловие

248 Глава 1. Введение в Arduino

252 Заводим Raspberry Pi. Мэтт Ричардсон и Шон Уоллес

253 Глава 6. Arduino и Raspberry Pi

266 Конспект хакера. Амперка

267 Принципиальные схемы

268 Управление электричеством

269 Широтно-импульсная модуляция

270 Эксперимент 2. Маячок с нарастающей яркостью

273 С чего начинаются роботы? О проекте Arduino для школьников (и не только). Гололобов В. Н.

274 Глава 4. Введение в язык программирования Arduino

284 Глава 5. Arduino, визуальное программирование

294 Глава 6. Введение в язык программирования Scratch

301 Глава 7. Отладка программы на виртуальной плате

Практическое программирование Arduino. Статьи

313 FLProg — система визуального программирования плат Arduino. Сергей Глушенко

332 Портал научно-практических публикаций portalnp.ru Dr. Bob Davidov

333 Создание интерактивных объектов и сред на базе платформы Arduino

358 Многоканальное устройство ввода и накопления аналоговых данных на базе MS Excel

380 Построение интерфейса пользователя локальной системы управления на базе контроллера Arduino UNO

410 Arduino55.ru - интернет магазин электронных компонентов



amperka.ru

Амперка

Официальный дистрибьютор *Arduino* в России

В **Амперке** — только настоящие итальянские платы Arduino и платы расширения Arduino. При покупке каждому покупателю обеспечивается гарантия и поддержка.

А ещё на сайте **amperka.ru** можно найти:

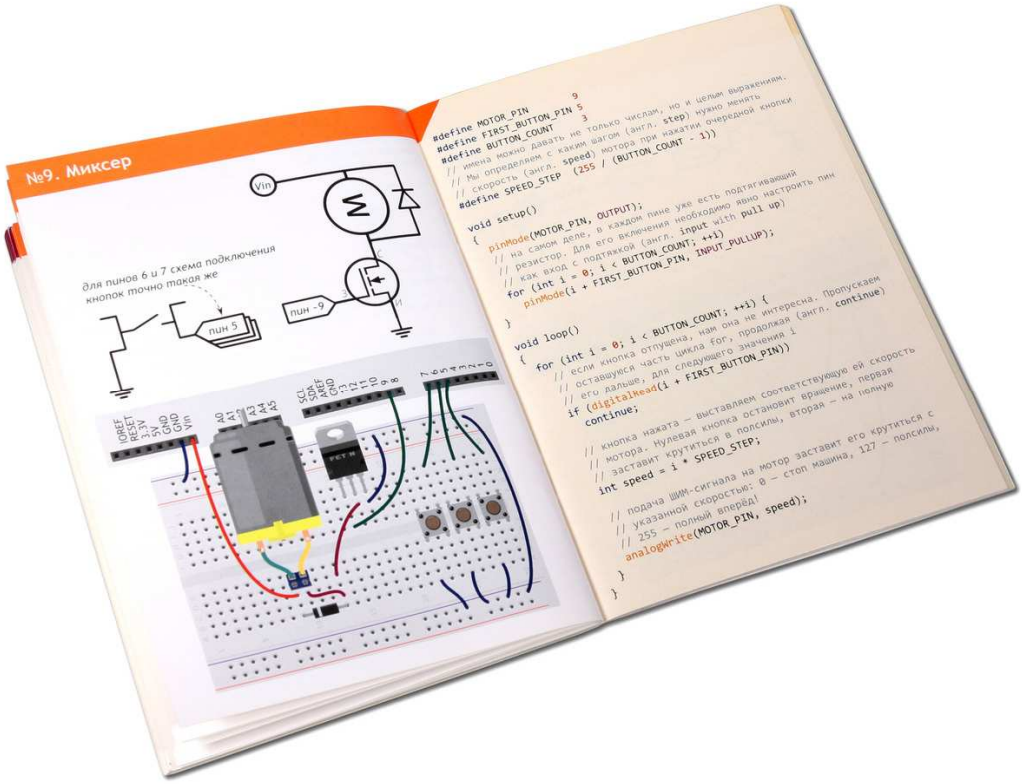
- Уютный форум
- Информативную вики
- YouTube-канал с обзорами и обучающими видео

Включайтесь! Скидка читателям «РЕ» по кодовому слову «RE-34» до 1 марта.

Готовые наборы от Амперки



Амперка производит стартовые наборы для новичков. В них вы найдёте всё необходимое, чтобы легко пройти путь от полного нуля до продвинутого уровня. Для освоения Arduino обратите внимание на наборы «Матрёшка». Они содержат саму плату, кучу периферии и красочный обучающий конспект.



Arduino Uno

amperka.ru
Амперка



Последняя модель Arduino Rev3 выполнена на базе процессора ATmega328P с тактовой частотой 16 МГц, обладает памятью 32 кб и имеет 20 контролируемых контактов ввода и вывода для взаимодействия с внешним миром.

Arduino — это открытая платформа, которая позволяет собирать всевозможные электронные устройства. Arduino будет интересен креативщикам, дизайнерам, программистам и всем пытливым умам, желающим собрать собственный гэджет. Устройства могут работать как автономно, так и в связке с компьютером. Всё зависит от идеи.

Платформа состоит из аппаратной и программной частей; обе чрезвычайно гибки и просты в использовании. Для программирования используется упрощённая версия C++, известная так же как Wiring. Разработку можно вести как с использованием бесплатной среды Arduino IDE, так и с помощью произвольного C/C++ инструментария. Поддерживаются операционные системы Windows, MacOS X и Linux.

Для программирования и общения с компьютером вам понадобится USB-кабель. Для автономной работы потребуется блок питания на 7,5—12 В. Если вы новичок в конструировании электроники, рекомендуем обратить внимание на готовые наборы: «Матрёшка X», «Матрёшка Y» или «Матрёшка Z».

Питание

Arduino Uno может питаться как от USB подключения, так и от внешнего источника: батарейки или обычной электрической сети. Источник определяется автоматически.

Платформа может работать при наличии напряжения от 6 до 20 В. Однако при напряжении менее 7 В работа может быть неустойчивой, а напряжение более 12 В может привести к перегреву и повреждению. Поэтому рекомендуемый диапазон: 7–12 В.

На Arduino доступны следующие контакты для доступа к питанию:

- Vin предоставляет тот же вольтаж, что используется для питания платформы. При подключении через USB будет равен 5 В.
- 5V предоставляет 5 В вне зависимости от входного напряжения. На этом напряжении работает процессор.
- 3.3V предоставляет 3,3 В.
- GND — земля.

Память

Платформа оснащена 32 кб flash-памяти, 2 кб из которых отведено под так называемый bootloader. Он позволяет прошивать Arduino с обычного компьютера через USB. Эта память постоянна и не предназначена для изменения по ходу работы устройства. Её предназначение — хранение программы и сопутствующих статичных ресурсов.

Также имеется 2 кб SRAM-памяти, которые используются для хранения временных данных вроде переменных программы. По сути, это оперативная память платформы. SRAM-память очищается при обесточивании.

Ещё имеется 1 кб EEPROM-памяти для долговременного хранения данных. По своему назначению это аналог жёсткого диска для Arduino.

Ввод / вывод

На платформе расположены 14 контактов, которые могут быть использованы для цифрового ввода и вывода. Какую роль исполняет каждый контакт, зависит от вашей программы. Все они работают с напряжением 5 В, и рассчитаны на ток до 40 мА. Также каждый контакт имеет встроенный, но отключённый по умолчанию резистор на 20 - 50 кОм. Некоторые контакты обладают дополнительными ролями:

- Serial: 0-й и 1-й. Используются для приёма и передачи данных по USB.
- Внешнее прерывание: 2-й и 3-й. Эти контакты могут быть настроены так, что они будут провоцировать вызов заданной функции при изменении входного сигнала.
- ШИМ: 3-й, 5-й, 6-й, 9-й, 10-й и 11-й. Могут являться выходами с широтно-импульсной модуляцией с 256 градациями.
- LED: 13-й. К этому контакту подключен встроенный в плату светодиод. Если на контакт выводится 5 В, светодиод зажигается; при нуле — светодиод гаснет.

Помимо контактов цифрового ввода/вывода на Arduino имеется 6 контактов аналогового ввода, каждый из которых предоставляет разрешение в 1024 градации.

Взаимодействие

Arduino Uno обладает несколькими способами общения с другими Arduino, микроконтроллерами и обычными компьютерами. Платформа позволяет установить последовательное (Serial UART TTL) соединение через контакты 0 (RX) и 1 (TX). Установленный на платформе чип ATmega16U2 транслирует это соединение через USB: на компьютере становится доступен виртуальный COM-порт. Программная часть Arduino включает утилиту, которая позволяет обмениваться текстовыми сообщениями по этому каналу.

Встроенные в плату светодиоды RX и TX светятся, когда идёт передача данных между чипом ATmega162U и USB компьютера.

Отдельная библиотека позволяет организовать последовательное соединение с использованием любых других контактов, не ограничиваясь штатными 0-м и 1-м.

С помощью отдельных плат расширения становится возможной организация других способов взаимодействия, таких как ethernet-сеть, радиоканал, Wi-Fi.

Arduino Leonardo

amperka.ru
Амперка



Arduino Leonardo — это плата схожая по характеристикам с Arduino Uno, но с несколько отличающимся микроконтроллером и его обвязкой.

В качестве микроконтроллера используется ATmega32U4. Он же используется и в качестве USB-UART преобразователя для прошивки. Этим Arduino Leonardo выделяется среди остальных плат Arduino, где для коммуникации используется дополнительный микроконтроллер. Это имеет следующие последствия:

- Arduino Leonardo смотрит в сторону USB через виртуальный serial-порт, не через аппаратный. Это означает, что 0-й и 1-й контакты аппаратного порта остаются свободными

и вы можете использовать их одновременно с коммуникацией с компьютером. Виртуальный serial-порт доступен через класс Serial, а аппаратный — через класс Serial1.

- При сбросе микроконтроллера его соединение с компьютером теряется и поднимается заново. Это усложняет процесс прошивки, но не должно являться особой проблемой, т.к. Arduino IDE поддерживает этот процесс.
- При открытии serial-соединения с компьютера загруженный скетч не перезагружается. Это означает, что если вы отсылаете данные по serial до реального открытия соединения, они уйдут в никуда и вы не сможете их увидеть.
- С точки зрения компьютера Arduino Leonardo является HID устройством (вроде клавиатуры или мыши), поэтому сделать на основе Arduino Leonardo новый компьютерный эмулятор проще, чем с другими платами

Кроме того, в сравнении с Arduino Uno есть различия в распиновке. В частности контакты 4, 6, 8, 9, 10 и 12 подключены к АЦП, а следовательно могут быть использованы как аналоговые входы (A6-A11) в дополнение к обычным A0-A5

По габаритам и допустимому вольтажу плата идентична Arduino Uno.

Если вы новичок и сомневаетесь в выборе между Arduino Leonardo и Arduino Uno — берите Arduino Uno, с ним проще работать, а уровень совместимости с другими железками выше.

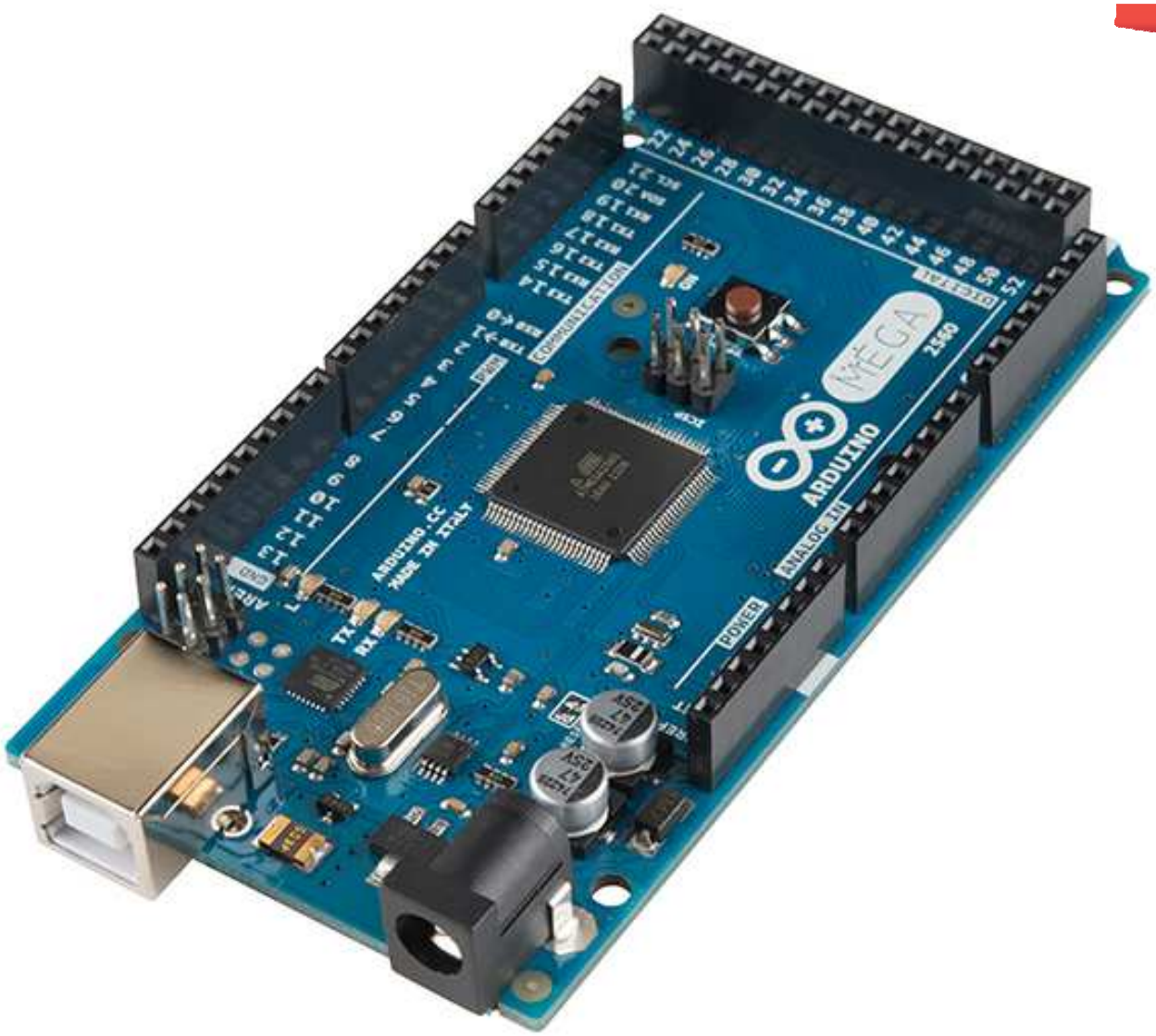
Сравнительная таблица

	Uno	**Leonardo**
Микроконтроллер	ATmega328P	ATmega32U4
Количество цифровых входов/выходов	20	20
…из них ШИМ поддерживают	6	7
Количество аналоговых входов	6	12
Количество контактов для аппаратного прерывания	2	2
Тактовая частота (МГц)	16	16
Количество аппаратных serial-портов	1	1
USB-разъём	Type B	micro-USB

Arduino Mega 2560

amperka.ru

Амперка



Arduino Mega 2560 — это расширенная версия Arduino Uno. Платформа выполнена на базе более продвинутого чипа ATmega2560, имеет больше контактов и большее количество аппаратных serial-портов для взаимодействия с компьютером и другими устройствами.

Сравнительная таблица

	Uno	Mega
Микроконтроллер	ATmega328P	ATmega2560
Количество цифровых входов/выходов	14	54
…из них ШИМ поддерживают	6	15
Количество аналоговых входов	6	16
Количество контактов для аппаратного прерывания	2	6
Тактовая частота (МГц)	16	16
Количество аппаратных serial-портов	1	4
Объём Flash-памяти (кБ)	32	256
Объём SRAM-памяти (кБ)	2	8
Объём EEPROM-памяти (кБ)	1	4

Совместимость

Платформа выполнена таким образом, чтобы быть максимально совместимой со своими младшими собратьями и модулями расширения. Левая часть платы по конфигурации контактов идентична Arduino Uno, как по расположению, так и по назначению. Это означает, что Arduino Mega 2560 может просто подменить Arduino Uno, если её возможностей перестало хватать.

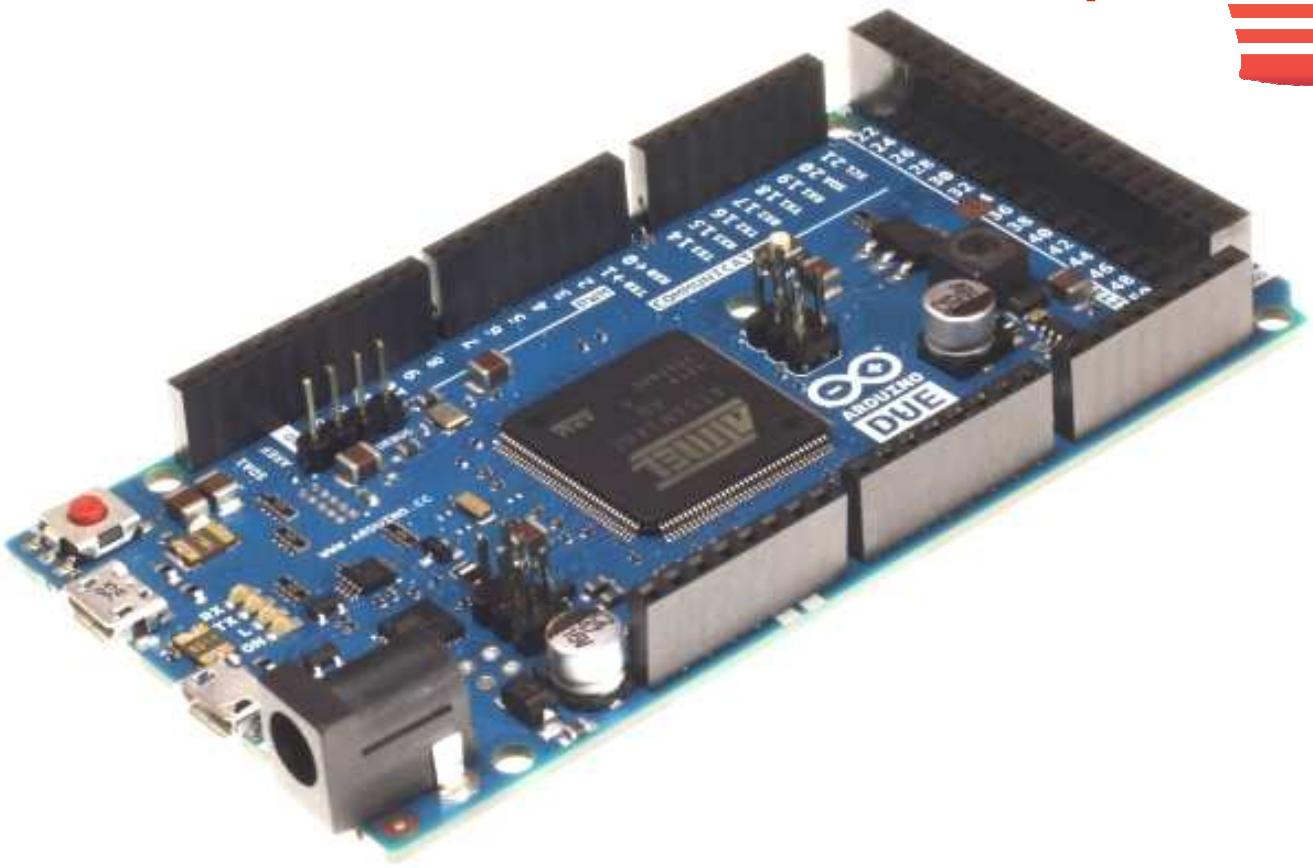
Питание, распределение напряжений, защита USB и принципы взаимодействия аналогичны базовой модели.

Габариты

Размер платы составляет 10,16 × 5,3 см (против 6,9 × 5,3 см базовой модели). Гнёзда для внешнего питания и USB выступают на пару миллиметров за обозначенные границы. На плате предусмотрены места для крепления на шурупы или винты. Расстояние между контактами составляет 0,1” (2,54 мм), но в случае 7-го и 8-го контакта — расстояние: 0,16”.

Arduino Due

amperka.ru
Амперка



Arduino Due — это мощная Arduino, основанная на 32-битном ARM-процессоре AT91SAM3X8E от Atmel. Он обладает тактовой частотой 84 МГц, а его 32-битная архитектура позволяет выполнять большинство операций на целыми числами в 4 байта за один такт.

Кроме этого, Arduino Due предоставляет:

- 96 Кб SRAM (оперативная память)
- 512 Кб флеш-памяти (для хранения программы)
- Прямой доступ к памяти (DMA) для задач, активно работающих с данными в памяти
- 54 цифровых входов/выходов; 12 из них поддерживают ШИМ (PWM)
- 4 аппаратных последовательных порта (UART)
- 12 аналоговых входов
- 2 цифро-аналоговых преобразователя (DAC) для 2 аналоговых выходов
- 2 шины TWI / I^2C
- SPI-разъём
- JTAG-разъём
- Поддержка USB On The Go (USB OTG) для подключения других USB-устройств

В отличии от большинства плат Arduino, родным напряжением Arduino Due является 3,3 В, а не 5 В. Соответственно, выходы для логической единицы выдают 3,3 В, а в режиме входа ожидают принимать не более 3,3 В.

Контакты

- Цифровые входы/выходы: контакты 0–53. Работают на напряжении 3,3 В. В режиме выхода могут выдавать ток 3 или 15 мА (в зависимости от контакта); в режиме входа — принимать ток 6 или 9 мА (в зависимости от контакта). К контактам также подключены подтягивающие резисторы по 100 кОм, которые по умолчанию выключены, но могут быть включены программно.
- Аппаратные последовательные порты (RX/TX): 0/1, 19/18, 17/16, 15/14. Передача данных осуществляется на уровне 3,3 В. Первая пара также соединена с чипом ATmega16U2, отвечающим за подключение через USB к компьютеру.
- Широтно-имульсная модуляция (ШИМ/PWM): контакты 2–13. Дают возможность выдавать аппаратный шим с разрешением 8 бит (256 градаций).
- SPI — отдельная группа контактов 2×3. На Arduino Due используется только для общения по SPI-интерфейсу с другими устройствами. Он не может быть использован для программирования контроллера, как на других Arduino. По расположению он в точности совпадает с расположением на Arduino Uno, Arduino Mega 2560, Arduino Leonardo, а следовательно даёт возможность работы с платами расширения его использующими, таких как Ethernet Shield.
- CAN-шина: контакты CANRX и CANTX. Позволяют использовать Arduino Due в автомобильных сетях. Поддержка с программной стороны пока не реализована производителем.
- Встроенный светодиод: контакт 13 (L). Для простой индикации. В отличии от Arduino Uno и Mega, он поддерживает ШИМ.
- Шины TWI/I^2C: 20(SDA)/21(SCL), SDA1/SCL1. Для общения с периферией по синхронному протоколу, через 2 провода.
- Аналоговые входы: контакты A0–A11. Принимают сигнал до 3,3 В. Большее напряжение может вывести процессор из строя. Аналоговые входы предоставляют разрешение до 12 бит (4096 градаций), хотя по умолчанию настроены на разрешение в 10 бит для совместимости со скетчами для других моделей Arduino.
- Цифро-аналоговый преобразователь: контакты DAC1 и DAC2. Позволяют выдавать настоящий аналоговый сигнал с 12-битным разрешением (4096 градации), например, для устройств, связанных с обработкой звука.
- Сброс процессора: RESET. Позволяет аппаратно перезагружать плату.
- Входное напряжение: Vin. Выдаёт напряжение, поданное внешним источником, либо может являться входом для внешнего питания.
- Стабилизированные 5 В: контакт 5V. Позволяет получать ровные 5 В и ток до 800 мА.
- Стабилизированные 3,3 В: контакт 3.3V. Позволяет получать ровные 3,3 В и ток до 800 мА.

- Общая земля: GND.
- Опорное напряжение для плат расширения: IOREF. Платы расширения должны «советоваться» с этим контактом, чтобы правильно определять родное напряжение родительской платы. Arduino Due выдаёт на IOREF 3,3 В.

Память

- На борту SAM3X — 2 блока по 256 Кб флеш-памяти для хранения программы
- Загрузчик (bootloader) располагается в отдельной памяти только для чтения и прошит на заводе Atmel
- Оперативная SRAM-память поделена на 2 банка: 64 и 32 Кб

Любая память доступна для последовательной адресации из программы. Содержимое флеш-памяти (программа) может быть очищено зажатием на несколько секунд кнопки Erase на плате.

Коммуникация

Arduino Due позволяет взаимодействовать с компьютером, другими Arduino, микроконтроллерами и различными устройствами вроде телефонов, планшетов, фотоаппаратов. Для этого плата предоставляет три аппаратных последовательных порта (UART/USART), две шины TWI/I²C, интерфейс SPI и USB-порт.

Один USB-порт (programming) используется для прошивки Arduino Due. Он подключён к чипу ATmega16U2 на плате, который является мостом между USB и аппаратным портом SAM3X, используемым для программирования процессора и связи с компьютером.

Второй USB-порт (native) может использоваться для связи с другими устройствами как в режиме slave (эмуляция мыши, клавиатуры), так и в режиме host (приём данных с фотоаппаратов, управление мышью, клавиатурой, телефоном).

Совместимость

Платформа по своему форм-фактору полностью совпадает с Arduino Mega 2560. Это означает механическую совместимость со всеми платами расширения для Arduino Mega, Arduino Uno, Arduino Leonardo.

Однако, в силу того, что родным является напряжение в 3,3 В, а не 5 В, как на других моделях, стоит обязательно удостоверяться в возможности подключения платы расширения к Arduino Due.

Arduino Yún

amperka.ru
Амперка



Вам не хватает производительности Arduino Ethernet для работы с сетью? Воспользуйтесь Arduino Yún!

Arduino Yún — это плата на основе Arduino Leonardo, объединяющая в себе достоинства двух платформ, поддерживаемых Свободным Сообществом — Arduino и Linux. Получившийся симбиоз предоставляет огромные возможности для использования интернета в своих проектах.

На Arduino Yun можно поднять небольшой сайт и использовать его как главное устройство умного дома. И даже если вы запрячете его в труднодоступное место, вы всегда можете поменять его прошивку через Wi-Fi. Вы можете сконструировать на его основе робота с веб-камерой, изображение с которой вещается прямо на ваш компьютер.

Благодаря библиотеке Temboo, ваш Arduino Yun может использовать более 100 различных интернет-сервисов, таких как Facebook, GMail, PayPal и т.д. Например, он может отправлять каждый час температуру воздуха в комнате в Twitter. Или отправлять файлы в ваш Dropbox.

Arduino

Arduino-часть содержит микроконтроллер ATmega32U4, работающий на частоте 16 МГц. Распиновка Arduino Yún аналогична Arduino Leonardo. Вместе с Arduino Yún вы можете использовать большинство плат расширения Arduino.

Linux

Linux-часть Arduino Yún использует микрокомпьютер Atheros AR9331, работающий под управлением операционной системы Linino. Linino — это специально подготовленная версия популярного дистрибутива Linux для встраиваемых систем — OpenWRT.

Микрокомпьютер работает на частоте 400 МГц, имеет 64 МБ оперативной и 16 МБ flash-памяти, встроенный Wi-Fi-интерфейс, Ethernet-интерфейс, USB-хост и слот для micro-SD карты. Linino содержит в себе пакетный менеджер opkg, который позволяет устанавливать большое количество Linux-приложений, а так же интерпретатор языка Python 2.7, с помощью которого вы можете писать свои приложения для Linino.

Память для Linux-приложений может быть расширена с помощью съёмного носителя (micro-SD карты или USB-флешки). Для этого нужно лишь создать папку с названием «arduino» в корне съёмного носителя.

Связь между Atmega32U4 и Atheros AR9331 осуществляется с помощью библиотеки Bridge.

Питание

Для питания Arduino Yún может использоваться micro USB разъём или контакт Vin. Ей необходимо постоянное напряжение 5 В. В отличии от Arduino Leonardo, плата не имеет встроенного регулятора питания, поэтому повышенное напряжение питания может повредить устройство.

Связь

Кконтакты 0 (RX) и 1 (TX). Используются для приёма (RX) и передачи (TX) последовательных данных с помощью аппаратного устройства связи ATmega32U4. Аппаратные последовательные порты в ATmega32U4 и AR9331 в Yún соединены между собой и используются для связи между этими процессорами. Последовательный интерфейс Atheros AR9331 напрямую связан с интерфейсом командной строки. Это означает, что вы можете получить доступ к программам Linux из своего скетча: просто посылайте bash-команды в текстовом виде и принимайте ответ.

Последовательный порт, связывающий с USB отделён от аппаратного serial-интерфейса, поэтому общение с AR9331 и с USB не связано и может производиться одновременно и независимо.

Программирование

Arduino Yún может быть запрограммирована через micro USB вход. Также доступно программирование через Wi-Fi, если ваша Arduino Yún находится в той же сети, что и ваш компьютер.

Для работы с Arduino Yun необходимо установить Arduino IDE версии 1.5.5 и выше.

Характеристики

AVR Arduino микроконтроллер	
Микроконтроллер	ATmega32U4
Рабочее напряжение	5 В
Количество цифровых входов/выходов	20
…из них PWM поддерживают	7
Количество аналоговых входов	12
Количество контактов для аппаратного прерывания	5 Объём Flash-памяти
Объём SRAM-памяти	2,5 КБ
Объём EEPROM-памяти	1 КБ
Тактовая частота	16 МГц
Linux микропроцессор	
Процессор	Atheros AR9331
Архитектура	MIPS @400 МГц
Рабочее напряжение	3,3 В
Ethernet	IEEE 802.3 10/100 Мбит/сек
Wi-Fi	IEEE 802.11b/g/n
USB	Type-A 2.0 хост/устройство
SD-модуль	Micro-SD
Объём RAM-памяти	64 МБ DDR2 Объём Flash-памяти

Arduino Ethernet

amperka.ru

Амперка



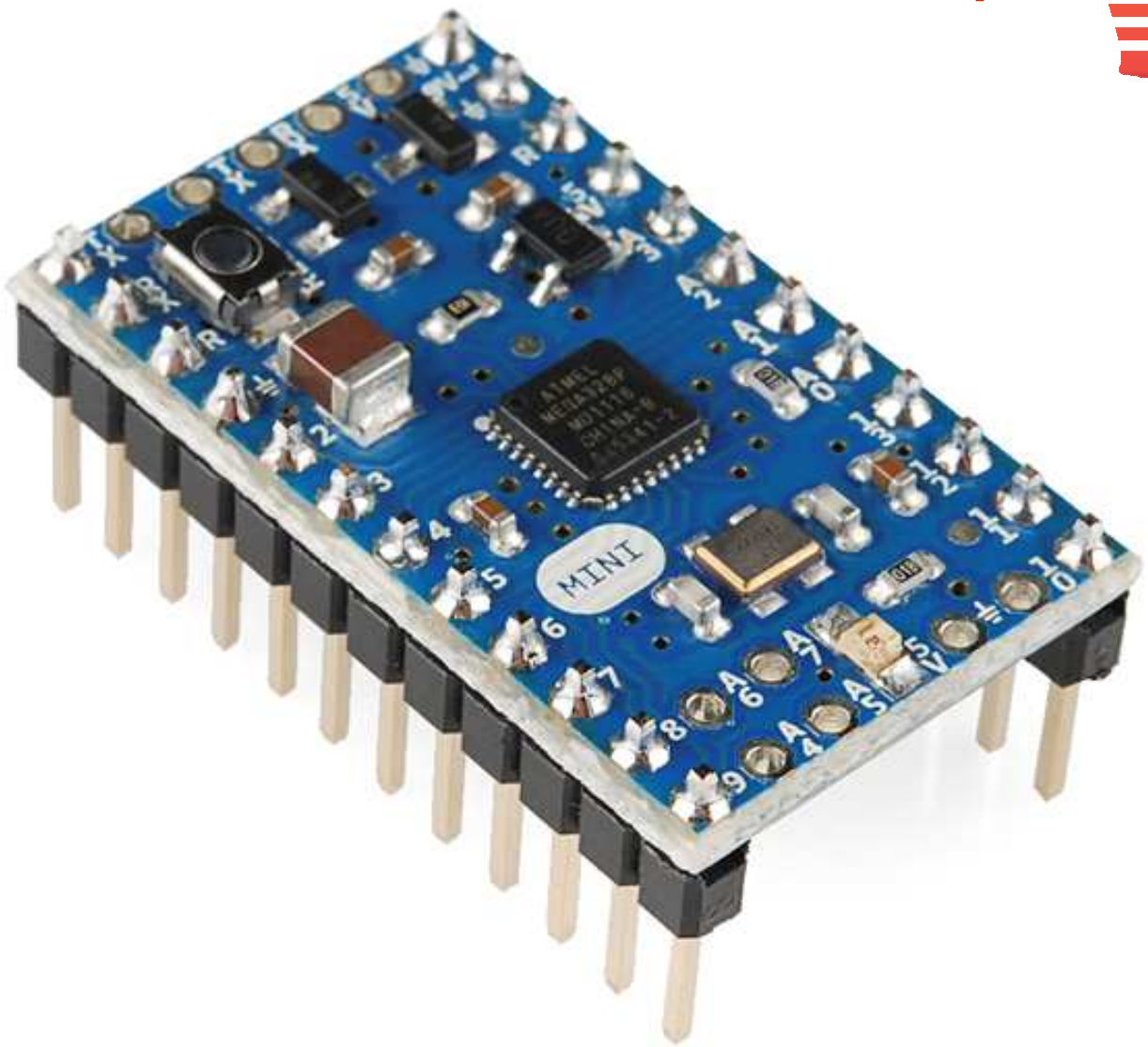
Arduino Ethernet — это симбиоз Arduino Uno и Ethernet Shield на одной плате размером с Arduino Uno.

На Arduino Ethernet отсутствует разъём для USB, на его месте расположен разъём 8P8C для витой пары. Поэтому, чтобы прошивать микроконтроллер Arduino понадобится отдельный модуль USB-Serial converter, который временно устанавливается на штырьковые контакты, на плате.

Arduino Mini

amperka.ru

Амперка



Arduino Mini — это Arduino Uno, исполненный на компактной плате. Он лишён USB порта и разъёма для питания. В качестве чипа используется тот же ATmega328P. Arduino Mini предназначен для применения в местах, где занимаемое пространство критично.

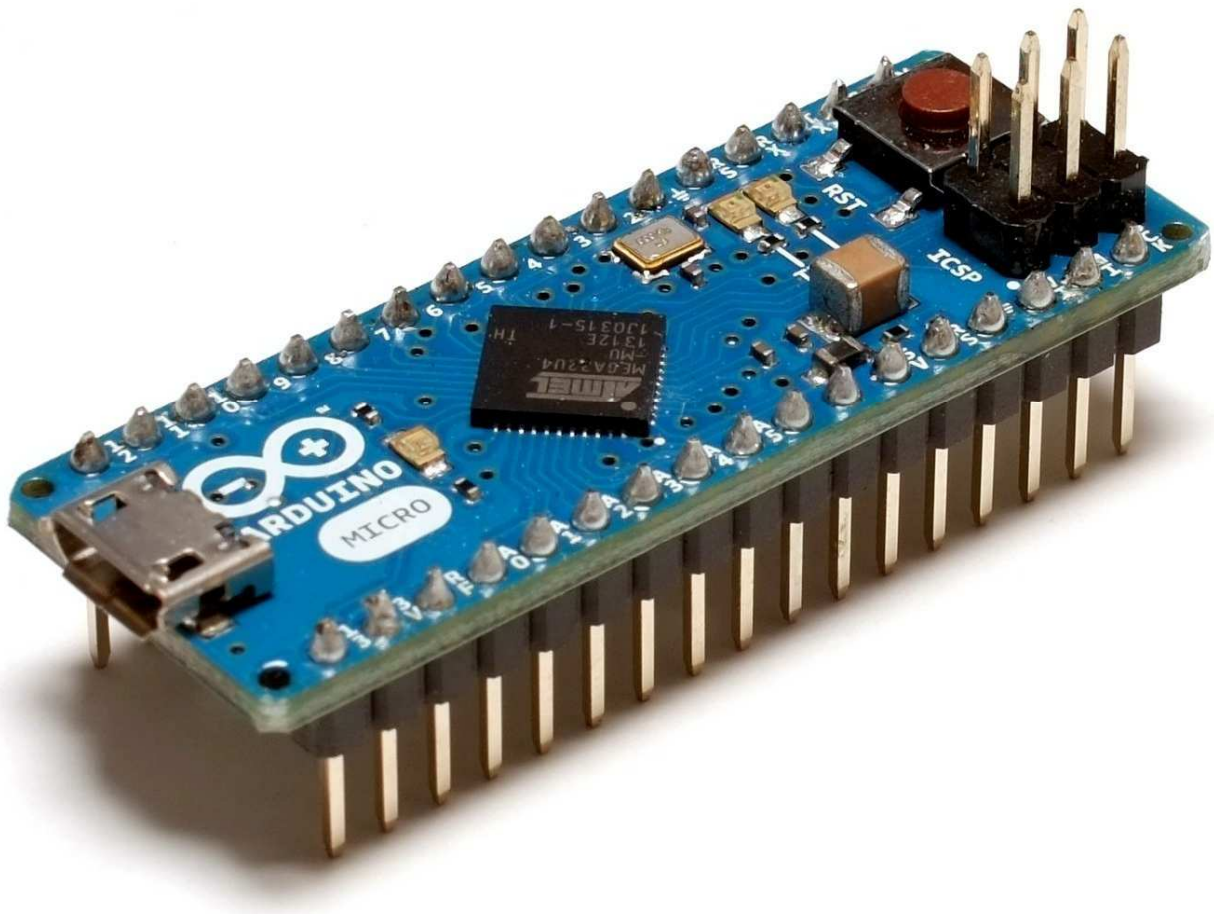
Взаимодействие

Для прошивки микроконтроллера, в виду отсутствия собственного USB-порта, можно использовать отдельный USB-Serial Converter. На время соединения портов конвертера с serial-портом платформы, взаимодействие осуществляется так же, как и с обычным Arduino.

Arduino Micro

amperka.ru

Амперка



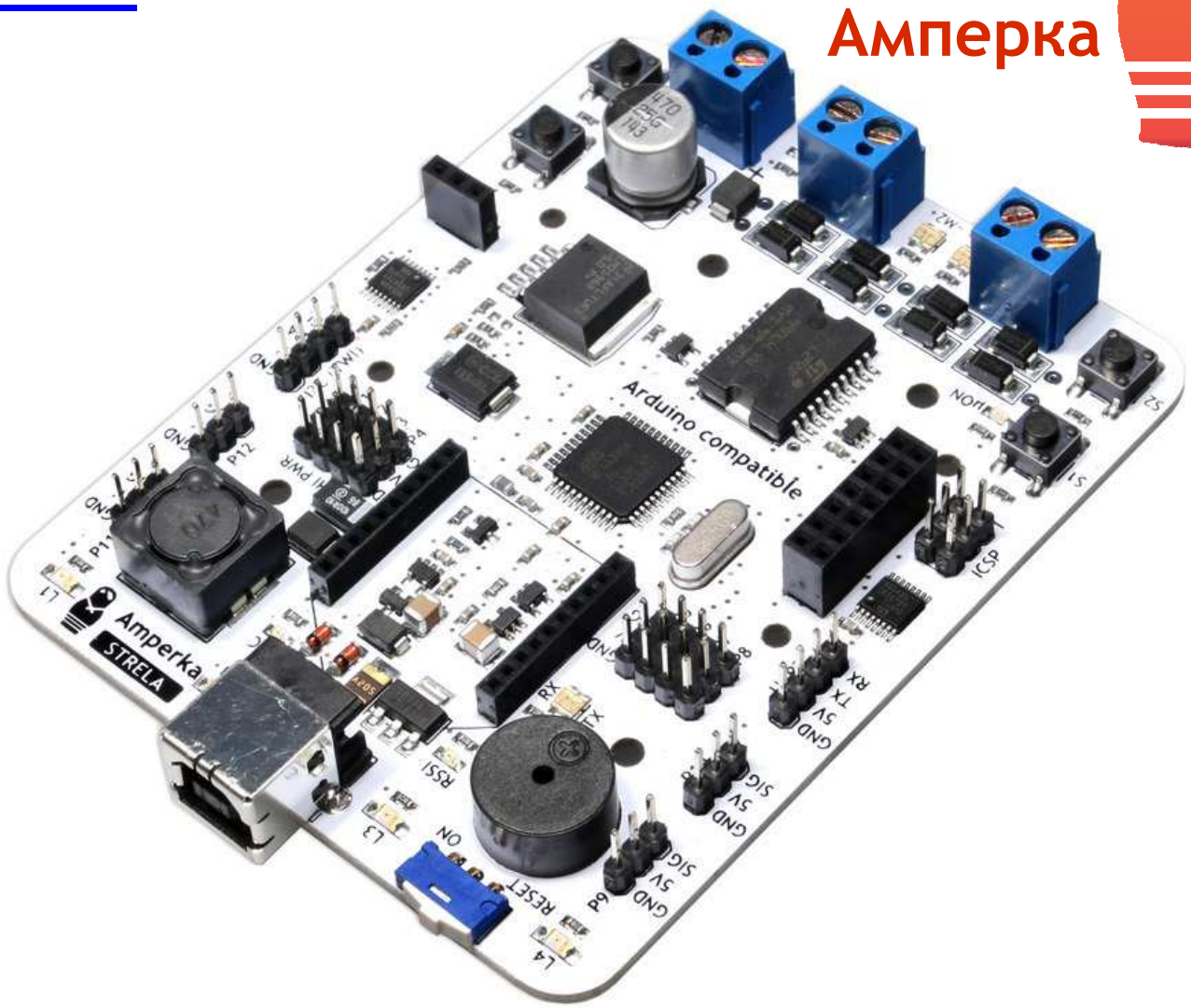
Arduino Micro — это Arduino Leonardo, исполненный на компактной плате. Отличие заключается в отсутствии собственного гнезда для внешнего питания, но оно может быть подведено непосредственно к контакту Vin.

В остальном, начинка и способы взаимодействия совпадают с Arduino Leonardo. Он также имеет один микроконтроллер ATmega32U4 и для прошивки через USB, и для исполнения программ; также может выступать в роли клавиатуры или мыши; предоставляет то же количество памяти, цифровых, аналоговых и ШИМ-портов.

Strela

amperka.ru

Амперка



Вы хотите построить робота, но не знаете какая электроника вам понадобится? Платформа Strela идеально подходит для создания лёгкого мобильного робота и содержит в себе всё необходимое для этого.

Arduino-совместимая платформа Strela разработана и выпускается Амперкой как материнская плата для постройки роботов. Одна эта плата призвана заменить собой Arduino с несколькими шилдами, которые обычно применяются в роботах. А благодаря библиотеке Strela, работа с платой не вызовет затруднений даже у начинающих роботостроителей.

«Стрела» отлично подходит для соревнований. Вы можете сделать на её основе колёсного робота, который отлично себя покажет при езде по линии, в сумо, в лабиринте и во многих других категориях. А благодаря наличию на плате разъёма для подключения беспроводных модулей связи формата XBee не составит труда сделать робота с беспроводным управлением.

Микроконтроллер

«Мозг» платы — микроконтроллер ATmega32U4. Это тот же самый микроконтроллер, что установлен в Arduino Leonardo. Этот контроллер используется и в качестве USB-UART преобразователя для прошивки.

Входы/выходы ATmega32U4

Одиннадцать входов/выходов микроконтроллера ATmega32U4 выведены на специальные тройки контактов «питание–земля–сигнал», где питание соединено с контактом 5V, земля — с GND, а сигнал с соответствующим цифровым или аналоговым контактом. Такое расположение входов/выходов позволит подключить к платформе большое количество периферии вроде сенсоров через стандартные 3-проводные шлейфы. Это позволяет не прибегать к пайке, и к отдельно стоящему breadboard'у.

Контакты питания пинов P1–P4 соединены широкими контактными дорожками с DC–DC преобразователем. Это позволяет подключать к этим контактам достаточно мощную нагрузку, напримерсервоприводы.

- Контакты P4–P9, P11, P12 подключены к АЦП, а следовательно могут быть использованы как аналоговые входы.
- Широтно-имульсная модуляция (ШИМ/PWM) доступна на контактах P1–P4.
- Контакт P12 может быть использован для подключения ИК-приёмника, такого как TSOP22.

Входы/выходы I²C-расширителей портов

На платформе Strela имеются 2 расширителя цифровых входов/выходов PCA9554. Один из них имеет адрес I²C-адрес 0x20, используется для управления встроенными светодиодами и кнопками. Второй имеет I²C-адрес 0x21 для управления LCD-экраном MT-08S2A-2YLG (в комплект не входит) и контактом P10. При отсутствии экрана вы можете использовать эти контакты для управления вашими устройствами.

Оборудование доступное для управления по I2C:

- 4× Кнопка
- 4× Светодиод
- LCD-экран MT-08S2A-2YLG
- Пин P10

Связь

Платформа Strela поддерживает несколько различных протоколов связи:

- **USB** используется для прошивки робота, и как виртуальный последовательный порт для связи с компьютером.
- **SPI**. Контакты SPI выведены на стандартный блок контактов ICSP. Три контакта шины SPI могут быть использованы в качестве цифровых входов/выходов микроконтроллера.
- **UART**. Контакты UART выведены на четырёхштырьковый разъём (RX, TX, 5V, GND). Контакты RX и TX используются и для коммуникации с беспроводными модулями связи формата XBee, а так же могут быть использованы в качестве цифровых входов/выходов микроконтроллера.
- **TWI/I²C** используется для связи с установленными на плате расширителями портов ввода/вывода, которые имеют адрес 0x20 и 0x21. Контакты TWI/I²C выведены на четырёхштырьковый разъём (SCL, SDA, 5V, GND), к ним могут быть подключены другие TWI/I²C-устройства. Что бы не мешать работе расширителям портов ввода/вывода, I²C-устройства должны иметь отличный от них адрес.
- **XBee**. На плате находится разъём для подключения беспроводных модулей связи формата XBee. Установив соответствующий модуль связи, вы сможете общаться с платформой через XBee, Wi-Fi или Bluetooth.
- **ИК**. На плате расположено гнездо для подключения ИК-приёмника, такого как TSOP22. Поэтому можно управлять роботом с обычного пульта управления от бытовой техники. Например, с такого пульта можно дать старт роботу во время соревнований.

Силовая часть

Выводы микроконтроллера являются слаботочными, поэтому ток мотора, при подключении его напрямую, выведет их из строя. Эту проблему решает так называемый H-мост. Он позволяет управлять скоростью и направлением вращения мотора с помощью логических сигналов микроконтроллера. Для управления двигателями робота на плате предусмотрен H-мост L298P. Именно эта микросхема отвечает за управление моторами в Motor Shield. Плата имеет 2 независимых канала. Используя их, можно подключить на выбор:

- Пару DC-моторов
- Один двухфазный шаговый мотор.
- Один DC-мотор с током до 4 А, если объединить каналы

На плате расположены светодиоды-индикаторы, показывающие направление и скорость по каждому из каналов.

Питание

Новички при постройке робота часто сталкиваются с проблемой перезагрузки управляющего контроллера при броске нагрузки на двигателях. Обычно эту проблему решают с помощью раздельного питания двигателей и микроконтроллера, или при помощи DC-DC преобразователя. Платформа Strela уже содержит встроенный в плату DC-DC преобразователь LM2596-5.0. Это означает, что вам понадобится всего один источник питания для робота. DC-DC преобразователь на выходе точно выдерживает напряжение 5 В, необходимые для питания микроконтроллера. Поэтому ваш робот не будет перезагружаться в неподходящий момент при резком увеличении оборотов двигателей.

Плата может быть запитана от источника постоянного тока с напряжением 7–24 В. Она может получать питание и от USB, это удобно во время отладки и программирования робота. При этом силовая часть будет получать очень низкое напряжение, и двигатели могут не крутиться.

Reset

На плате присутствует переключатель, который бывает полезен при отладке или при подготовке к старту на соревнованиях. В положении «RESET» контакт Reset микроконтроллера замыкается на землю. В таком положении на микроконтроллер подано напряжение, но он не работает, поэтому робот никуда не убежит при отладке у компьютера. Чтобы запустить микроконтроллер необходимо перевести рычажок в положение «ON».

Пьезодинамик

На плате установлен пьезодинамик, который может пропищать победную мелодию на финише или оповестит о срабатывании датчика. Для работы с ним в Arduino IDE можно использовать встроенную функцию tone или богатую возможностями библиотеку Tone.

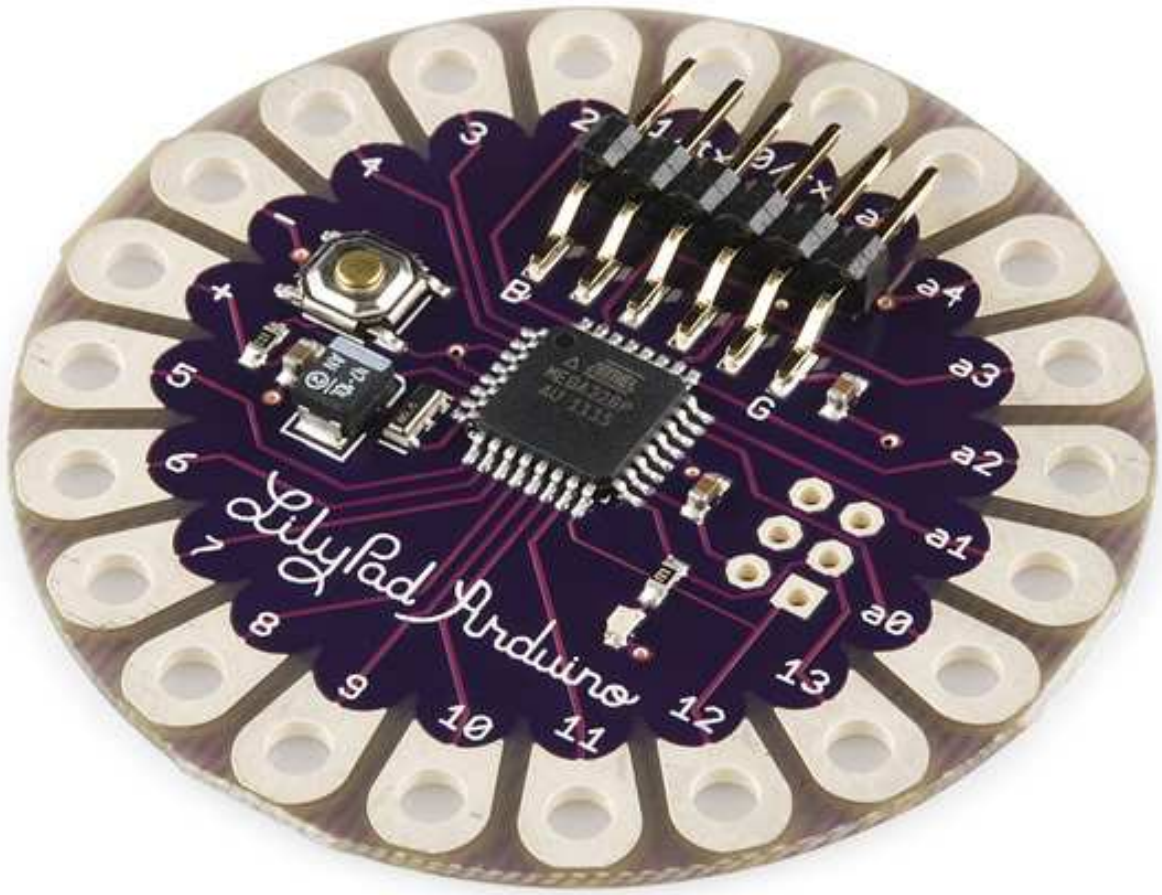
Характеристики

Микроконтроллер	ATmega32U4
Рабочее напряжение	5 В
Максимальный ток через контакт входа/выхода	40 мА
Количество цифровых входов/выходов, выведенных на штырьковые контакты	11
... из них могут использоваться как аналоговые входы	8
Количество каналов PWM	4
Выведенные интерфейсы	UART, SPI, I2C

Объём Flash-памяти	32 КБ
Объём SRAM-памяти	2,5 КБ
Объём внутренней EEPROM-памяти	1 КБ
Тактовая частота	16 МГц
Загрузчик	Arduino Leonardo
I²C-расширитель портов	2× PCA9554
Количество цифровых входов/выходов	16
... из них выведены на штырьковые контакты	1
... используется для управления встроенными светодиодами	4
... используется для считывания положения встроенных кнопок	4
... используется для управления экраном MT-08S2A-2YLG	7
Питание	
DC–DC преобразователь	LM2596-5.0
Выходное напряжение	5 В
Максимальное входное напряжение	24 В
Максимальный ток нагрузки	3 А
Силовая часть	
H-мост	L298P
Количество каналов управления	2
Максимальный ток нагрузки на канал	2 А
Размеры	105×79×14 мм

LilyPad Arduino 328

amperka.ru
Амперка



Вы собираетесь начать проект умной одежды? Тогда вам необходим LilyPad Arduino 328 — то, что делает одежду по настоящему умной.

LilyPad Arduino 328 Main Board — электронный мозг, спроектированный специально для использования в проектах умной одежды. Он выполнен на том же самом микроконтроллере что и Arduino Uno. С его помощью вы можете изготовить большое количество разнообразных устройств, взяв за основу одежду, аксессуары или мягкие игрушки. LilyPad Arduino 328 поможет вам наделить умными функциями всё, что может быть прошитым токопроводящими нитками. Это может быть скатерть, показывающая температуру еды; футболка, светящаяся в

такт музыке; или шорты, вибрирующие когда кому-то понравилось фото вашей кошки в facebook.

На плате вы не найдёте острых углов и ножек компонентов. Контактные колодки припаяны методом поверхностного монтажа, т.е. только с одной стороны. Используя такие LilyPad-модули, вы сводите риск быть натёртым своим же устройством к минимуму.

Модули LilyPad производятся американской компанией SparkFun. Все они специально разработаны для того, чтобы удобно пришиваться к тканям и соединяться между собой с помощью токопроводящих ниток. Одежду с нашитыми модулями можно стирать руками или в стиральной машине при бережном режиме, если не использовать агрессивные отбеливающие средства.

Контакты

На плате расположены 22 контакта-лепестка, предназначенных для пришивания и соединения с другими устройствами при помощи токопроводящих ниток. Два контакта предназначены для питания платы, они обозначены как «+» и «−». Остальные контакты присоединены к управляющему микроконтроллеру ATmega328P и выполняют те же самые функции, что и в Arduino Uno. Каждый из этих 20 контактов может быть использован как цифровой вход или выход. Некоторые контакты имеют специализированные функции:

- Serial: контакты 0 (RX) и 1 (TX). Используются для приёма (RX) и передачи (TX) последовательных данных.
- SPI-интерфейс присутствует на контактах 10, 11, 12, 13.
- TWI / I²C контакты находятся на контактах A4 (SDA) и A5 (SCL).
- Контакты A0–A5 могут быть использованы как аналоговые входы. Каждый из них предоставляет разрешение в 1024 градации. Значение напряжения меряется между землёй и напряжением питания.
- Внешнее прерывание: 2-й и 3-й. Эти контакты могут быть настроены так, что они будут провоцировать вызов заданной функции при изменении входного сигнала.
- PWM: 3-й, 5-й, 6-й, 9-й, 10-й и 11-й. Могут являться выходами с широтно-импульсной модуляцией (pulse-width modulation) с 256 градациями.
- Встроенный светодиод: контакт 13. Горит при высоком уровне на этом контакте.

Кроме контактов-лепестков на плате присутствуют штырьковые контакты для подключения USB-Serial преобразователя, необходимого для программирования платы. Также на плате присутствует кнопка, предназначенная для перезагрузки микроконтроллера.

Питание

LilyPad Arduino 328 не имеет встроенного преобразователя напряжения. Плата может быть запитана от источника напряжения 2,7–5,5 В через специальные контактные лепестки. Для питания LilyPad удобно использовать специальный держатель для батарейки-таблетки или LilyPad-разъём для аккумулятора.

Внимание! Не подключайте LilyPad к источнику напряжения выше 5,5 В! При подключении питания к LilyPad не перепутайте полярность питания. Эти действия приведут к повреждению контроллера.

Программирование

Плата программируется при помощи Arduino IDE. Плата не имеет встроенного USB-разъёма, поэтому, чтобы запрограммировать LilyPad Arduino 328 понадобится отдельный модуль USB-Serial converter, который временно устанавливается на штырьковые контакты на плате. Модуль в комплект не входит: вам понадобится всего один для нескольких LilyPad.

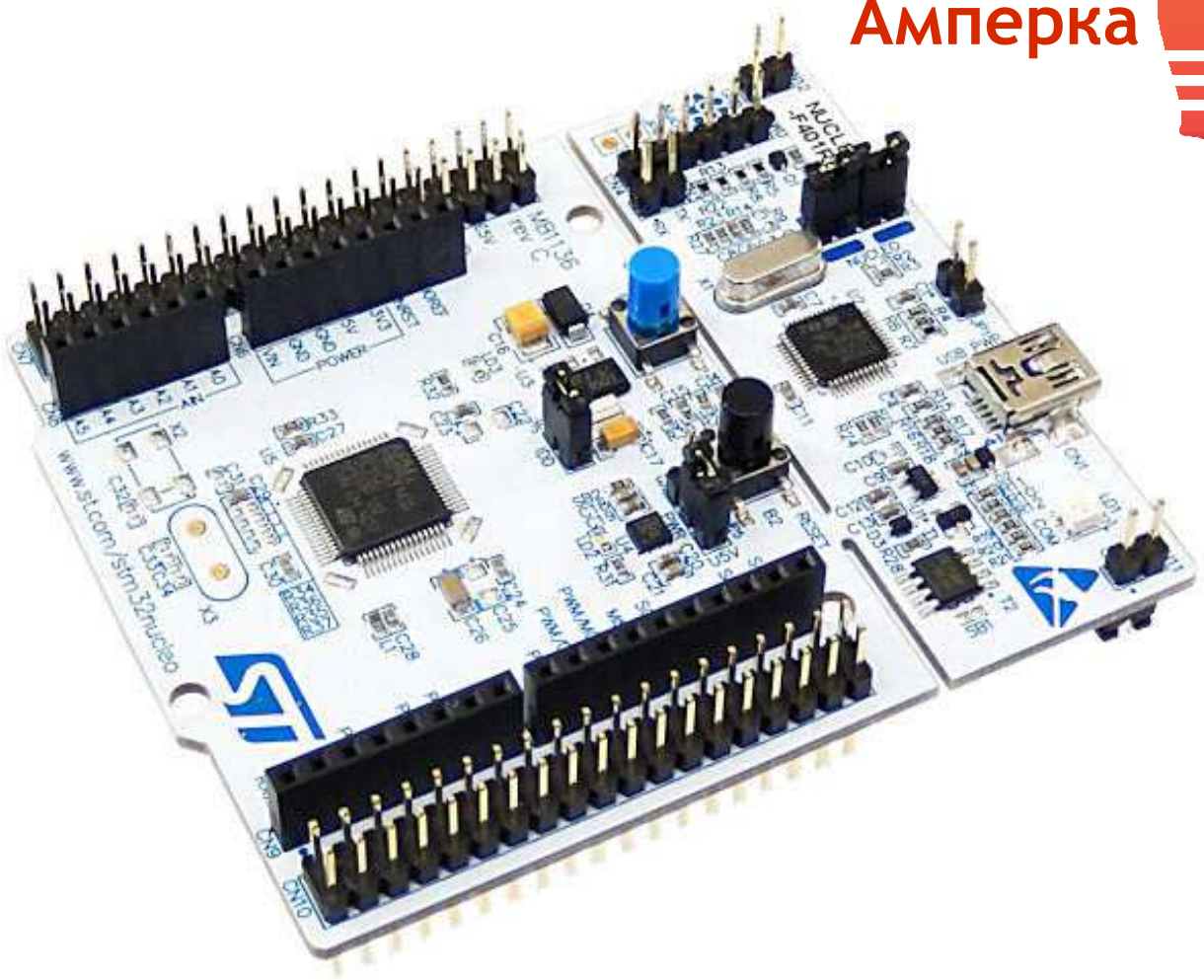
Характеристики

Микроконтроллер	ATmega328P
Количество цифровых входов/выходов	20
…из них PWM поддерживают	6
Количество аналоговых входов	6
Количество контактов для аппаратного прерывания	2
Объём Flash-памяти	32 кб
Объём SRAM-памяти	2 кб
Объём EEPROM-памяти	1 кб
Тактовая частота	8 МГц
Напряжение питания	2,7–5,5 В
Максимальный постоянный ток через контакт ввода/вывода	40 мА
Диаметр	50 мм
Толщина печатной платы	0,8 мм

STM32 Nucleo F401RE

amperka.ru

Амперка



Nucleo — это высокопроизводительная платформа на ARM-процессоре, поддерживающая популярную онлайн среду разработки mbed.

При помощи Nucleo можно разрабатывать устройства, для которых требуется высокая производительность или сложные математические вычисления. Эта платформа основана на 32-разрядном ARM-процессоре STM32F401 с ядром Cortex-M4, работающим на частоте 84 МГц.

Вы сможете полноценно использовать цветные дисплеи, обрабатывать аудиопоток, управлять сложными многосуставными роботами и даже наделять их интеллектом.

Процессор имеет 512 кБ Flash и 96 кБ SRAM-памяти. Flash выполняет те же функции, что и жёсткий диск на компьютере, а SRAM аналогична оперативной памяти.

Nucleo разработан и произведён швейцарской компанией STMicroelectronics — одним из ведущих производителей ARM-процессоров.

Родным напряжением Nucleo является 3,3 В. Однако все входы и выходы толерантны к 5 В, поэтому вы можете подключать к платформе любые модули и шилды, рассчитанные на 5 В.

Программирование

Nucleo поддерживает разные среды разработки. Самой простой для освоения является онлайн среда разработки mbed. Вам не потребуется устанавливать её на компьютер: среда работает прямо из браузера. Компиляция проекта происходит удалённо, вам остаётся только прошить плату бинарным файлом, полученным от среды.

Прошивка платы тоже не представляет проблем: Nucleo определяется как «флешка», на которую вы просто копируете бинарный файл прошивки. Работает это во всех операционных системах, поддерживающих «флешки»: Windows, Linux, MacOS, FreeBSD, QNX и других.

Также, если вы пользуетесь средой mbed, то к вашему распоряжению библиотека классов для работы с периферией. Библиотека документирована и имеет очень простой интерфейс для пользования.

Подробнее о начале работы с mbed и Nucleo вы можете прочитать в вики-статье.

Если вы уже продвинутый разработчик и вам не подходит онлайн среда mbed, вы можете воспользоваться любой стандартной средой разработки для ARM-процессоров: Keil, IAR, GCC.

Периферия

Процессор, на базе которого построена платформа, обладает большим набором периферии. Почти каждый пин может работать не только как цифровой вход или выход, но и иметь другие режимы: АЦП, SPI, I^2C, PWM и т.д. В дополнение на каждый пин может быть назначено прерывание.

10 высокоскоростных 12-разрядных АЦП позволяют оцифровывать сигналы частотой до 240 кГц. Три SPI-интерфейса работают на частоте 42 МГц, позволяя подключать по нему видеокамеры. I^2C интерфейс, работающий на частоте до 1 МГц, легко осилит аудиомодуль. Наиболее востребованных UART-интерфейсов на плате целых три, два из них работают на скоростях до 10500000 бод, а третий — до 5250000 бод. В процессор также встроены часы реального времени и ИК-порт.

Для того, чтобы процессор мог не только принимать или передавать данные по таким высокоскоростным интерфейсам, он имеет модуль DMA.

Совместимость с Arduino

Nucleo F401RE имеет на плате разъёмы, соответствующие Arduino Uno R3. Если вы уже работали с Arduino, то вам будет легко перейти на Nucleo: вы сможете использовать большинство шилдов и модулей, сделанных специально для Arduino.

Питание

Для питания может использоваться mini USB-разъём, контакт 5V (ровно 5 В) или Vin (7–12 В). Одновременное использование двух способов питания невозможно: источник необходимо выбрать перемычкой JP5.

Для программирования и общения с компьютером вам понадобится mini USB-кабель.

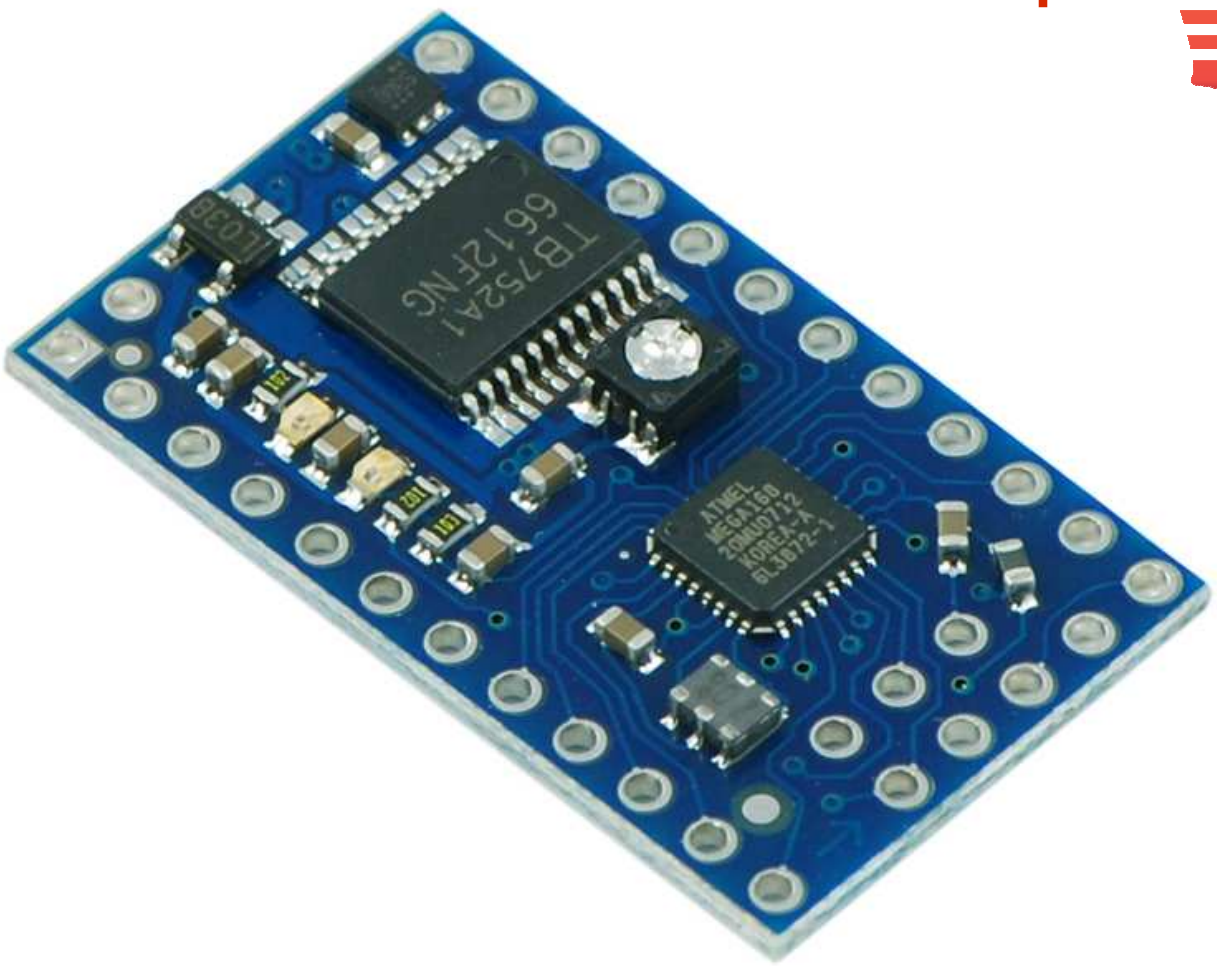
Характеристики

- Ядро: Cortex-M4
- Рабочая частота: 84 МГц
- Разрядность: 32 бита
- Набор машинных инструкций: ARMv7E-M
- Производительность: 105 DMIPS
- Объём SRAM: 96 кБ
- Объём Flash: 512 кБ
- Количество цифровых пинов: до 81
- Количество аналоговых входов: 10
- Разрядность АЦП: 12 бит
- Количество SPI: 4
- Количество I^2C: 3
- Количество UART: 3
- Количество таймеров: 10
- Напряжение питания: 5 В, 7–12 В

Baby Orangutan B-328

amperka.ru

Амперка



Вы опытный Arduino-конструктор, среда Arduino уже кажется вам тесноватой и хочется более глубокого погружения в мир программирования микроконтроллеров? Baby Orangutan откроет для вас новые горизонты!

Baby Orangutan — это плата, предназначенная для изготовления мобильных роботов. Она выполнена на базе процессора ATmega328P с тактовой частотой 20 МГц, имеет на борту двухканальный драйвер двигателей TB6612FNG, регулятор напряжения, защиту от переполюсовки питания, два встроенных светодиода и потенциометр. И всё это при размере меньшем, чем у Arduino Mini — всего 31×18 мм!

Baby Orangutan идеально подходит для переделки старых игрушек, или создания миниатюрных роботов на небольших шасси, таких как робоплатформа Pololu Zumo. Она отлично умещается на breadboard mini. Платформу можно использовать как компактное устройство, управляющее мощным оборудованием, таким как реле или электромагнитные клапаны.

Это оригинальная плата, производится американской компанией Pololu.

Программирование

Baby Orangutan изначально не предназначена для программирования в Arduino IDE. Микроконтроллер не имеет загрузчика и частота его работы не совместима с Arduino. Написать программу для микроконтроллера Baby Orangutan можно при помощи специализированных сред разработки, таких как Atmel Studio, или с использованием произвольного C/C++ инструментария. Baby Orangutan можно запрограммировать через ICSP-разъём при помощи программатора, такого как AVRISP STK500, или при помощи Arduino-платы, превратив её в программатор при помощи скетча Arduino ISP.

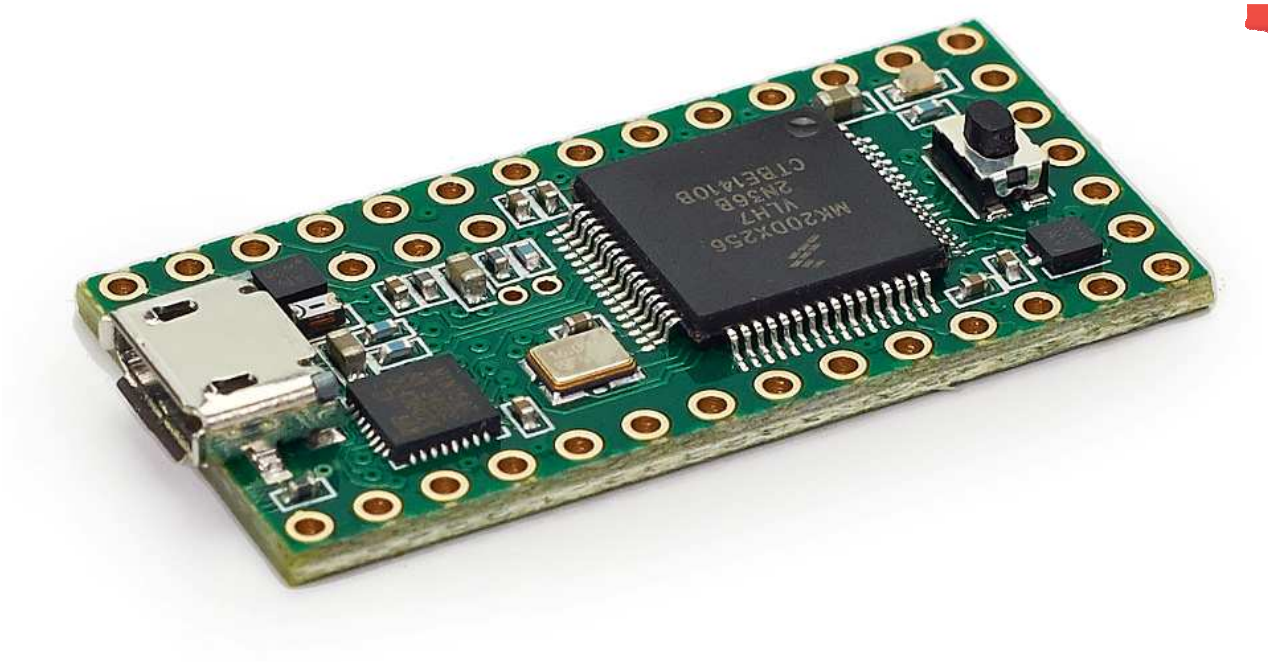
Характеристики

Микроконтроллер	ATmega328P
Рабочее напряжение	5 В
Максимальный ток через контакт входа/выхода	40 мА
Количество цифровых входов/выходов, выведенных на штырьковые контакты	18
... из них могут использоваться как аналоговые входы	8
Количество каналов PWM	2
Объём Flash-памяти	32 КБ
Объём SRAM-памяти	2 КБ
Объём внутренней EEPROM-памяти	1 КБ
Тактовая частота	20 МГц
Питание	
Входное напряжение	6–13,5 В
Силовая часть	
H-мост	TB6612FNG
Количество каналов управления	2
Максимальный ток нагрузки на канал	1 А
Размеры	31×18 мм

Teensy 3.1

amperka.ru

Амперка



Teensy — это семейство компактных платформ, поддерживающих Arduino IDE. Старшие представители этого семейства (версий 3.x) построены на высокопроизводительных ARM-процессорах с ядром Cortex-M4. Они обладают высокой производительностью, большим количеством памяти и широким набором интерфейсов.

По размерам плата сопоставима с Arduino Mini.

Процессор и память

Teensy 3.1 построена на 32-разрядном ARM-процессоре Cypress MK20DX256 с ядром Cortex-M4, частотой 72 МГц, 64 кБ оперативной памяти и 256 кБ энергонезависимой Flash-памяти. Это дает возможность разрабатывать сложные и требовательные к ресурсам программы, такие как обработка звука, управление многосуставными роботами и распознавание образов.

Процессор также имеет развитый DMA-контроллер, позволяющий пересылать данные в обход процессора. Например, вы сможете получать изображение по SPI и сразу выводить его на дисплей. Процессор при этом останется разгруженным.

Пины и интерфейсы

Миниатюрная плата содержит целых 3 UART'а, 2 SPI и по одному интерфейсу I²C и CAN. Каналов АЦП здесь столько, что можно сделать анализатор параллельных интерфейсов, суммарно — 21 аналоговый вход. 12 каналов ШИМ дают свободу в подключении большого количества периферии, управляемой модулируемым сигналом. Почти каждая ножка может работать как цифровой вход или выход. Суммарно на Teensy можно получить до 34 цифровых пинов. Все они толерантны к 5 вольтам.

Автономность

Одно из основных достоинств Teensy 3.1 — это заточенность под автономную работу. В режиме сна плата имеет потребление менее 0,25 мА. Напряжение питания процессора — 3,3 В, что позволяет питать плату от одного литиевого аккумулятора. В таком режиме ваше устройство может «проспать» несколько месяцев, а от трёх пальчиковых батареек — более года.

Программирование

Платформа поддерживает среду Arduino IDE 1.0.x, что сильно облегчает знакомство уже освоившим Arduino. Бóльшая часть библиотек Arduino работают и на Teensy, хотя это не гарантируется производителем.

Комплектация

Плата поставляется с нераспаянными колодками. Штырьковых разъёмов в комплекте нет. Вам предстоит добыть и припаять их самостоятельно.

Характеристики

- Процессор: Cypress MK20DX256
- Ядро: 32-х разрядный ARM Cortex-M4 72 МГц
- Оперативной памяти: 64 кБ
- Энергонезависимой памяти: 256 кБ Flash, 2 кБ EEPROM
- Количество линий ввода-вывода: до 34 (все толерантны к 5 В)
- Количество каналов АЦП: 21
- Количество каналов ШИМ: 12
- Количество UART: 3 шт
- Количество I²C: 2 шт
- Количество SPI: 1 шт
- Количество CAN: 1 шт
- Напряжение питания: 3,3–5,5 В
- Габаритные размеры: 17×35 мм

Netduino 2

amperka.ru
Амперка



Если вашим любимым языком программирования является C#, Netduino 2 позволит вам создать своё электронное устройство с комфортом. Эта плата подойдёт и тем, кто только начинает свой путь в программировании. Мощная среда программирования Microsoft Visual Studio поможет новичкам не ошибиться в трудной ситуации, а благодаря отладочному режиму с поддержкой breakpoint'ов и watch'ей вы сможете проследить исполнение своей программы шаг за шагом.

Netduino 2 совместима с большинством плат расширения Arduino и сочетает в себе вычислительную мощь и удобство разработки.

Процессор и память

Плата оснащена 32-разрядным ARM-микроконтроллером STM32F205RF с ядром Cortex-M3, частотой 120 МГц, 60 кБ оперативной памяти и 192 кБ энергонезависимой Flash-памяти.

Входы/выходы

22 пина, расположенных в стандартной для Arduino компоновке могут быть использованы как цифровые входы или выходы.

Цифровые входы и выходы рассчитаны на работу с напряжением 3,3 В. Однако, они толерантны и к напряжению 5 В.

Плата содержит в себе 4 последовательных порта, по одному интерфейсу I^2C и SPI. Для работы с аналоговым сигналом присутствуют 6 аналоговых входов с разрешением в 12-бит (4096 градаций). 6 выходов поддерживают ШИМ.

На плате присутствуют встроенная кнопка и два светодиода. Кнопка используется для сброса микроконтроллера. При желании вы можете использовать её в своём проекте в качестве элемента управления. Белый светодиод PWR используется как индикатор питания. Синий светодиод LED вы можете использовать в своём проекте для произвольных нужд.

Программирование

Плата программируется при помощи традиционной для .NET разработки среды Microsoft Visual Studio. Все необходимое для начала работы с Netduino 2 вы можете узнать из статьи на вики.

Характеристики

- Процессор: STM32F205RF
- Ядро: 32-х разрядный ARM Cortex-M3 120 МГц
- Оперативной памяти: 60 кБ
- Энергонезависимой памяти: 192 кБ Flash, 4 кБ EEPROM
- Напряжение питания микроконтроллера: 3,3 В
- Количество линий ввода-вывода: 22 (все толерантны к 5 В. Подтяжка к земле или питанию, толерантна к 3,6 В)
- Максимальный ток через контакт ввода-вывода: 25 мА
- Максимальный ток через все контакты микроконтроллера: 125 мА
- Количество каналов АЦП: 6
- Количество каналов ШИМ: 6
- Количество UART: 4 шт. (один из них с поддержкой RTS/CTS)
- Количество I^2C: 1 шт.
- Количество SPI: 1 шт.
- Напряжения питания платы: 5 В, 7,5–9 В

GPRS Shield v2

amperka.ru

Амперка



GPRS Shield — это плата расширения, позволяющая Arduino работать в сетях сотовой связи по технологиям GSM/GPRS для приёма и передачи данных, SMS и голосовой связи.

Плата построена на базе модуля SIMCom SIM900. Также на ней расположены: слот для SIM-карты, стандартное совмещённое 3,5 мм гнездо для аудио-входа и выхода, гнездо для батарейки CR1225, обеспечивающей работу встроенных часов реального времени, и разъём для выносной внешней антенны. Общение с платой производится через serial-соединение с помощью набора AT-команд

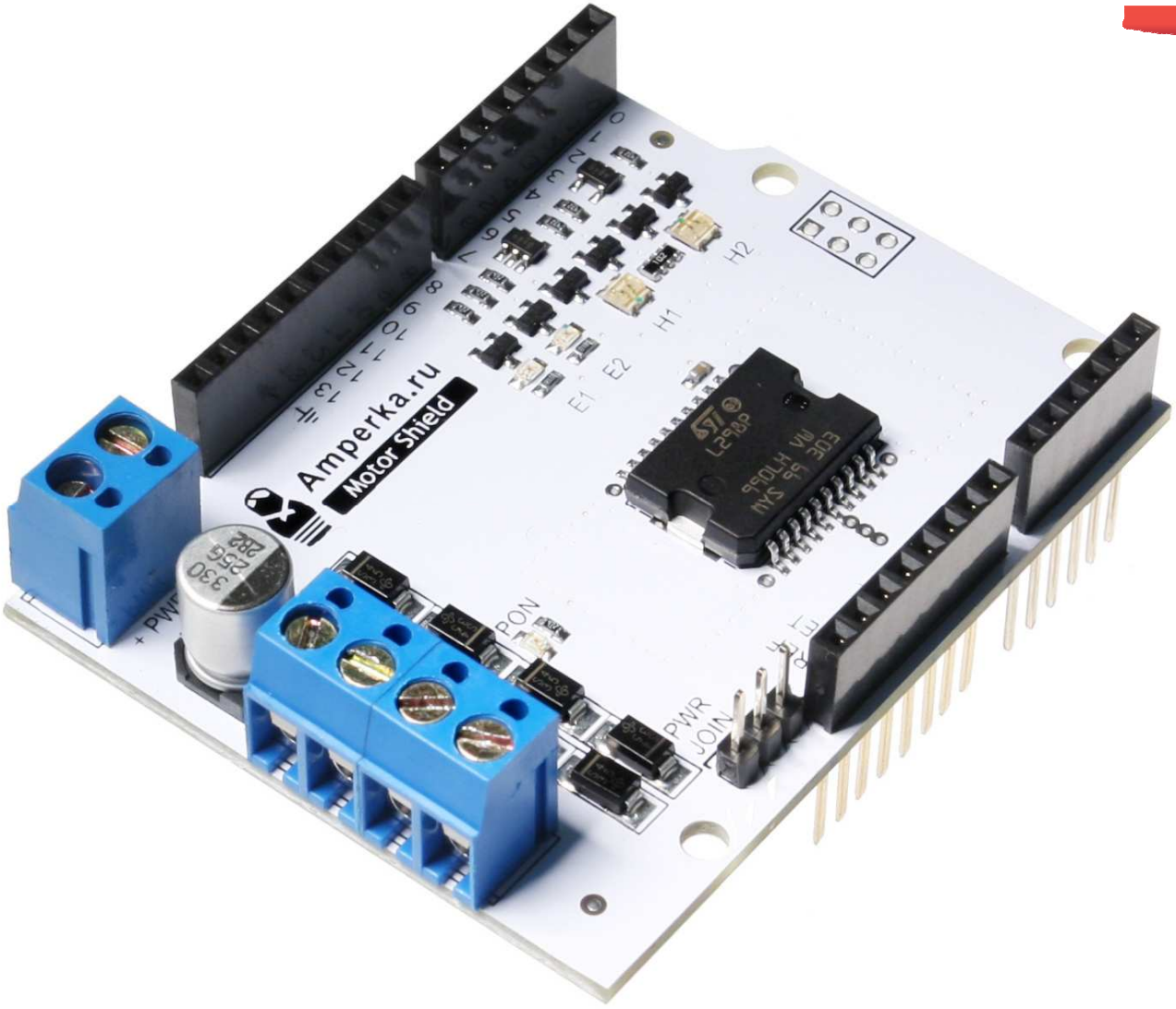
Характеристики

- Количество полос — 4: 850, 900, 1800, 1900 МГц
- Класс GSM — «2+»
- Класс GPRS — «B»
- Multislot class — 8/10

Motor Shield (2 канала, 2 А)

amperka.ru

Амперка



Хотите подключить мотор к своей Arduino? Это не совсем тривиально: выводы микроконтроллера являются слаботочными, поэтому ток мотора, при подключении его напрямую, выведет их из строя. Эту проблему решает так называемый H-мост. Он позволяет управлять скоростью и направлением вращения мотора с помощью логических сигналов микроконтроллера.

На сегодняшний день, самым популярным H-мостом, является микросхема L298P. Motor Shield — плата расширения для Arduino на базе чипа L298P, позволяющая управлять моторами с напряжением 5–24 В в режиме раздельного питания и 7–12 В в режиме объединённого питания.

Плата имеет 2 независимых канала. Используя их, можно подключить на выбор:

- Пару DC-моторов
- Один двухфазный шаговый мотор.
- Один DC-мотор с током до 4 А, если объединить каналы

Выходы под каждый из двигателей выполнены в виде клеммника с винтом, поэтому пайка не требуется.

При разгоне и торможении двигатели сами индуцируют кратковременный обратный ток большой величины, который может выжечь контакты микроконтроллера. На Motor Shield установлены возвратные диоды, которые это предотвращают.

На Motor Shield установлен комплект сквозных колодок Arduino Rev3, что означает возможность свободной установки других плат расширения, использующих незанятые пины. Однако, вы не сможете набрать этажерку из нескольких Motor Shield'ов для независимого управления большим числом двигателей: все платы будут работать параллельно, т.к. используют одни и те же пины. Но вы можете откусить или отогнуть управляющие ножки и перекинуть их на незанятые пины, чтобы добиться независимой работы.

На плате предусмотрена возможность выбора источника питания: от платы Arduino или от внешнего источника, подключаемого к клеммам «+» и «−». По умолчанию выбран раздельный режим, но переставив джампер, вы связываете контуры питания Arduino и Motor Shield и вам достаточно питать только одну из плат.

На плате расположены светодиоды-индикаторы, показывающие направление и скорость по каждому из каналов и подачу питания.

Для управления шасси с 4 моторами не нужно иметь два Motor Shield'а: вы можете подключить левую пару к одному каналу, а правую — ко второму. Ведь моторы на одной стороне должны работать синхронно.

Распиновка

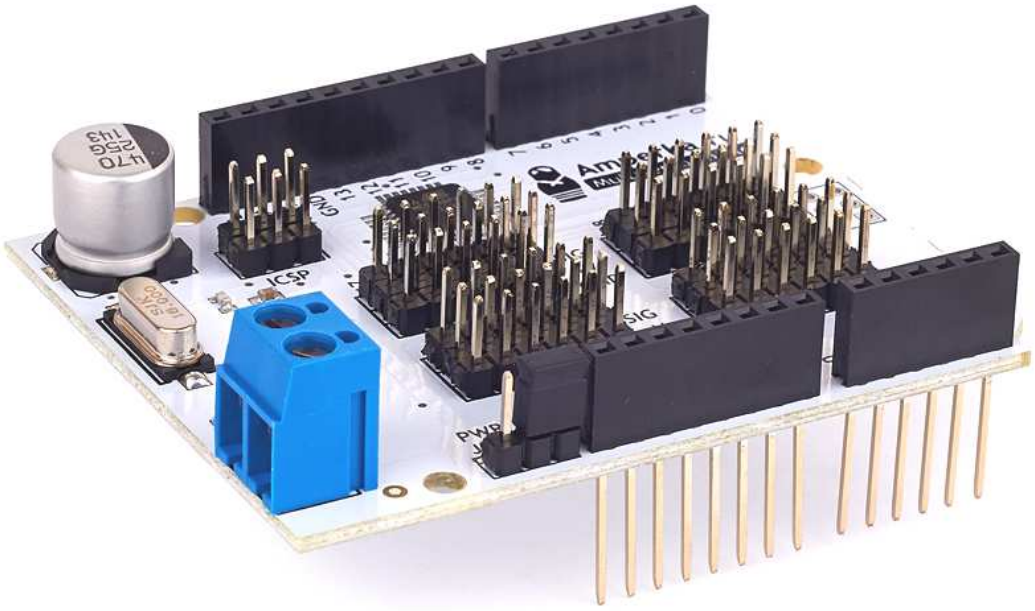
Для коммуникации с микроконтроллером используются цифровые контакты Arduino:

- 4 — направление, правый
- 5 — скорость (ШИМ), правый
- 6 — скорость (ШИМ), левый
- 7 — направление, левый

Multiservo Shield

amperka.ru

Амперка



Хотите управлять большим количеством сервоприводов одновременно? Воспользуйтесь Multiservo Shield. При помощи Multiservo Shield можно собирать многосуставных роботов, руки-манипуляторы и другие интересные вещи.

Установив эту плату расширения на Arduino, вы сможете управлять 18-ю сервоприводами. На борту Multiservo Shield установлен отдельный микроконтроллер ATmega48, все силы которого направлены на то, чтобы аккуратно и точно в нужное время передавать управляющие сигналы на подключённые сервоприводы.

Это позволяет избежать подёргиваний приводов в произвольные моменты времени, как это происходит при использовании стандартной библиотеки Servo.

Общение с Arduino осуществляется по шине I^2C. То есть всего через 2 пина. Поэтому даже при одновременном рулении 18-ю сервомашинками практически все пины Arduino останутся свободными.

Питание для двигателей подводится через винтовой клеммник. От клеммника питание по толстым дорожкам подходит к соответствующим пинам в тройках-контактах. Вам остаётся лишь подключить свои сервоприводы к этим контактам: дополнительной разводки не требуется.

Питание микроконтроллера ATmega48 берётся с пина 5V от Arduino. Вы можете установить специальную перемычку в положение «PWR JOIN», чтобы отправить питание с винтового клеммника ещё и на Vin Arduino. Таким образом, можно обойтись единственным источником питания.

Благодаря библиотеке Multiservo, работа с модулем не вызовет затруднений. Интерфейс библиотеки Multiservo тот же, что и в стандартной библиотеке Servo.

В дополнение к 18-ти серво-тройкам, управляемым выделенным микроконтроллером, на плату также вынесены 6 выводов Arduino напрямую. Таким образом, возможное количество сервоприводов в вашем устройстве может достигать 24-х штук.

Характеристики

- Максимальный постоянный ток на сервы: 10 А
- Интерфейс управления: I^2C
- Количество сервоприводов: 18 по I^2C, 6 через Arduino
- Потребляемый микроконтроллером ток: 15 мА
- Диапазон рабочих температур: −40…+85 °C

Music Shield v2

amperka.ru

Амперка



Music Shield — плата расширения с аудио-кодеком на базе чипа VS1053b даёт возможность Arduino проигрывать звуковые файлы в форматах MP3, WAV, MIDI, Ogg Vorbis. Файлы считываются с microSD флэш-карты, которая устанавливается в слот на плате. Для Arduino Mega доступна также запись в формате Ogg Vorbis.

На Music Shield помимо всего прочего расположены стандартный совмещённый 3,5 мм джек для линейного входа и выхода, пара светодиодов для отражения состояния, джойстик-манипулятор. Светодиоды и джойстик предназначены по задумке для управления воспроизведением, но это жёстко не регламентируется: вы можете использовать их для произвольных целей или не использовать вовсе.

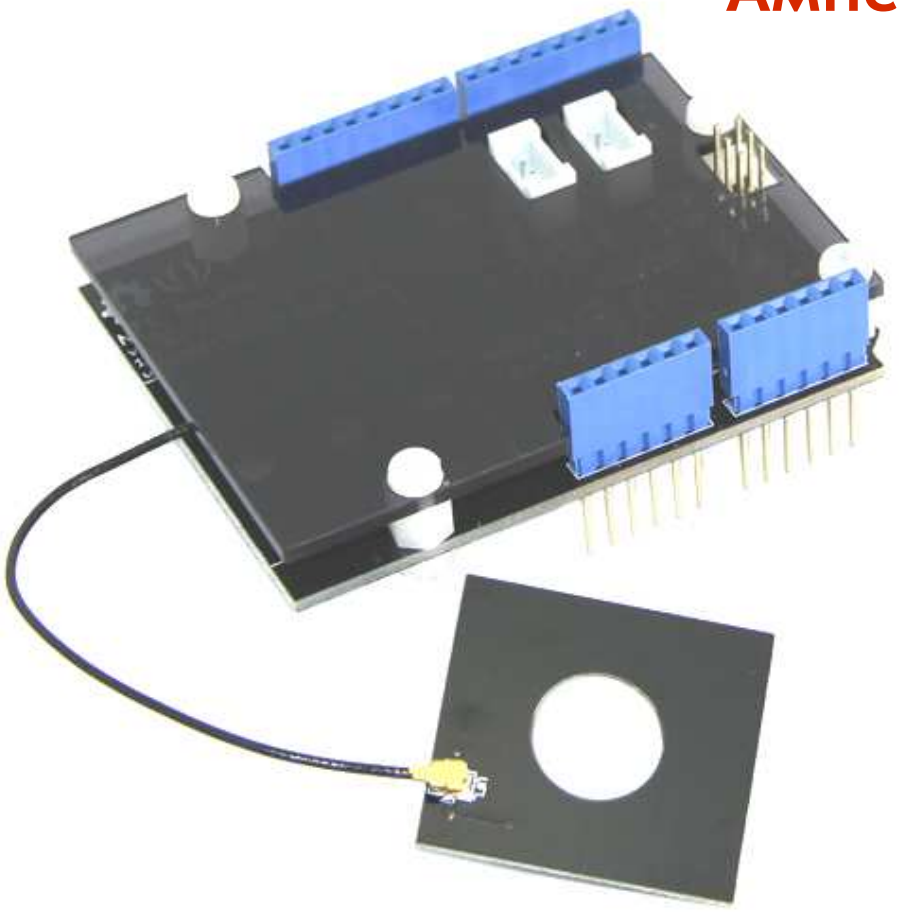
Линейный выход может раскачивать наушники и колонки сопротивлением 16 или 32 Ом. В эту категорию попадает бóльшая часть любительской аппаратуры.

Управление проигрыванием происходит через шину SPI.

NFC Shield v2

amperka.ru

Амперка



NFC Shield — плата расширения для Arduino, которая даёт возможность считывать и записывать данные на 13,56 мегагерцовые RFID карты и метки, обмениваться данными с другими устройствами, поддерживающими NFC (Near Field Communication). Поддержкой NFC обладает ряд современных смартфонов.

Плата построена на базе модуля PN532 и обладает выносной антенной, которая может быть закреплена либо на корпусе самой платы, либо на корпусе вашего устройства. Для коммуникации с Arduino используется шина SPI.

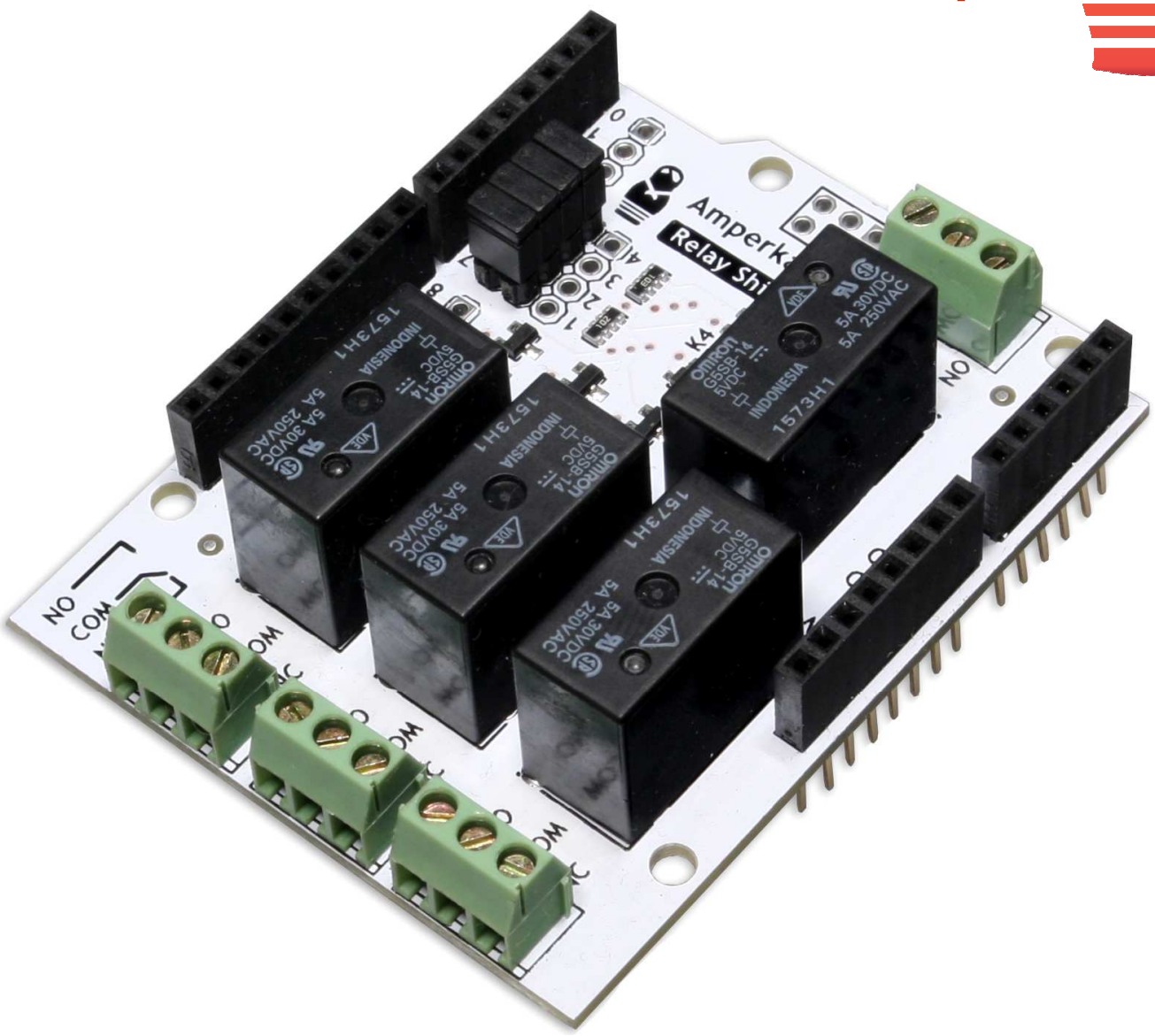
NFC Shield обеспечивает поддержку протоколов ISO14443 Type A/B.

Характеристики

- Расстояние срабатывания: до 5 см
- Напряжение питания: 3,7 – 5 В
- Потребляемый ток: 100 – 150 мА

Relay Shield (4 канала по 5 А)

amperka.ru
Амперка



В вашем проекте нужно замыкать и размыкать цепь с бытовым напряжением 220 В? Воспользуйтесь Relay Shield.

По своей сути реле — это просто механический рубильник, которым можно управлять при помощи микроконтроллера, такого как Arduino. При этом электрическая связь между управляющей электроникой и коммутируемой нагрузкой отсутствует. А это значит, что никакие помехи не испортят ваши планы. С помощью реле вы можете включать и выключать электроприборы, которые подключены к бытовой электросети 220 В. Без реле не обойтись при обустройстве «умного дома». Реле также может пригодится вам и для управления какими-нибудь устройствами с кнопками. Например, для автоматического набора номера на доисторическом настольном телефоне, достаточно включить реле параллельно кнопкам.

Relay Shield — плата расширения для Arduino, на которой расположены 4 независимых реле G5SB-14, подключённых к цифровым пинам Arduino. О текущем состоянии каждого из них можно судить по индикаторным светодиодам, расположенным на плате. К каждому реле подведён клеммник на 3 провода, что позволяет использовать реле как в режиме «нормально разомкнутое», так и в режиме «нормально замкнутое». Это удобно в ряде случаев. Например, если нужно что-то включать при обесточивании микроконтроллера.

Обмотки реле подключены к пинам Arduino с 4-го по 7-й через джамперы. Подача логического сигнала на эти пины замыкает или размыкает соответствующее реле. Если эти контакты уже используются — ничего страшного. На такой случай в Relay Shield мы предусмотрели специальные лужёные отверстия. Просто снимите джампер и подключите реле к любому другому цифровому контакту. Управление реле производится через встроенный ключ, что даёт вам возможность использовать в качестве управляющего сигнала любое напряжение от 3 до 5 вольт. То есть, в качестве основной платы подойдёт любая модель Arduino.

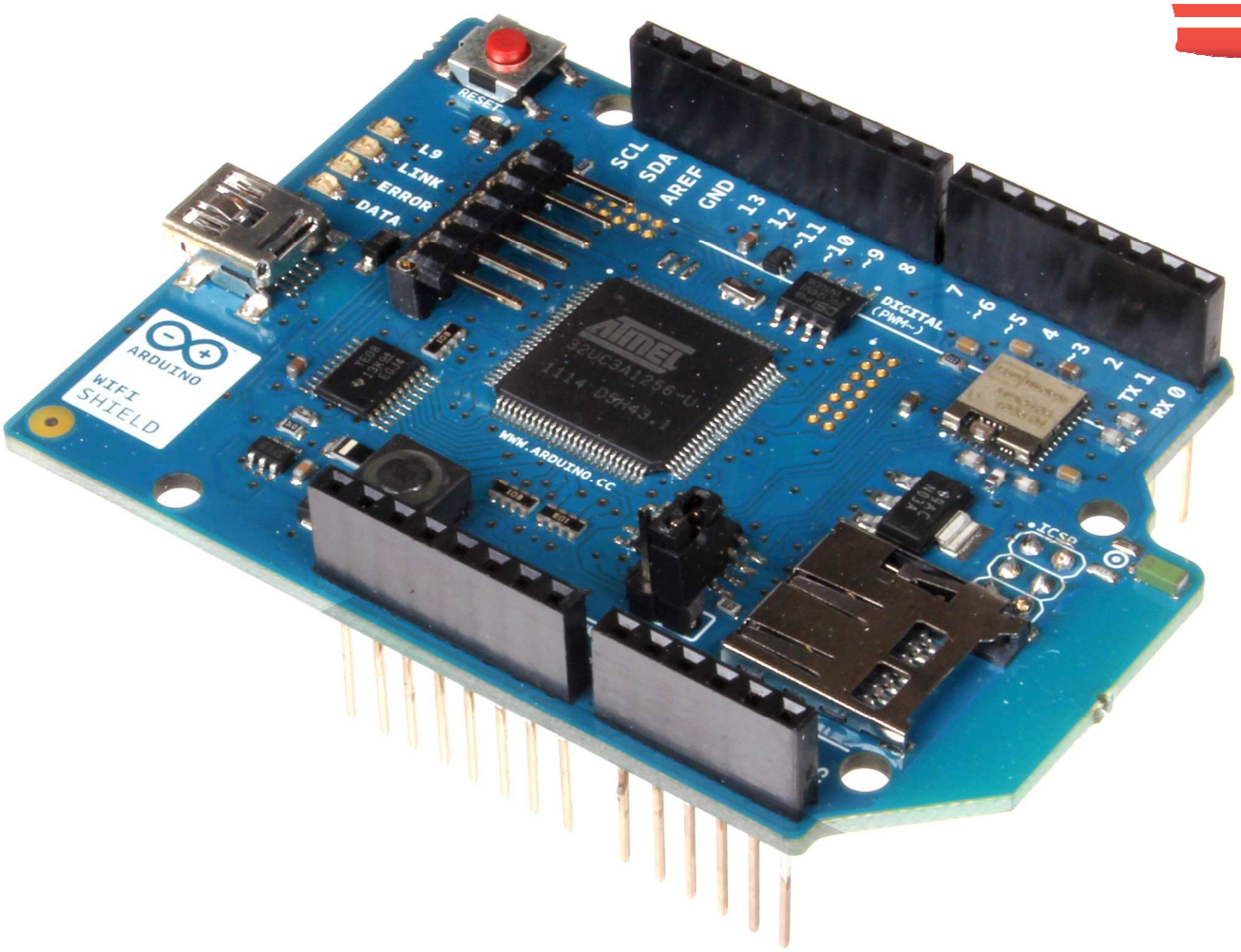
Характеристики реле

- Ток обмотки: 80 мА
- Максимальное коммутируемое напряжение: 30 В постоянного тока; 250 В переменного тока
- Максимальный коммутируемый ток: 5 А (NO), 3 А (NC)
- Рекомендованная частота переключения: до 1 Гц
- Время жизни: не менее 50□000 переключений

Wi-Fi Shield

amperka.ru

Амперка



Wi-Fi Shield — плата расширения для Arduino, которая даёт возможность организовать беспроводное соединение по стандарту 802.11 b/g (Wi-Fi) для общения с другими устройствами или выхода в интернет. Поддерживается шифрование WEP и WPA2 Personal.

Плата построена на базе модуля HDG104 и собственного 32-битного микроконтроллера ATmega UC3, который предоставляет IP-стек, дающий возможность использовать протоколы TCP и UDP.

На Wi-Fi Shield расположен слот для флеш-карт microSD объёмом до 2 Гб. Вы можете использовать эту возможность для протоколирования данных или хранения раздаваемых медиа-ресурсов.

Занимаемые пины

Для коммуникации с Arduino используется шина SPI. Занятыми оказываются пины:

- MOSI, MISO, CLK (11, 12, 13 на Arduino Uno)
- 10-й пин, который выполняет роль чип-селектора (CS) для Wi-Fi
- 4-й пин, который является чип-селектором для SD. Он используется только, если вы используете флеш-карту
- 7-й пин используется для «рукопожатия» между Arduino и Wi-Fi Shield'ом. Он не должен использоваться для других целей

Совместимость

Другие платы расширения, работающие через SPI могут работать совместно с Wi-Fi Shield за исключением случаев, когда они также используют те же чип-селекторы или 7-й пин.

Сама по себе плата работает от напряжения в 5 В. Однако на ней расположен преобразователь уровней сигналов, который делает Wi-Fi Shield совместимым с платами, работающими на 3,3 В. Такими как, например, Arduino Due.

Программирование

Для работы через Wi-Fi из программы вы можете использовать стандартную библиотеку Wi-Fi. Пример использования можно найти на официальном сайте.

Для отладки и перепрошивки модуля на плате выведен FTDI-совместимый разъём. Вы можете использовать USB-Serial адаптер и этот разъём для подключения через USB. Инструкция по отладке и перепрошивке доступны на официальном сайте.

Для прошивки бортовой ATmega в режиме DFU на плате расширения доступен разъём mini-USB и джампер, переводящий плату в DFU.

Индикация

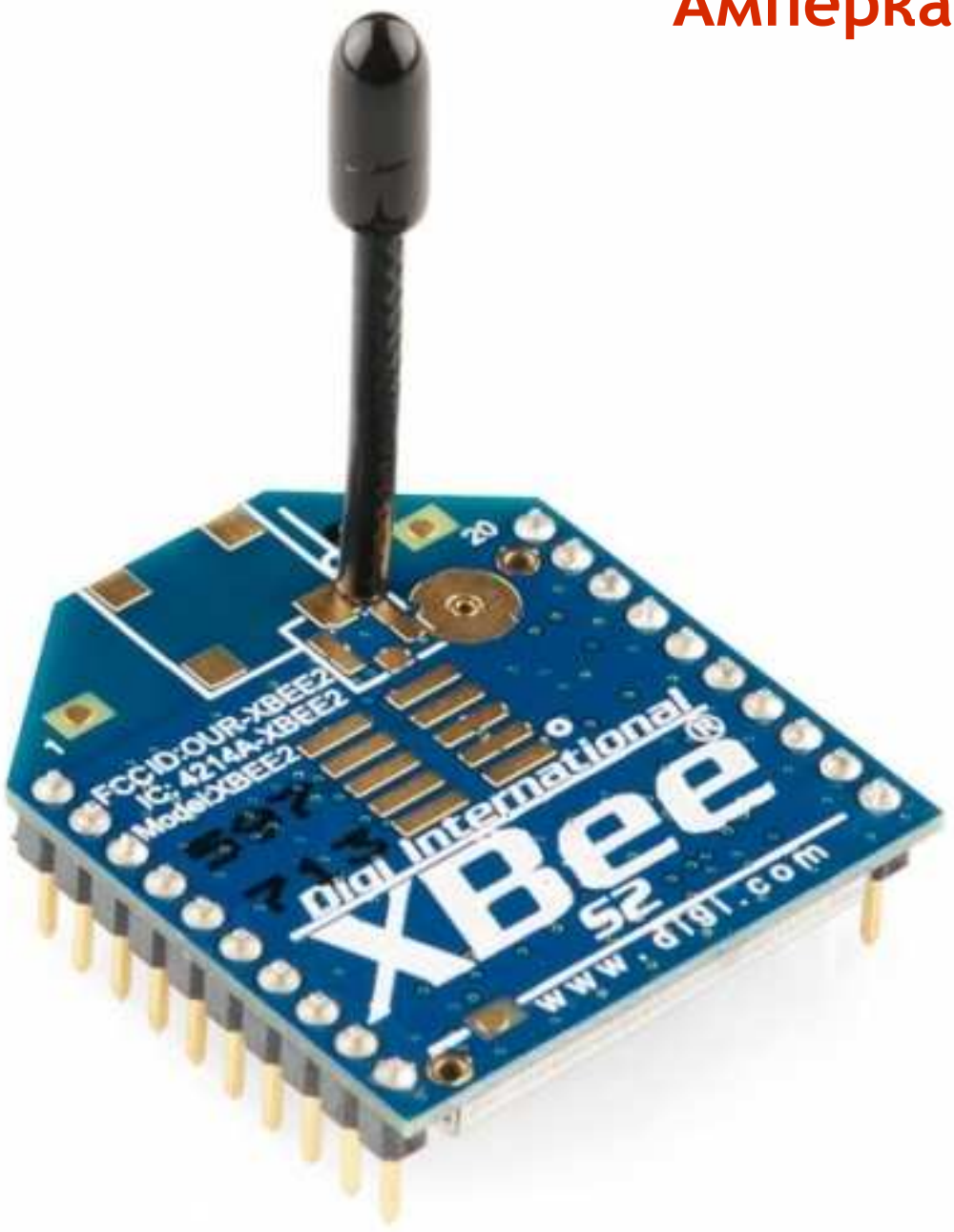
На плате расположены 4 светодиода:

- L9 (жёлтый) — привязан к 9-му пину. Используйте его по своему усмотрению
- LINK (зелёный) — сигнализирует о подключении к сети
- ERROR (красный) — сигнализирует об ошибке коммуникации
- DATA (синий) — сигнализирует передачу данных

XBee

amperka.ru

Амперка



XBee — это модуль, дающий вашему устройству возможность использования протокола ZigBee. Это стандарт беспроводной передачи данных вроде Wi-Fi и Bluetooth, но ориентированный на экономию электроэнергии и большую защищённость канала при меньшей скорости.

Мощности передатчика хватает для общения на расстоянии до 120 м на улице и до 35 м в помещении. Скорость обмена данными: до 250 кбит/с. Устройство работает на частоте 2,4 ГГц. Возможны как простые соединения «точка-точка», так и сети со сложной топологией.

Bluetooth Bee

amperka.ru

Амперка



Модуль Bluetooth производства DFRobot, выполненный на чипе CSR BC417143, даёт возможность взаимодействовать с другими устройствами по протоколу bluetooth 2.0.

С точки зрения программирования bluetooth-модуль видится как прозрачный serial-порт. Это позволяет общаться, в том числе, с мобильными телефонами и ноутбуками.

Wi-Fi Bee v2

amperka.ru

Амперка



Вашему устройству требуется беспроводной выход в интернет или локальную сеть? Установите на него Wi-Fi Bee модуль и забудьте об этой проблеме.

С помощью него вы можете через интернет управлять вашим роботом или собирать информацию с датчиков, открывать и закрывать двери, включать и выключать устройства. А благодаря встроенным часам реального времени и возможности автоматической установки времени через интернет, вы можете сделать точные часы, не требующие подстройки.

Модуль выполнен в форм-факторе XBee. Для подключения к Arduino будет удобно использовать Wireless Shield или Mega IO Shield. Так же вы без труда сможете сделать робота по Wi-Fi, если воспользуетесь платформой Strela.

Антенна модуля — встроенная, она размещена на корпусе.

Модуль имеет встроенную поддержку сетевых протоколов. А это значит, что вам не придётся реализовывать их самостоятельно — всё уже готово. Модуль управляется AT-командами. Для удобства работы с Arduino существует библиотека WiFly.

Модуль может быть полезен как при работе с Arduino, так и в качестве отдельного устройства, питающегося от батареи. Для этого на плате установлен собственный DC-DC преобразователь. Допустимое входное напряжение в этом случае может быть от 2 до 3 вольт.

Этот модуль имеет 8 аналговых входов, которые могут опрашивать датчики и отправлять считанное 14-битное значение на удалённый web-сервер. Диапазон измерения аналогового сигнала: 0–1,2 В.

Поддерживаемые стандарты шифрования

- WEP-128 (только «open mode»)
- WPA2-PSK (только AES)
- WPA1-PSK (только TKIP)

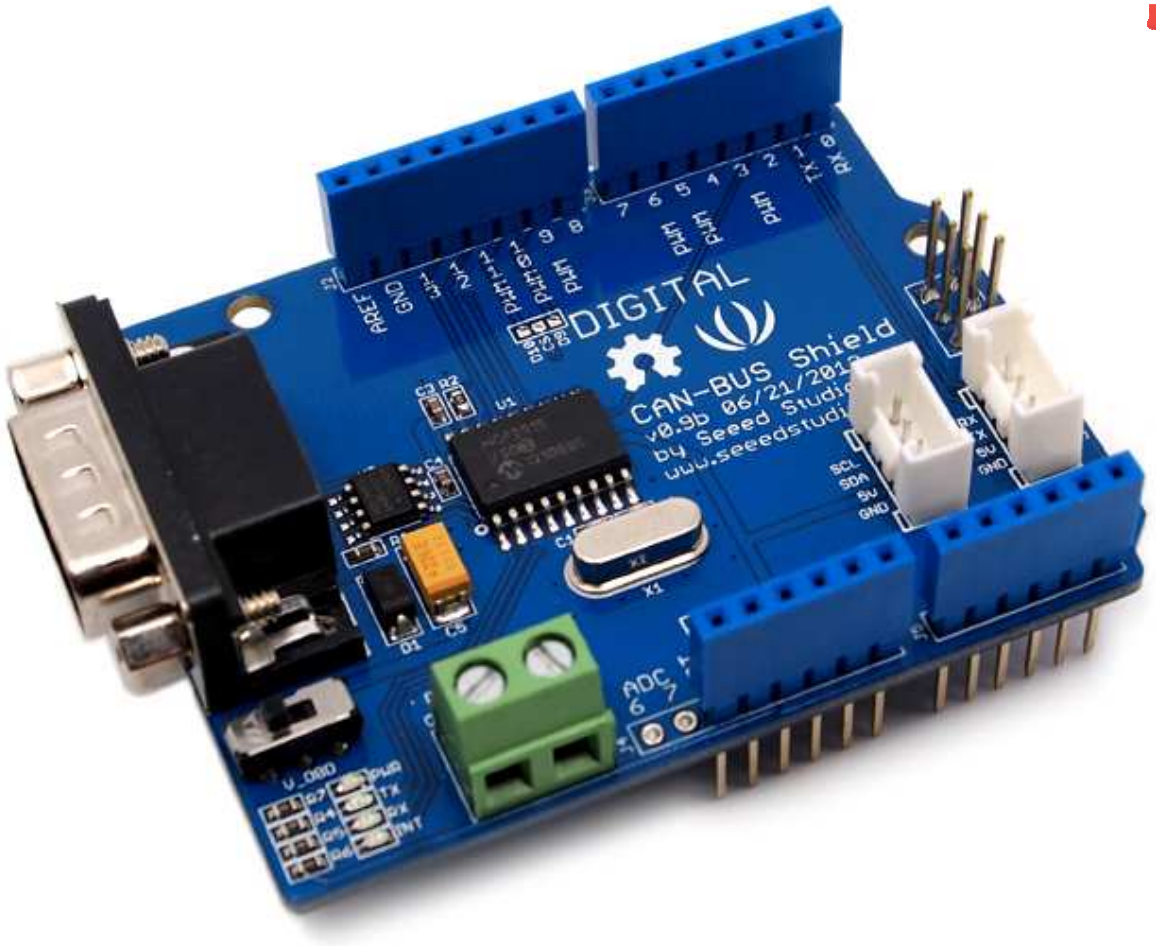
Характеристики

- Мощность передатчика: 0–12 дБ
- Диапазон частот: 2402–2480 МГц
- Поддерживаемые каналы: 0–13
- Номинальное напряжение питания: 3,3 В
- Напряжение питания от батареи: 2–3 В

CAN-BUS Shield

amperka.ru

Амперка



Вам не хватает дальности SPI и I²C, а RS-232 слишком медленный и ненадёжный? Тогда, вам стоит присмотреться к шине CAN. Этот индустриальный стандарт широко распространён в промышленной автоматизации, технологиях умного дома, автомобильной промышленности и других областях.

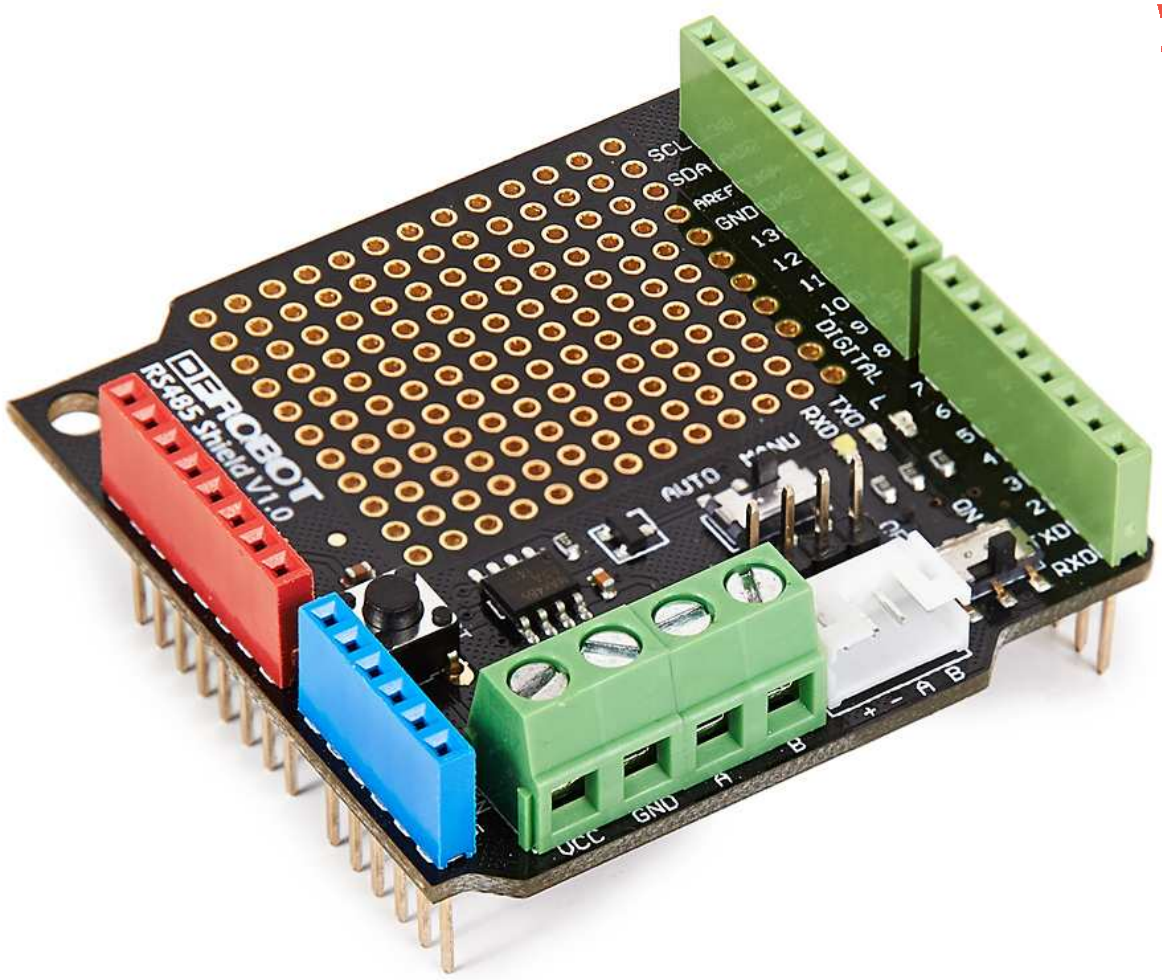
Существует одна проблема: микроконтроллеры AVR, являющиеся основой плат Arduino, не поддерживают данную шину. Поэтому необходимо использовать дополнительную обвязку микроконтроллера, аппаратно реализующую стандарт CAN. CAN-BUS Shield — плата расширения для Arduino, разработанная на основе микросхемы MCP2515.

Данная плата, по сути, представляет собой конвертер SPI — CAN.

RS-485 Shield

amperka.ru

Амперка



Хотите дёшево и просто объединить несколько плат Arduino в сеть? Достаточно установить на каждую плату RS-485 и протянуть всего два провода!

Вы сможете разворачивать масштабные сети интеллектуальных датчиков, строить системы умного дома или организовывать автоматический полив в большом количестве мест вашего дачного участка.

Никаких витых пар. Никаких обжимок и разъёмов 8P8C. Никаких концентраторов. Только два провода, к которым параллельно подключаются все контроллеры. Модуль предоставляет возможность передавать байты данных. Протокол передачи данных предоставляется определить самостоятельно.

Кроме того, по RS-485 вы можете общаться со сторонними устройствами, которые поддерживают этот интерфейс.

Компоновка платы

Плата построена на микросхеме Maxim MAX485 и производится компанией DFRobot.

Микросхема MAX485, на которой основан шилд, имеет набор средств защиты от внешний воздействий: защита от короткого замыкания линий, защита от перегрева и защита от электростатических разрядов.

На плате есть площадка с отверстиями для прототипирования. Шаг отверстий стандартный — 2,54 мм.

Особенности интерфейса

RS-485 — симплексный интерфейс. Это означает, что в любой момент времени данные могут передаваться только одним узлом сети. Причём данные передаются сразу по двум проводам дифференциальным методом: передаваемый бит кодируется разностью потенциалов между двумя проводами линии. Такой метод передачи существенно снижает уровень помех, т.к. они наводятся одинаково на оба провода и «не видны» при измерении разности потенциалов на этих проводах. Как результат, устройства могут общаться на расстоянии до 1200 метров.

Платы расширения не имеют опторазвязки, однако, могут составлять сеть, как с единственным, так и с множеством источников питания. Данные передаются дифференциально, поэтому гальваническая развязка узлов питания не нужна.

Характеристики

- Максимальное расстояние связи: 1200 м
- Максимальное количество узлов в сети: 32
- Напряжение питания: 5 В

РадиоЛоцман.Arduino

www.rlocman.ru/arduino

Широкий взгляд на тему :

- Последние новости
- Уникальные статьи
- Периодика и книги
- Сообщество на форуме
- Поиск товаров
- Datasheets



И еще многое другое на портале :

РадиоЛоцман. ...

www.rlocman.ru

Как создавали и продвигали Arduino

David Kushner

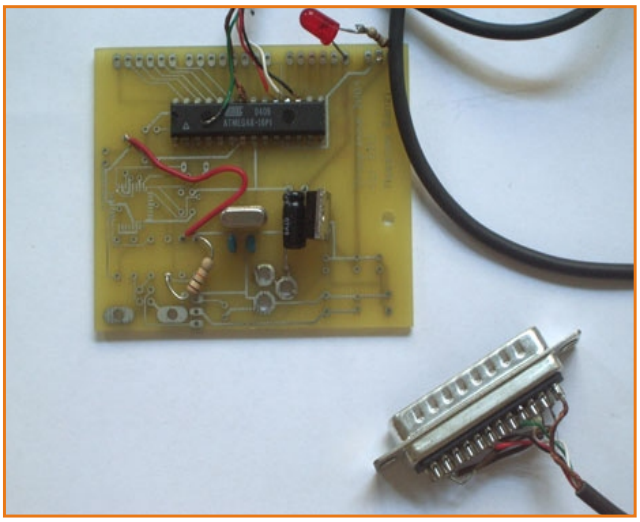
IEEE Spectrum

История о том, как пятеро друзей создали маленькую плату, которая взяла штурмом мир электронных самоделок

Живописный город Ивреа, стоящий на реке Дора Балтея в северной Италии, известен своими королями-неудачниками. В 1002 году король Ардуин стал правителем страны, а через два года был свергнут королем Германии Генри II. Сегодня бар ди Ре Ардуино, расположенный на булыжной улице этого городка, назван в честь короля и стоит на том самом месте, где, как верят некоторые, родился король.

Бар является пивнушкой Массимо Банци (Massimo Banzi), итальянского соучредителя проекта в сфере электроники, который был назван в честь этого места. Ардуино – недорогая микроконтроллерная плата, которая позволяет даже новичку делать действительно потрясающие вещи. Вы можете подсоединить к Arduino различные типы датчиков, источников света, электромоторов и множество прочих устройств, и использовать легко осваиваемое программное обеспечение, чтобы запрограммировать поведение вашего творения. Вы можете создать интерактивный дисплей или движущегося робота, и затем поделиться вашим проектом с миром, поместив информацию о нем в сеть.

Выпущенный в 2005 году как скромный инструмент для студентов Банци в Институте проектирования взаимодействий города Ивреа (Interaction Design Institute Ivrea, IDII), Arduino породил международную революцию в сфере международных электронных самоделок. Вы можете купить эту плату всего за $30 или собрать ее с нуля. Все схемы и исходные коды доступны бесплатно на условиях открытых лицензий. В результате Arduino стал самой влиятельной аппаратной системой своего времени с открытым исходным кодом.



Первая плата прототипа, сделанная в 2005 году, имела простейший дизайн и еще не называлась Arduino. Немного позже, в том же году, Массимо Банци придумал ей имя (Фото: Массимо Банци)



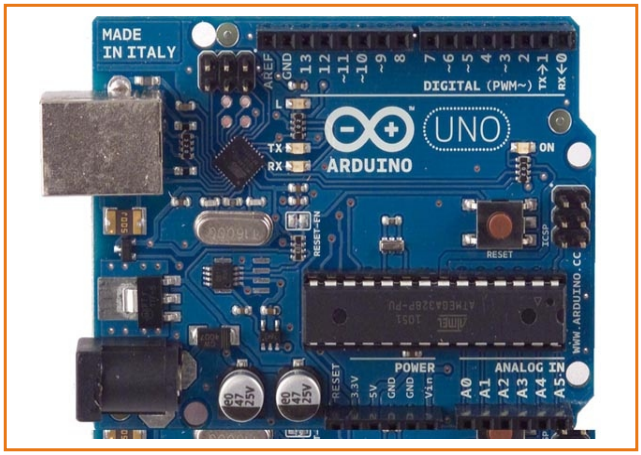
Ядро команды Arduino (слева направо): Дэвид Куартилльз (David Cuartielles), Джанлука Мартино(Gianluca Martino), Том Иго (Tom Igoe), Дэвид Мелис (David Mellis), и Массимо Банци (Massimo Banzi) на конференции Maker Faire в Нью-Йорке (Фото: Рэнди Зильберман Клетт)

Маленькая печатная плата теперь является, своего рода, источником вдохновения для «художников» в электронике, людей, увлеченных электронными поделками, студентов и любого, у кого есть мечта собрать что-нибудь этакое. Во всем мире продано более 250,000 комплектов Arduino, и это не учитывая множества клонов. «Это позволило людям, делать вещи, которые они не сделали бы каким-либо иным способом» – говорит Дэвид А. Мелис (David A. Mellis), который прежде был студентом IDII до того, как выполнил квалификационную работу в Медиа лаборатории Массачусетского технического института, и теперь является ведущим разработчиком программного обеспечения для Arduino.

Сегодня имеется множество интересных разработок на базе Arduino, таких как алкотестеры, светодиодные кубы, системы домашней автоматизации, дисплеи отображения Twitter-сообщений и, даже, наборы для анализа ДНК. Уже появились целые клубы и сообщества любителей Arduino. Google недавно выпустил основанный на Arduino комплект разработчика для своего смартфона на базе Android. Как высказался на этот счет Дэйл Доэрти (Dale Dougherty), редактор и издатель журнала «Make», библии любителей самоделок, Arduino стал «мозгами создателей проектов».

Но Arduino – это не только общедоступный проект, стремящийся сделать технологии более доступными, это также стартаповая компания, которой руководит Банци и небольшая группа его друзей. И эта компания столкнулась с проблемой, которую не может решить даже их волшебное изделие – как пережить нахлынувший успех и вырасти. «Мы должны сделать следующий скачок», – говорит Банци, – «и стать крупной компанией».

В свою очередь, когда-то Arduino вырос из другой крупной проблемы – как научить студентов создавать электронные устройства, причем быстро. Это был 2002 год, и Банци,



бородатый и добродушный разработчик программного обеспечения, был взят на работу в IDII в качестве доцента для продвижения новых способов разработки интерактивного проекта – новой области, также известной как физические вычисления. Но с урезанным бюджетом и ограниченным по времени доступом к лабораторной базе, его возможности использования необходимых приборов были небольшими.

Как и многие из его коллег, Банци полагался на BASIC Stamp, микроконтроллер, разработанный Калифорнийской компанией Parallax, который инженеры использовали в течение приблизительно десятилетия. Программируемый посредством языка BASIC, Stamp, в целом, представлял собой небольшую печатную плату с размещенными на ней источником питания, микроконтроллером, памятью и портами ввода/вывода для соединения с различной аппаратурой. Но у BASIC Stamp были две проблемы, которые обнаружил Банци – у него не хватало достаточной вычислительной мощности для некоторых задуманных его студентами проектов, и он был достаточно дорогим – плата с основными дополнениями могла стоить около $100. Кроме того, Банци нуждался в чем-то, что могло бы работать на компьютерах Macintosh, которые были практически у каждого разработчика в IDII. Что, если они сами смогли бы сделать плату, которая непосредственно удовлетворяла бы их потребностям?

У Банци был коллега из MIT, разработавший удобный язык программирования Processing. Processing быстро завоевывал популярность, поскольку позволял даже неопытным программистам создавать сложные, и в то же время красивые, системы визуализации данных. Одной из причин его успеха была чрезвычайно удобная в процессе работы интегрированная среда разработки (IDE). Банци задался вопросом, смогли бы они разработать подобный программный инструментарий, чтобы программировать микроконтроллер, вместо того, чтобы рисовать графику на экране.

Студент, обучавшийся у Банци, Эрнандо Барраган (Hernando Barragán), сделал в этом направлении первые шаги. Он создал платформу разработки, названную Wiring, которая включала в себя как дружественную пользователю IDE, так и готовую к использованию плату. Это был многообещающий проект, который продолжает развиваться по сей день, но Банци уже задумал более крупный. Он хотел создать платформу, которая была бы еще проще, дешевле и легче в использовании.

Банци и его сотрудники были сторонниками программного обеспечения с открытым исходным кодом. И с того момента, когда встала цель создать простую и легкодоступную платформу, они почувствовали, что будет лучше разработать открытую платформу, доступную для многих людей, нежели закрытую. Другим фактором, способствовавшим этому решению, было то, что после пяти лет работы IDII исчерпывал свои фонды и собирался закрываться. Преподаватели боялись, что их проекты не выживут или будут незаконно присвоены кем-либо. По этому поводу Банци вспоминает: «Итак, мы сказали: забудьте про это. Давайте же сделаем открытую платформу».

Новая модель с открытым исходным кодом довольно долго использовалась для развития программного обеспечения, но не аппаратного. Для того чтобы заставить ее работать должным образом, они должны были найти подходящее лицензионное решение, которое могло бы подойти к их плате. После некоторых

исследований они поняли, что если бы взглянули на свой проект немного под другим углом, то смогли бы использовать лицензию от Creative Commons, некоммерческой организации, соглашения которой обычно используются в сфере творческих работ, таких как музыка и литература. «Вы можете рассматривать аппаратное обеспечение как часть культуры, которой вы хотите поделиться с другими людьми» – говорит Банци.

Приступив к работе, группа разработчиков сразу задалась целью сделать устройство, цена которого была бы подходящей для студенческого кармана – $30. «Цена устройства должна быть эквивалентна стоимости обеда в пиццерии» – говорит Банци. Также они хотели сделать его необычным, в чем-то выделяющимся и классно смотрящимся для помешанных на электронике людей. В то время как обычные платы, чаще всего, имеют зеленый цвет, они решили сделать свою плату синей, там где другие производители экономят на количестве выводов печатной платы, они решили добавить их как можно больше. И в качестве последнего штриха они нанесли небольшую карту Италии на нижнюю сторону платы. «Здесь много проектных решений, которые выглядят, по меньшей мере, странно для настоящего инженера» – говорит со смехом Банци – «Но я не настоящий инженер, поэтому я избрал такой глупый путь».

Для одного из «настоящих» инженеров в команде, Джанлуки Мартино, такой нетрадиционный подход к проектированию печатной платы был поучителен. Мартино описывает его как «новый способ мышления об электронике». Он говорит: «Это не инженерный способ разработки, где вы должны учитывать каждый электрод, это подход самоделкина».

Продукт, который создала команда, состоял из дешевых компонентов, которые могли быть легко найдены на рынке электроники, например, микроконтроллер ATmega328. Но ключевое решение состояло в том, чтобы гарантировать работу устройства по принципу plug-and-play: чтобы пользователь, достав плату из коробки и подключив к компьютеру, мог немедленно приступить к работе. Такие платформы как BASIC Stamp для этих целей требовали от пользователя наличия еще полдюжины компонентов, которые включались в общую стоимость продукта, в то время как устройство команды Банци нуждалось лишь USB-кабеле, посредством которого оно соединялось с компьютером – Mac или PC.

«Философия Arduino состоит в том, что если вы захотите научиться электронике, вы сможете изучать ее уже с первого дня, вместо того, чтобы сначала учиться алгебре» – говорит другой член команды, инженер по телекоммуникациям, Дэвид Куартилльз.

И вскоре команда Банци проверила это утверждение на практике. Они вручили 300 пустых печатных плат студентам IDII с простым наставлением: ищите инструкции в интернете, разработайте свою собственную плату, и используйте ее для чего-нибудь. Одним из первых проектов был самодельный будильник, который свисал с потолка на кабеле. Суть идеи заключалась в том, что всякий раз, при нажатии кнопки сброса сигнала пробуждения, будильник поднимался по кабелю все выше, и так до тех пор, пока вы не встанете окончательно.

Вскоре об этой плате услышали другие люди. И они тоже захотели приобрести себе такую. Первым покупателем стал друг Банци, который заказал себе одну штуку. Так проект начал набирать обороты, но была упущена одна существенная вещь – у их изобретения не было названия. И вот однажды ночью, за

стопкой алкогольного напитка, в местном пабе идея и пришла к ним – «Arduino», – прямо как этот бар и король.

Слово «Arduino» быстро распространилось в Интернете без какой-либо рекламы и привлечения средств маркетинга. Вначале это привлекло внимание Тома Иго, профессора в области физических вычислений, занятого по программе телекоммуникационного взаимодействия в Нью-йоркском университете. Сегодня он также входит в ядро команды Arduino. Иго обучал студентов нетехнических специальностей, используя BASIC Stamp, но был впечатлен возможностями Arduino. «Они предполагали, что вы не знаете электронику и программирование, что вы не хотите разрабатывать целое устройство, но, в то же время, смогли бы запрограммировать простую микросхему. Вы просто можете открыть коробку с платой, нажать кнопку загрузки, и она заработает» – говорит Иго. «Кроме того, я был поражен ее стоимостью в $30, которая делает плату доступной. Для меня это было одним из ключевых факторов».

Своим успехом Arduino во многом обязан существовавшим до него Processing и Wiring. От этих проектов Arduino унаследовал одну сильную черту – удобную для пользователя среду разработки. До появления Arduino программирование микроконтроллеров сопровождалось сложным и рутинным обучением. С Arduino даже те, кто не имел опыта работы с электронными устройствами, теперь могут прикоснуться к ранее недоступному для них миру электроники. Теперь новичкам не нужно тратить много времени на изучение соответствующего материала, они могут быстро разработать прототип, который будет полноценно рабочим. Это мощный шаг вперед, в то время как некоторые довольно популярные гаджеты представляют собой «черные ящики», закрытые и защищенные патентами.

По мнению Банци, самым, вероятно, важным результатом воздействия Arduino стала демократизация процесса разработки. «Пятьдесят лет назад чтобы написать программное обеспечение вам требовалась команда людей в белых халатах, которые знали все об электронных лампах. Но теперь, даже моя мама может программировать» – утверждает Банци, – «Мы позволили множеству людей создавать электронные устройства самостоятельно».

Но не все инженеры любят Arduino. Наиболее придирчивые из них критикуют этот продукт за достаточно глупый, с точки зрения инжиниринга, подход к разработке и заполнение рынка любительской радиоэлектроники серым дилетантским товаром. Меллис, однако, не рассматривает новинку как угрозу обесценивания роли инженера вообще. Он говорит: «Имея платформу, которая позволяет проектировщику или любой другой творческой личности немножко приблизиться к своей цели, люди могут теперь упростить себе взаимодействие со специалистами и сказать «Вот то, что я хочу в итоге сделать». Я не думаю, что это замена инженера, это всего лишь облегчает сотрудничество».

Для того, чтобы поддерживать распространение Arduino, команда Банци рассматривает способы более глубокой интеграции своей платформы в систему образования, от начальных школ до колледжей. Несколько университетов, включая Карнеги Меллона и Стэнфорд, уже используют Arduino. Меллис изучал как студенты и дилетанты постигают электронику на семинарах в медиа лаборатории Массачусетского института. Меллис приглашает 8 – 10 человек в лабораторию, где им

дают задачу, которую необходимо выполнить в течение дня. Среди проектов были такие, как разработка спикеров для iPod, FM-радио и компьютерной мыши с использованием тех же компонентов, которые используются в Arduino.

Но распространение идеологии Arduino – это лишь часть проблемы. Команда не должна отставать от современных требований, предъявляемых к их платам. И теперь, фактически, платформа Arduino представлена не одной платой, а целым семейством. В дополнение к оригинальному проекту, названному Arduino Uno, новые модели, имеющие более мощные средства на плате, носят название Arduino Mega, более компактные называются Arduino Nano, в водонепроницаемом корпусе – LilyPad Arduino, и недавно выпущенная модель с поддержкой сетевого подключения – Arduino Ethernet.

Arduino также породил целую отрасль домашнего производства для любителей электроники. В настоящее время по всему миру насчитывается более двухсот дистрибьюторов продукции Arduino, от крупных компаний, таких как SparkFun Electronics или in Boulder, Colo., до компаний семейного бизнеса, работающих на местный рынок. Банци недавно узнал о человеке из Португалии, который уволился из телефонной компании для того, чтобы продавать продукцию Arduino прямо из дома. Член команды Arduino Джанлука Мартино, ответственный за производство и распространение, говорит, что они работают в сверхурочное время, чтобы удовлетворить потребности развивающихся рынков Китая, Индии и Южной Америки. По его словам, в настоящее время, около 80% продаж Arduino приходится на Северную Америку и Европу, тогда как остальные 20% приходятся на весь остальной мир.

Поскольку команда не может позволить себе снабжать рынки сотнями тысяч плат, ей приходится производить от 100 до 3000 устройств в день на различных фабриках близ их родного города Ивреа. Для каждой из плат команда разработала систему контроля выводов, количество которых для модели Uno, например, равно 26: 14 цифровых входов/выходов, 6 аналоговых входов, и еще 6 для подключения питания. Это большая проблема для обеспечения требуемого качества, с учетом того, что необходимо производить и проверять тысячи модулей в день. Но плата Arduino достаточно дешева, чтобы позволить команде держать обещание заменять любую плату, которая отказывается работать. По словам Мартино, доля отказов ниже 1 процента.



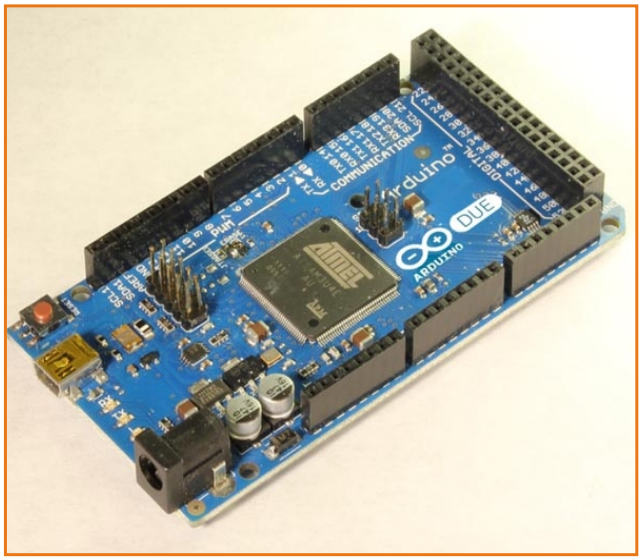
Команда Arduino заключила контракт с компанией, которая способна производить от 100 до 3000 плат в день на фабрике неподалеку от итальянского городка Ивреа (Фото: Массимо Банци)

Команда Arduino зарабатывает уже достаточно, чтобы содержать двух ее сотрудников в качестве постоянных работников, и уже планирует увеличение возможностей и производительности своей платы. На состоявшейся в

сентябре в Нью-Йорке конференции Maker Faire, спонсируемой журналом Make, команда представила свою первую плату с 32-разрядным ARM-процессором. Это будет ответом на требования поддержки более мощной периферии. Например, MakerBot Thing-O-Matic является набором для изготовления 3-D принтера на базе Arduino, но с использованием более быстрого процессора он получит больше преимуществ для решения более сложных задач.

Очередное ускорение проект получил в 2011 году, когда Google выпустил основанный на Arduino комплект разработчика для своей системы Android. Android ADK (вспомогательный комплект разработчика) от Google является платформой, позволяющей телефону с ОС Android взаимодействовать с двигателями, датчиками, и другими устройствами. Вы можете создать приложение для Android, которое использует камеру телефона, датчики движения, сенсорный экран, и интернет-связь, чтобы управлять, например, дисплеем или роботом. Энтузиасты говорят, что дополнительная поддержка Android открывает еще большие возможности для проектов Arduino.

Но к вопросу чрезмерного усложнения Arduino команда, все же, подходит с крайней осторожностью. «Проблема заключается в том, чтобы найти способ объединения на одной платформе всего множества вещей,



Недавно команда представила модель Arduino Due, плату с 32-разрядным процессором Cortex-M3 ARM, который предоставляет больше вычислительной мощности для разработчиков, занятых сложными проектами

которые хотят иметь люди», – говорит Меллис, – «но при этом не сделать продукт слишком сложным для новичков».

Сейчас они наслаждаются своей популярностью. Поклонники приезжают издалека лишь для того, чтобы выпить в пабе городка Ивреа, в честь которого феномен получил название Arduino. Банци говорит: «Люди приходят в бар и заявляют: «Мы здесь из-за платы Arduino». Но тут имеется одна проблемка. Бармены не знают что такое плата Arduino».

Открытая платформа Arduino высвобождает творческий потенциал

Lee Goldberg

Electronic Products

Во многом таким же образом, как мир программирования перевернули операционные системы с открытым исходным кодом GNU, Linux и другие, аппаратные платформы с открытым исходным кодом, поддерживающие Arduino, начинают вносить свои изменения в устоявшиеся правила для многих встраиваемых приложений. Изначально разработанная для сообщества радиолюбителей и устройств формата «сделай сам», платформа прототипирования электронных устройств с открытым исходным кодом Arduino основана на гибком, легком в использовании аппаратном и программном обеспечении. Однако для многих коммерческих разработчиков немаловажную роль играет низкая стоимость платформы и быстро растущая экосистема бесплатного ПО и приложений. В этом кратком обзоре мы рассмотрим аппаратную и программную часть платформы Arduino, а также некоторые приложения на ее базе.

История Arduino

Arduino – это не специализированный продукт или технология, это экосистема аппаратного и программного обеспечения, инструментов и человеческих ресурсов, в центре которой 8-разрядные микроконтроллеры семейства megaAVR. Входящие в состав платформы компактная плата, основный на C++ язык программирования и среда программирования своей универсальностью и простотой исполь-

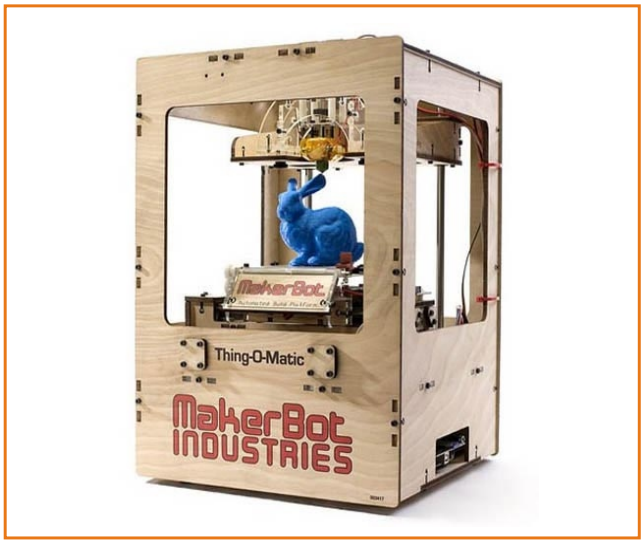


Рисунок 1. ***Дешевый, в сравнении с традиционными станками с ЧПУ, 3D принтер Thing-O-Matic компании MakerBot использует процессорную плату Arduino для управления шаговыми двигателями, печатающей термоголовкой и другими электромеханическими устройствами. Этот продукт превращает недорогой АБС-пластик в готовые объекты для создания прототипов или малотиражных изделий с допусками размеров 0.02 мм.***

зования во многом обязаны пробной платформе Wiring, разработанной в 2003 году Эрнандо Барраган (Hernando Barrag). Его целью было упрощение разработки электронных систем для любителей, исследователей и других непрофессионалов, обеспечивая легкий доступ ко всем возможностям и ресурсам, которые нужны опытным разработчикам. Выполненные на базе микроконтроллеров ATmega1281 и ATmega 2561, компактные и недорогие процессорные платы Wiring и WiringMini использовались для разработки интерактивных игрушек, роботов, играющих в футбол и контроллеров для сканирующего электронного микроскопа.

Платформа Arduino была разработана в Италии Массимо Банци (Massimo Banzi) и Дэвидом Куартилльз (David Cuartielles) в 2005 году как инструмент для студенческих проектов. Это расширение платформы Wiring, к которой добавлены несколько новых возможностей программирования и стандартная система соединений, позволяющие подключать к процессорной плате широкий круг взаимозаменяемых дополнительных модулей.

Аппаратная часть платформы Arduino

Как и ее предшественник, аппаратная платформа с открытым исходным кодом Arduino основана на универсальных микроконтроллерах Atmel семейства ATmega. Процессорное ядро AVR микроконтроллеров с модифицированной Гарвардской архитектурой сочетает в себе обширный набор инструкций с 32 рабочими регистрами общего назначения. Все 32 регистра непосредственно подключены к арифметико-логическому устройству (АЛУ), позволяя получить доступ к двум независимым регистрам за один машинный цикл. Микроконтроллеры ATmega могут иметь 4 или 8 КБайт внутрисхемно программируемой Flash-памяти, а также SRAM и EEPROM в различных комбинациях объема.

Процессорное ядро микроконтроллеров окружено разнообразной периферией и устройствами ввода/вывода. Хотя некоторые устройства различаются конфигурацией, многие из них имеют:

- 23 линии ввода/вывода общего назначения;
- 6- или 8-канальный 10-разрядный АЦП;
- шесть каналов ШИМ;
- последовательный интерфейс USART;
- 2-проводной байт-ориентированный последовательный интерфейс;
- последовательный порт SPI;
- три гибкийх таймера/счетчика;
- счетчик реального времени с отдельным генератором;
- программируемый сторожевой таймер со встроенным генератором.

Форм–фактор на любой случай

Платформа постоянно совершенствуется и пополняется новыми процессорными платами, разработанными под специфический форм-фактор или конкретное приложение. Многие проекты снабжены стабилизатором напряжения и набором разъемов для доступа к сигналам и линиям ввода/вывода. Одной из популярных плат, созданных командой Arduino, является плата Arduino Uno (Рисунок 2). На плате установлен кварцевый резонатор 16 МГц, разъем интерфейса USB, разъем питания, разъем интерфейса внутрисхемного программирования и кнопка сброса. Встроенный стабилизатор напряжения позволяет питать



Рисунок 2. ***Процессорная плата Arduino Uno.***

плату от сетевого адаптера или батареи, а также от интерфейса USB.

Сигналы 14 цифровых линий ввода/вывода и 6 аналоговых входов платы Arduino Uno выведены на разъемы. Шесть линий ввода/вывода могут использоваться как выходы ШИМ, – удобная функция для чего угодно – от диммирования освещения до управления электродвигателями. Эти же разъемы позволяют подключать мезонинные платы расширения (называемые Arduino Shield), которые могут содержать дополнительные электронные устройства и разъемы ввода/вывода.

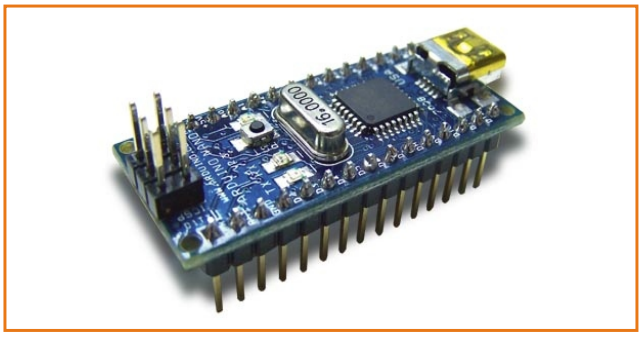


Рисунок 3. ***Arduino Nano.***

Arduino Nano – чуть более компактная плата на микроконтроллерах ATmega328 или ATmega168, разработанная специально для

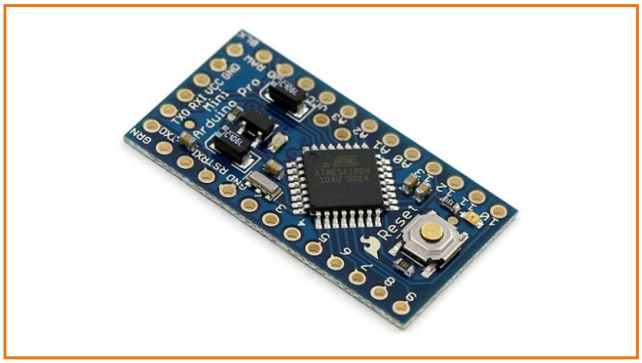


Рисунок 4. ***Arduino Mini.***



Рисунок 5. ***Arduino Ethernet.***

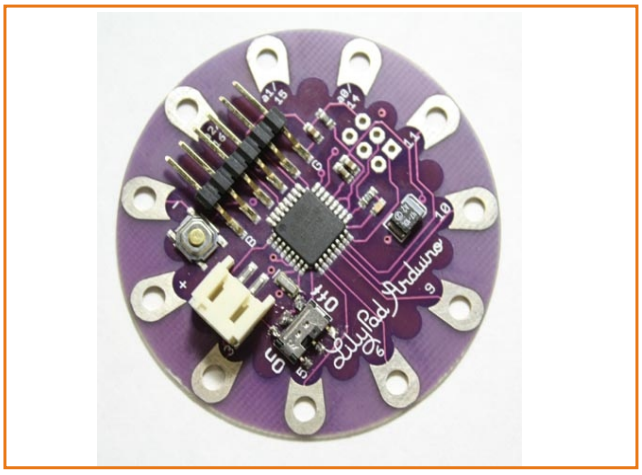


Рисунок 6. ***Arduino LilyPad.***

установки на макетные платы (Рисунок 3). В приложениях, для которых объем играет решающую роль, может использоваться плата

Arduino Mini (Рисунок 4). Другие доступные варианты процессорных плат содержат небольшую область для макетирования или встроенные коммуникационные средства, такие как Ethernet или Bluetooth (Рисунок 5). Одним из уникальных представителей семейства Arduino является плата LilyPad, разработанная для применения в «умной» одежде (Рисунок 6). Она может быть зашита в ткань и с использованием проводящих нитей соединена со встроенными источниками питания, датчиками и исполнительными устройствами. Плату даже можно стирать.

Программирование Arduino

Аппаратная платформа Arduino поддерживается пакетом программного обеспечения с открытым исходным кодом, в который входит интегрированная среда разработки (IDE), стандартный язык программирования с компилятором и загрузчик, запускаемый непосредственно на процессорной плате.

Разработка программного обеспечения осуществляется с помощью Arduino IDE – основанного на языке программирования Processing и проекте Wiring кросс-платформенного приложения, написанного на Java. Хотя первоначально оно создавалась как введение в программирование для новичков, не знакомых с разработкой ПО, Arduino IDE имеет возможности расширения, позволяющие опытным программистам работать более комфортно и продуктивно. Редактор кода включает в себя такие функции, как подсветка синтаксиса, подсветка скобок и автоматический отступ, а также имеется функция компиляции и загрузки программы в микроконтроллер один щелчком мыши.

Интегрированная среда разработки включает большой набор библиотек функций Си/Си++, поддерживающих большинство операций ввода/вывода. Полная версия среды разработки, коллекция расширений и демонстрационных приложений, коллекция разнообразных учебных программ – все доступно для свободного скачивания на сайте сообщества Arduino.

Открытая платформа Arduino высвобождает творческий потенциал.

Arduino Shields – расширение возможностей аппаратной платформы

Arduino – это не специализированный продукт или технология, это экосистема аппаратного и программного обеспечения, инструментов и человеческих ресурсов, и в центре всего – 8-разрядные микроконтроллеры семейства megaAVR, а с не давнего времени и 32-разрядный ARM микркоконтроллер на базе ядра Cortex-M3.

Наряду с основной аппаратной платформой – процессорной платой, немаловажную роль играют платы расширения Arduino Shields. Благодаря этим платам и модулям значительно расширяется область применения Arduino, а также открываются новые возможности для простых радиолюбителей и коммерческих разработчиков.

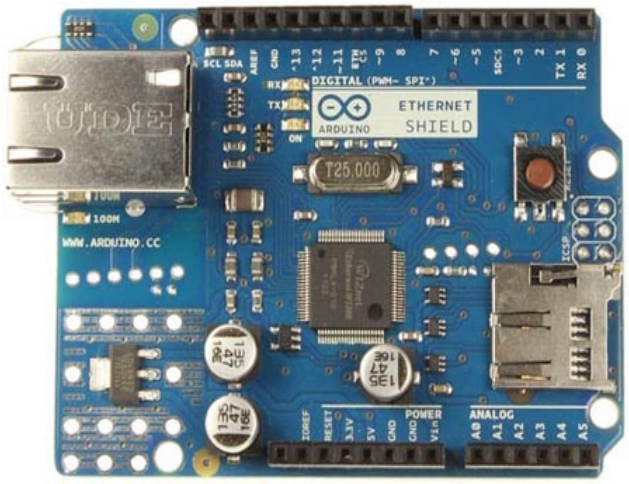
Практически все процессорные платы Arduino благодаря универсальному форм-фактору поддерживают подключение модулей расширения, мезонинных модулей и дополнительных плат, которые присоединяются к шине микроконтроллера и линиям ввода/вывода посредством специальных штыревых разъемов с шагом 2.54 мм. Универсальная система подключения значительно упрощает процесс внедрения в проекты схем расширения с использованием либо промышленных, предварительно собранных плат от различных производителей, либо одного из представленных на рынке наборов для самостоятельной сборки совместимых по выводам плат расширения.

Открытая платформа Arduino увлекает как любителей, так и коммерческих разработчиков встраиваемых систем, в результате чего появилось множество специализированных плат расширения, включая платы реле, платы АЦП и ЦАП высокого разрешения, платы контроля электрической сети, источников питания и преобразователей, платы драйверов электродвигателей. Arduino легко подключается к проводным сетям, благодаря наличию официальной платы Ethernet Shield, нескольких коммерческих вариантов плат Ethernet и соотвтетствующих программных библиотек.

Официальная плата Arduino Ethernet Shield

С помощью платы Arduino Ethernet Shield подключить платформу Arduino к сети Интернет можно за считанные минуты. Просто соедините модуль с процессорной платой, подключите сетевой кабель с разъемом RJ45 и следуйте несложным инструкциям. В итоге вы получите возможность контролировать свое устройство через Интернет. Каждый элемент

платформы (аппаратная часть, программная часть и документация) доступен бесплатно и имеет открытый код.



Плата расширения Arduino Ethernet совместима с процессорными платами Arduino Mega и Arduino Uno.

Основные характеристики платы расширения Ethernet Shield:

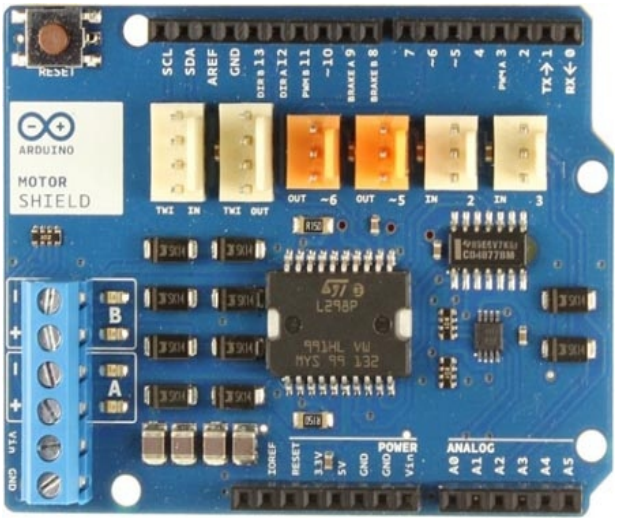
- Требуется процессорная плата Arduino
- Рабочее напряжение 5 В
- Ethernet контроллер Wiznet W5100 со встроенным буфером 16 КБайт
- Скорость подключения 10/100 Мбит/с
- Подключение к Arduino по интерфейсу SPI
- Установлен слот для карт памяти microSD
- Поддержка подключения модуля PoE:
  - IEEE802.3af совместимый
  - Низкие пульсации и шумы выходного напряжения
  - Диапазон входного напряжения 36 … 57 В
  - Защита от перегрузки и короткого замыкания
  - Выходное напряжение 9 В
  - Высокоэффективный DC/DC преобразователь
  - Напряжение изоляции вход/выход 1500 В
- Встроенный контроллер сброса.

Работа с платой расширения, а также с картой памяти осуществляется с помощью программной библиотеки, поддерживающей до 4 одновременных входных подключений по протоколам TCP или UDP. Встроенный разъем RJ45 содержит интегрированный линейный трансформатор и позволяет использовать функцию Power over Ethernet.

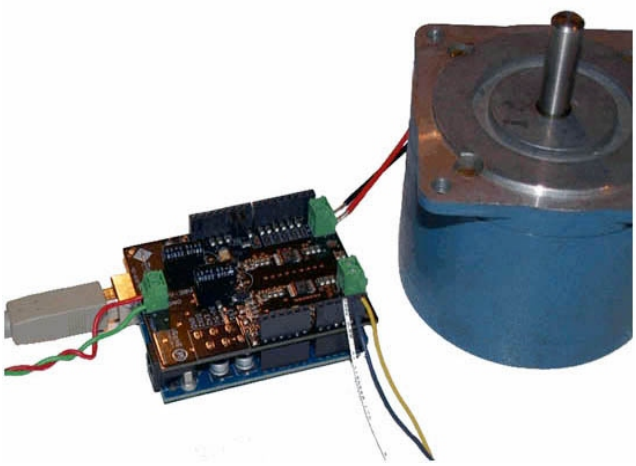
Официальная плата Arduino Motor Shield

Силовой модуль для управления электродвигателями предназначен для упрощения разработки робототехнических устройств и моторизированных систем.

Модуль выполнен на базе полномостового драйвера L298, предназначенного для управления индуктивной нагрузкой (реле, соленои-



Официальная плата Arduino Motor Shield может управлять двумя электродвигателям с контролем скорости и направления вращения.



Рассчитанная на рабочее напряжение до 30 В и рабочий ток до 2.8 А, эта коммерческая плата предназначена для управления двумя электродвигателями постоянного тока и может применяться в робототехнике или других моторизированных приложениях.

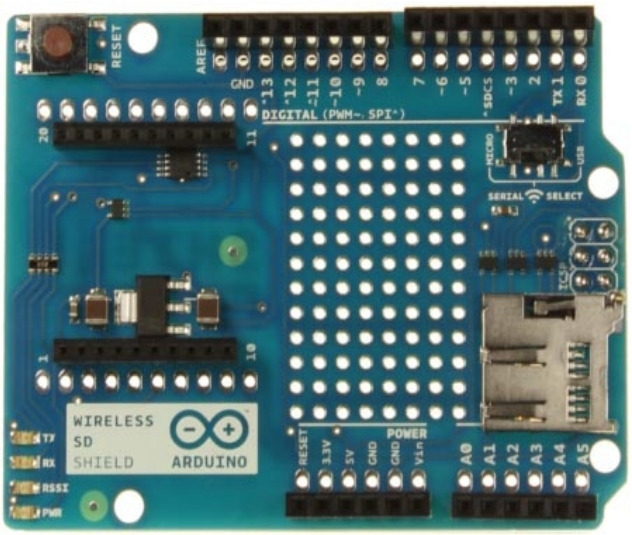
ды, двигатели постоянного тока и шаговые двигатели) и позволяет работать с двумя электродвигателями, контролируя скорость и направление вращения каждого независимо. Модуль рассчитан на работу только от внешнего источника питания.

Основные характеристики Arduino Motor Shield:

- Одновременное управление двумя электродвигателями постоянного тока или одним шаговым двигателем
- Независимый контроль скорости и направления движения
- Рабочее напряжение 5 … 12 В
- Установлен интегральный драйвер управления индуктивной нагрузкой L298
- Максимальный ток 2 А на канал, или общий максимальный ток 4 А
- Возможность измерения тока, потребляемого каждым двигателем (1.65 В/А).

Платы для беспроводных коммуникаций

На сегодняшний день приобретают большую популярность беспроводные устройства (системы безопасности, устройства управления и контроля, датчики и исполнительные механизмы), действующие по различным беспроводным протоколам. Проекты на базе Arduino идут в ногу со временем, благодаря возможности простого добавления беспроводных функций. Коммерческие платы расширения и модули, которые поддерживают ZigBee, Bluetooth и IEEE 801.11 протоколы также доступны для пользователей. Кроме того, имеется несколько плат, которые содержат завершенные сотовые трансиверы на базе GSM с поддержкой GPRS сервиса и подкачаются к платформе Arduino Uno и Mega с питанием от интерфейса USB.



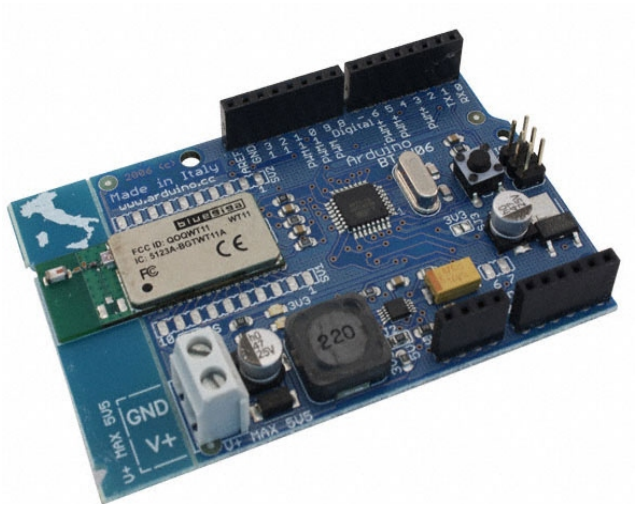
Официальная плата расширения Arduino Wireless SD Shield используется для организации беспроводной передачи данных с использованием, в основном, беспроводных модулей Xbee. Однако возможно использование и других беспроводных модулей с соответствующей топологией платы.

Официальная плата ***Arduino Wireless SD Shield*** добавляет возможность беспроводного обмена с отладочной платой. В основном, плата расширения рассчитана на подключение модулей XBee, однако возможно использование и других модулей с соответствующей топологией. Кроме того, на плате установлен слот для карт памяти SD, имеется небольшая область для макетирования и переключатель, который выбирает режим обмена с беспроводным модулем.



Плата расширения Xbee Shield (Digi-Key) с установленным радиомодулем.

Плата ***Arduino BT*** (Digi-Key) – это завершенная плата на базе микроконтроллера ATmega168 и беспроводного модуля Bluetooth Bluegiga WT11, обеспечивающая обмен по протоколу Bluetooth. Она имеет 14 цифровых входов/выходов (6 из которых могут использоваться как ШИМ выходы), 6 аналоговых входов, кварцевый резонатор 16 МГц, разъем внутрисхемного программирования. Отличительной особенностью является возможность программирования микроконтроллера по беспроводному интерфейсу.



Плата Arduino BT (Digi-Key) выполнена на микроконтроллере ATmega168 и модуле Bluegiga WT11.

Следует напомнить, что все официальные продукты Arduino имеют открытый код, т.е. для пользователей доступны принципиальные схемы плат, файлы топологии, программные библиотеки и демонстрационные приложения.

Во многом таким же образом, как платформа с открытым исходным кодом Linux постепенно внедрялась в коммерческий рынок, низкая стоимость и универсальность Arduino начинают привлекать к себе коммерческих разработчиков аппаратного обеспечения и встраиваемых систем. Хотя многие начальные и базовые приложения были разработаны для любителей и студентов, ожидается, что технология Arduino, использующаяся для их создания, быстро проложит себе дорогу к промышленным продуктам. ■

Аналоговый функционал Arduino: как использовать его в своих проектах

Lee Goldberg, Electronic Products

Будь вы инженер-разработчик, или просто любитель, вы сможете понять и оценить простоту использования аналоговых входов/выходов открытой аппаратной платформы Arduino в приложениях, работающих с «реальным миром». Компактные аппаратные модули обеспечивают легкий доступ к аналоговым входам и выходам микроконтроллера, которые могут использоваться для измерения и мониторинга напряжения, чтения различных типов аналоговых датчиков и простых сигналов.

Несмотря на то, что цифро-аналоговый преобразователь микроконтроллера имеет относительно скромное разрешение и скорость преобразования, он вполне подходит для многих распространенных приложений, от управления освещением и электродвигателями, до переключения коэффициентов усиления усилителей. В статье мы рассмотрим аппаратные и программные ресурсы, которые образуют основу аналогового функционала платформы Arduino и покажем, как использовать их в своих проектах.

Если Вы еще не знакомы с Arduino, обратите внимание на ссылки в конце статьи.

В соответствии с принятой философией – сделать применение цифровых технологий в реальном мире максимально простым, – аппаратная платформа Arduino была разработана таким образом, чтобы обеспечить возможность полного использования аналоговых функций, встроенных в универсальный 8-разрядный микроконтроллер ATmega компании Atmel. Все варианты микроконтроллеров, используемых в платформе Arduino, оснащены внутренним многоканальным АЦП, который имеет 10-битное разрешение и способен производить до 15000 выборок в секунду, представляя результаты в форме целых чисел от 0 до 1023. Основной функцией аналоговых выводов микроконтроллера является чтение входных аналоговых значений, но любой из них может быть сконфигурирован как цифровая линия ввода/вывода общего назначения.

Хотя некоторые AVR микроконтроллеры оснащены цифро-аналоговыми преобразователями, в нынешнем поколении плат Arduino используются представители семейства, аналоговые сигналы на выходах которых получаются с помощью широтно-импульсной модуляции (ШИМ). Коэффициент заполнения импульсов, имеющих частоту приблизительно 490 Гц, программируется на каждом выходе

ШИМ, что дает возможность получать эквивалентное действующее напряжение в диапазоне от 0 В до 5 В с 8-битным разрешением (Рисунок 1). Несмотря на некоторые ограничения в своих возможностях, выходы Arduino могут быть использованы для многих задач, таких как управление светодиодами и электродвигателями.



Рисунок 1. ***Широтно-импульсная модуляция позволяет превратить цифровые линии ввода/вывода общего назначения в аналоговые выходы Arduino.***

Большинство плат Arduino, как «официальных», так и других изготовителей, предоставляют легкий доступ к аналоговым (и цифровым) сигналам микроконтроллера посредством разъемов по краям платы. Количество аналоговых каналов и их физическое соответствие варьируются в зависимости от конкретного используемого микроконтроллера и форм-фактора платы, но многие варианты придерживаются соглашений, принятых для официальных проектов Arduino – Arduino Uno (Рисунок 2), Mega и Nano.



Рисунок 2. ***Аналоговые входы (A0 – A5) на плате Arduino Uno (rev3) и выходы ШИМ (цифровые выходы 3, 5, 6, 9, 10, 11) физически доступны через стандартные однорядные разъемы по краям платы.***

Разработка программного кода для функций аналогового ввода/вывода также несложна, т.к. среда разработки Arduino IDE содержит набор соответствующих собственных команд. Эти команды позволяют считывать аналоговые значения, генерировать выходной ШИМ сигнал и конфигурировать опорное напряжения для АЦП.

Аналого–цифровое преобразование и чтение данных

Задача использования аналоговых входов платформы Arduino в реальных приложениях довольно проста, но требует определенного внимания при выборе надлежащего источника опорного напряжения для АЦП. Чтобы определить верхний предел диапазона входного напряжения АЦП, здесь можно использовать источник опорного напряжения по умолчанию (Default), внутренний (Internal) или внешний

(External). В режиме Default микроконтроллер в качестве опорного источника использует выход установленного на плате Arduino стабилизатора напряжения. В зависимости от конкретного типа платы, это либо 5 В, либо 3.3 В.

В режиме Internal используется встроенный в микроконтроллер прецизионный источник опорного напряжения. Выходное напряжение этого источника различно в различных микроконтроллерах, и как правило составляет 1.1 В (для ATmega168 и ATmega328) или 2.56 В (для ATmega8 и других из серии mega). Режим External позволяет подключить внешний источник опорного напряжения к выводу микроконтроллера AREF через резистор номиналом 5 кОм. Вывод AREF имеет внутренний защитный резистор 32 кОм, который совместно с резистором 5 кОм образует делитель напряжения. Это означает, что при напряжении 2.5 В, приложенном через резистор, на выводе AREF будет 2.5 32 / (32 + 5) 2.2 В.

Чтения аналоговых напряжений с помощью языка программирования Arduino заключается в выборе источника опорного напряжения с использованием команды *analogReference (type)* и вызова функции *analogRead(pin)*, где

pin указывает на номер вывода разъема платы.

После назначения источника опорного напряжения он остается активным до выбора другого источника. Несмотря на то, что микроконтроллеры AVR поддерживают скорость преобразования до 15000 выборок в секунду, аппаратно-программная платформа Arduino, как правило, ограничивает это значение до 10000 выборок.

Создание аналоговых выходов ШИМ

Для генерации аналогового напряжения на одном из выводов ШИМ платы Arduino требуется конфигурирование этого вывода с помощью команды *pinMode(pin, mode)* и последующего вызова функции *analogWrite(pin, value)*, где

pin – указывает номер вывода ШИМ на разъеме платы Arduino,

value – значение, задающее выходное напряжение ШИМ (8-битное число в диапазоне 0-255).

На сконфигурированном однажды выходе будет постоянно присутствовать сигнал ШИМ частотой 490 Гц с заданными параметрами до следующего вызова функции *analogWrite()* (или *digitalRead()*, или *digitalWrite()*) для этого же вывода микроконтроллера.

Линии ввода/вывода микроконтроллера способны отдавать в нагрузку ток до 40 мА, поэтому они пригодны для непосредственного управления небольшими массивами светодиодов. Для управления более мощными источниками света или электродвигателями аналоговый выход микроконтроллера можно соединить с мощным транзистором или мостовой схемой. В приложениях более требовательных к качеству аналогового сигнала (усилители, источники тока) могут применяться простые RC фильтры.

Дополнительные аналоговые функции

Некоторые микроконтроллеры AVR (включая ATmega8 и ATmega168) имеют аналоговый компаратор, позволяющий сравнивать входное напряжение с другим внешним напряжением, которое может поступать с выхода ШИМ или от встроенного источника опорного напря-

жения микроконтроллера. Выход компаратора может опрашиваться микроконтроллером, или же может инициировать прерывание. Несмотря на дополнительный программный код, работа компаратора с выработкой прерывания позволяет процессору реализовать функции определения повышенного/пониженного напряжения без постоянного чтения аналоговых данных. Эта функция может использоваться в детекторах движения с регулируемым порогом и датчиках удара в биомедицинском мониторинге.

Для плат Arduino, микроконтроллеры которых не имеют встроенного аналогового компаратора, относительно легко можно добавить внешний компаратор LM741, LM339N или TLC3704, установив его в макетной области платы. Если свободного места на вашей плате нет, можно использовать специальную плату для прототипирования (Рисунок 3).

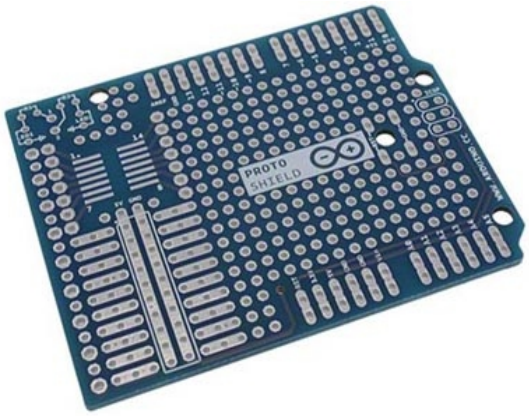


Рисунок 3. ***Плата для прототипирования позволит подключать ваши собственные аналоговые (или цифровые) схемы ввода/вывода к стандартным платам Arduino.***

Ссылки

1. «Радиолоцман», 2011, 11, стр. 13, «Как создавали и продвигали Arduino».
2. «Радиолоцман», 2011, 12, стр. 44, «Открытая платформа Arduino высвобождает творческий потенциал».
3. «Радиолоцман», 2012, 01, стр. 34, «Открытая платформа Arduino высвобождает творческий потенциал. Arduino Shields – расширение возможностей аппаратной платформы».

Сетевые возможности Arduino-приложений и систем

Lee Goldberg

Electronic Products

Добавление сетевых коммуникационных возможностей к открытой аппаратно-программной платформе Arduino реализуется просто и относительно дешево, но благодаря им открывается мир новых приложений, как для любителей, так и для инженеров и коммерческих разработчиков (Рисунок 1). Из предлагаемой ознакомительной статьи вы узнаете, как добавить Ethernet, Wi-Fi или интерфейс стандарта 802.15 в систему на базе Arduino.

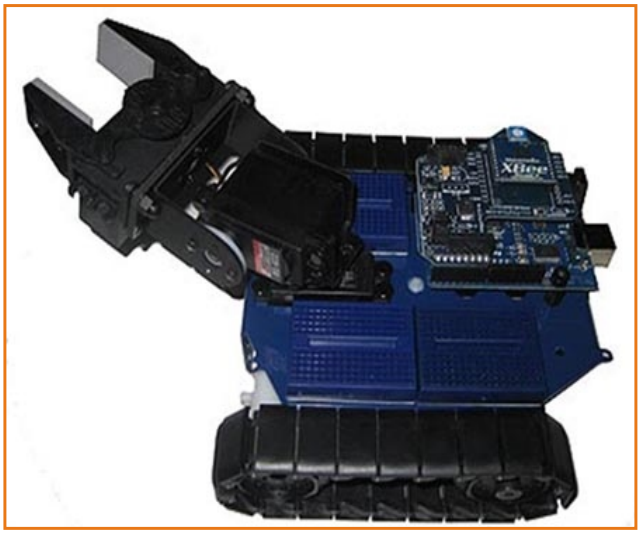


Рисунок 1. *Робот-игрушка RobotShop Rover под управлением Arduino, который программируется и получает команды от ПК по интерфейсу USB. Модуль приемопередатчика XBee 2.4 ГГц может использоваться для удаленного управления и получения данных от робота.*

Если вы еще не знакомы с Arduino, обратите внимание на ссылки в конце статьи.

На самом деле, практически каждое устройство Arduino имеет какие-то коммуникационные возможности. На самом базовом уровне платформа должна предоставлять некоторые средства для разработки и загрузки кода приложения с использованием персонального компьютера, а также должна обеспечиваться поддержка отладки и трассировки. Многие платы Arduino (такие как Uno, Mega2560) имеют интегрированный порт USB, используемый и в качестве канала связи, и в качестве маломощного источника питания. В некоторых специализированных платах такая роскошь как соединение по USB отсутствует, и в них используется возможность связи по последовательному интерфейсу RS-232 (или эквивалентному), что позволяет использовать для обмена с ними практически любое устройство с последовательным портом.

В платформе Arduino предусмотрены возможности последовательного обмена с другими микросхемами (расположенными на основной плате, или на отдельных модулях) посредством интерфейсов SPI, I2C или TWI. Во многих случаях это реализуется на основе ресурсов платы или небольших адаптеров.

Для многих современных приложений требуется наличие возможности сетевого обмена, посредством которого можно осуществлять локальный обмен данными с другими платформами Arduino или через Интернет. Шина CAN или интерфейс I2C часто используются для проводного M2M соединения (Machine-to-Machine – M2M – общее название технологий, позволяющих машинам обмениваться информацией друг с другом, или же передавать ее в одностороннем порядке), в то время как радиочастотный 802.15-совместимый интерфейс (2.4 ГГц) очень популярен для беспроводной передачи данных. Если для Вашего проекта требуется связь с «внешним миром», есть целый ряд аппаратных модулей расширения Arduino, которые могут предоставить Ethernet, Wi-Fi или CDPD/GSM связь.

Аппаратные средства – базовые понятия

Хотя процессорные платы Arduino отличаются друг от друга форм-факторами и возможностями, все они отвечают набору требований к аппаратной платформе с открытым исходным кодом, что делает их частично или полностью совместимыми между собой. К примеру, во многих платах Arduino используется одинаковое распределение сигналов по контактам разъемов, предоставляющих доступ к шине микроконтроллера и линиям ввода/вывода, что упрощает подключение плат расширения Ethernet, ZigBee или иных сетевых плат расширения.

В соответствии с этой философией, среда разработки Arduino была разработана для поддержки широкого спектра целевых плат с различными микроконтроллерами, различными компонентами ввода/вывода, различными тактовыми частотами и загрузчиками. Изначально разработанная для поддержки микроконтроллеров AVR серии ATmega компании Atmel (микроконтроллеров с загрузчиком Arduino Uno), с недавних пор среада разработки поддерживает варианты плат Uno и Mega на базе 32-разрядных микроконтроллеров Microchip серии PIC32MX.

Программные средства – базовые понятия

Программный комплекс Arduino состоит из интегрированной среды разработки и соответствующих программных библиотек. Приложения (скетчи) пишутся на языке Processing, который, фактически, является надстройкой над Java. Приложения перед компиляцией предварительно обрабатываются для определения аппаратно-зависимых аспектов целевой платы (или любых плат расширения, которые они использует).

Характеристики каждой платы определены в файле с именем boards.txt с использованием нескольких переменных с соответствующим префиксом. Плата описывается последовательностью переменных с именем самого параметра, типа BOARD.name, BOARD.build.f_cpu (например, "ATmega328" или "ATmega1280") и BOARD.build.core, которые содержат необходимые данные необходимые для компилятора и приложения.

Процесс компиляции собирает все базовые и пользовательские библиотеки, используемые приложением, включая библиотеку поддержки последовательного интерфейса, которая служит для обмена данными между платой Arduino и ПК или другим микроконтроллером, эмулируя 9-контактный последовательный порт, который часто встречается в компьютерах. Сопутствующая библиотека сетевых функций позволит плате Arduino нала-

дить обмен данными через Интернет, а также организовать функции клиент/сервер.

Набор базовых библиотек также включает в себя поддержку специфических сетевых интерфейсов, включая USB, SPI, TWI/I2C и Ethernet. Дополнительные, отдельно поставляемые библиотеки, могут использоваться для реализации Web-интерфейса, а также для поддержки других проводных и беспроводных протоколов.

Сетевые технологии, доступные на платформе Arduino

Рассмотрим примеры сетевых технологий, имеющихся в экосистеме Arduino. Эти примеры, как правило, касаются «официальных» решений, предлагаемых сообществом Arduino, однако сейчас существует множество аналогичных коммерческих и некоммерческих вариантов, доступных от различных производителей и групп пользователей.

USB

Многие платы Arduino уже содержат в себе интерфейс USB, но некоторые, в компактном форм-факторе или бюджетные варианты, его не имеют. Самый простой способ добавить USB к таким платам – использование дополнительного адаптера USB Serial Light adapter (Рисунок 2), выполняющего функцию моста USB-UART (преобразует сигналы Rx и Tx последовательного порта микроконтроллера в сигналы интерфейса USB).

В адаптере используется микроконтроллер ATmega8U2, запрограммированный для выполнения функций конвертера USB-UART. Так как ПО микроконтроллера использует стандартные драйверы USB COM, установка внешних драйверов не потребуется. Кроме того, пользователи могут самостоятельно изготовить такой адаптер, как на микроконтроллере ATmega8U2, так и на специализированной микросхеме FT232RL производства компании Future Technology.

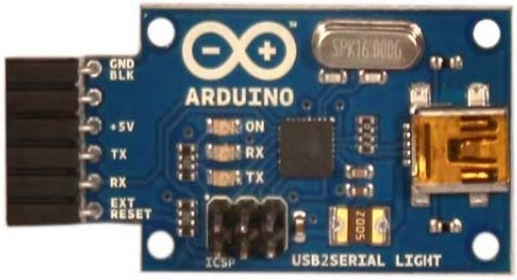


Рисунок 2. *Адаптер Arduino USB Light adapter может использоваться для реализации USB интерфейса на платах Arduino.*

Ethernet

Платы Arduino Uno и Mega, а также многие их производные, позволяют использовать стандартную плату расширения Ethernet Shield (Рисунок 3), построенную на базе 10/100 Ethernet контроллера Wiznet W5100. Плата также имеет установленный слот для карт памяти microSD, которая может использовать-

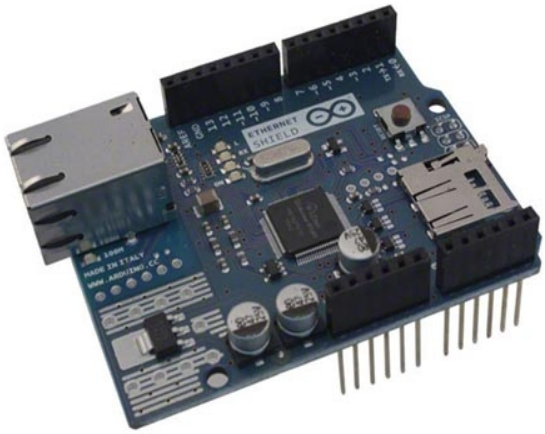


Рисунок 3. *Плата расширения Arduino Ethernet Shield, поддерживающая 10/100 Мбит/с Ethernet, может работать совместно с модулем PoE.*

ся для хранения файлов, доступ к которым возможен через сеть. Arduino «общается» с контроллером Ethernet и картой памяти по интерфейсу SPI (посредством разъема ICSP). Питание плата может получать по сетевому Ethernet кабелю (технология Power-over-Ethernet), но для этого требуется специальный модуль DC/DC преобразователя. Основная Ethernet библиотека содержит файлы, которые позволяют платам Arduino Uno и Mega инициализировать контроллер Ethernet и работать с ним, организовать Клиент или Сервер для четырех одновременных входящих или исходящих соединений.

802.15/XBee

Для Arduino доступно множество различных опций беспроводной передачи данных по стандарту IEEE 802.15.3 частотного диапазона 2.4 ГГц. В целом, сетевой протокол ZigBee не очень популярен из-за его требований к большому объему памяти и вычислительной мощности. Вместо него получил достаточно широкое распространение ZigBee-подобный протокол, названный XBee. Он также поддерживает звездообразную и полносвязную топологию сети и использует стандартный набор модемных AT команд для управления передачей данных.

Одной из самых популярных реализаций беспроводной передачи данных является плата расширения XBee Shield, разработанная компанией Libelium в тесном сотрудничестве с Arduino (Рисунок 4a и 4b). Плата выполнена на базе радио модулей XBee или XBee-Pro, производства компании Digi. Модули поддерживают различные топологии сети и могут использоваться в качестве беспроводного моста USB-Serial.

CDPD/GSM передача данных

Добавление мобильного сотового подключения в проект Arduino может быть выполнено с помощью GSM платы расширения (CDPD – цифровая сотовая пакетная передача данных). Плата расширения Arduino GPRS/GSM Shield выполнена на базе радио модуля Quectel M10, который управляется AT командами по последовательному интерфейсу. Для идентификации пользователя и системы в сотовой сети потребуется SIM карта. Много продуктов доступно и от других производителей, к примеру, CuteDigi, HW Kitchen и Sparkfun, которые используют в своих решениях 4-х диапазонный радио модуль Spreadtrum SM5100B.

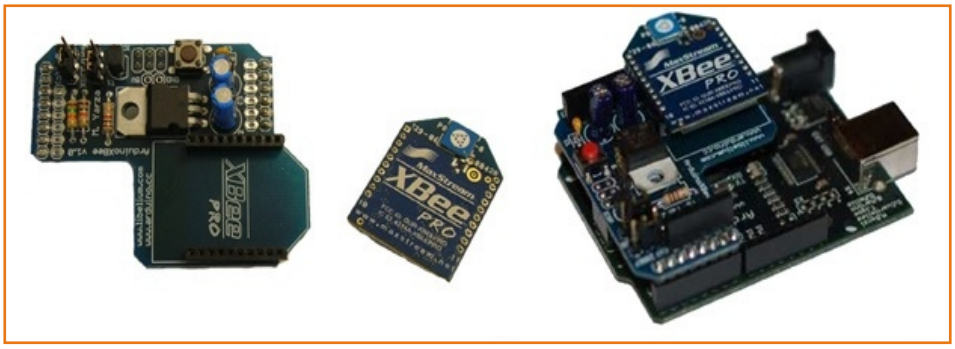


Рисунок 4. *Платы расширения Libelium XBee Shield позволяют непосредственно подключить радио модули MaxStream к популярным платам Arduino.*

Другие приложения

Платформа Arduino также включает в себя платы расширения и специализированные процессорные платы, поддерживающие и другие проводные и беспроводные сетевые технологии, включая Wi-Fi,

CAN и ModBus. Кроме того, существуют расширения для поддержки локальных интерфейсов, таких как SPI и низкоскоростной сигнальный интерфейс 1-Wire. Об этих возможностях мы постараемся рассказать в последующих статьях. ■

Ссылки

1. «Радиолоцман», 2011, 11, стр. 13, «Как создавали и продвигали Arduino».
2. «Радиолоцман», 2011, 12, стр. 44, «Открытая платформа Arduino высвобождает творческий потенциал».
3. «Радиолоцман», 2012, 01, стр. 34, «Открытая платформа Arduino высвобождает творческий потенциал. Arduino Shields – расширение возможностей аппаратной платформы».
4. «Радиолоцман», 2012, 02, стр. 41, «Аналоговый функционал Arduino: как использовать его в своих проектах».



Arduino Due
– официальная плата Arduino на микроконтроллере ARM

wired.com

Открытая платформа Arduino, безусловно, помогает в решении самых разных задач множеству разработчиков и любителей, требуя от них лишь минимальных представлений об электронике и программировании. Платформа содержит все необходимое – интегрированную среду разработки, компилятор, программатор и отладочную плату. Огромное количество библиотек и плат расширения прочно укрепило лидерство Arduino в соревновании с другими платформами. Тем не менее, имеется немало претензий в отношении производительности, памяти и возможностей Arduino, основанной на микроконтроллере AVR. С появлением Arduino Due такое положение вещей должно измениться, по крайней мере, в некоторых областях. Правда, новая плата сделает бесполезными созданную среду разработки и большинство библиотек, но компенсацией за это будет многократно возросшая производительность и намного больший объем памяти.

Долгожданная Arduino Due только выходит на рынок. В ней 8-битный 16 МГц микроконтроллер AVR, использовавшийся в популярной плате Arduino Uno, заменен 32-разрядным процессором с рабочей частотой 84 МГц, Кроме того, увеличено количество входов и значительно расширен перечень возможностей (Рисунок 1).

Для любителей, увлеченных разработкой робототехники и электроники, это волнующий момент. Но остальным из нас, что же предложит новый микроконтроллер по сравнению с предшественником?

«Имея 32-битный ARM процессор, работающий на частоте 84 МГц, можно выполнять намного больше задач, и гораздо быстрее», – объясняет соучредитель Arduino Массимо Банци (Massimo Banzi) в письме порталу

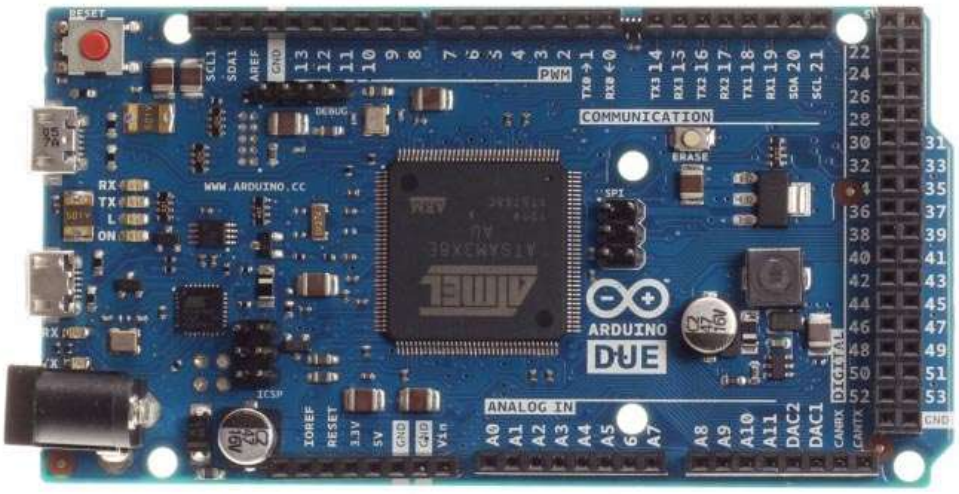


Рисунок 1. *С помощью платы Arduino Due с микроконтроллером Atmel AT91SAM3X8E можно сконструировать 3-D принтер, имеющий более высокое разрешение.*

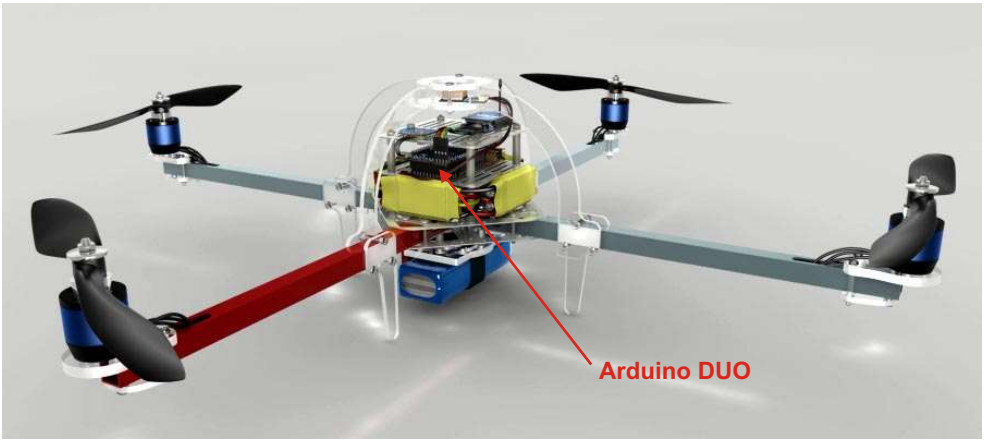


Рисунок 2. *Для управления квадрокоптером используется плата Arduino Due.*

Wired.com. – «Например, представьте себе квадрокоптеры (Рисунок 2), которые строят Крис Андерсон (Chris Anderson) и его сообщество. Там необходимо получать данные от множества датчиков настолько быстро, насколько возможно, а затем все эти данные обрабатывать для формирования правильных команд управления. С быстрым процессором и таким полезными дополнительными функциями, как, скажем, прямой доступ к памяти, можно повысить стабильность, реактивность и управляемость этих летательных аппаратов, к тому же используя для этого меньше чипов».

«Сердцем» Arduino Due является микроконтроллер Atmel SAM3X8E с ядром Cortex-M3. По функциональным возможностям плата аналогична Arduino Leonardo, выпущенной на несколько месяцев раньше. Arduino Due имеет два порта MicroUSB: один предназначен для программирования и коммуникаций, а второй позволяет использовать Due в качестве клиента или Host-устройства, предоставляя плате возможность поддерживать подключение USB клавиатуры или мыши, или же самой выступать в роли этих устройств. «В последние годы интерфейс USB Host востребован пользователями, и это нововведение позволит расширить сферу применения платформы», – утверждает Массимо Банци.

Благодаря возможностям нового микроконтроллера Atmel платформа Arduino Due сделала гигантский скачок вперед в части характеристик аналого-цифрового преобразователя, и позволила разработчикам расширить границы своих творений. «До сих пор многие люди создавали интереснейшие научные приборы с открытой документацией, используя плату Arduino. Теперь же с Arduino Due они получат 12-разрядные аналоговые входы и 12-разрядные аналоговые выходы», – говорит Банци. Теоретическая скорость преобразования увеличилась колоссально – до 1 миллиона выборок в секунду. Для сравнения, платы Arduino Uno, Leonardo и Mega 2560 имеют теоретическую скорость преобразования 15 тысяч выборок в секунду.

Due также является первой в семействе Arduino платой со встроенными цифро-аналоговыми преобразователями, которых у Due два. В этой связи анонсированы библиотеки аудио функций, поддерживающие заложенную в плату возможность воспроизведения wav-файлов. Одновременно появились слухи об использовании в Arduino Due кода OGG плеера, разработанного Google.

Команда Arduino сотрудничала с разработчиками Google Android Developer Kit на платформе ADK 2012 (Android Accessory Development Kit), которые для своей платы использовали версию Due. Но, несмотря на общую популярность Android, Банци видит много возможностей роста в сообществе разработчиков ADK.

«К сожалению, мы видели совсем немного приложений, разработанных сообществом», – говорит он. – «Я думаю, что Google должна уделять больше внимания сотрудничеству с сообществом ADK, чтобы создавать хорошие руководства, учебники, документацию, и тем самым привлечь пользователей. Если вдуматься, Apple имеет аналогичную технологию, которая является ее собственностью, защищена соглашением о неразглашении и требует использования специализированно-

го чипа, однако это не препятствует разработке множества аппаратных аксессуаров для iPhone/iPad. Напротив, Android реализована как хорошая платформа с открытым исходным кодом, но такого развития она до сих пор не имеет».

При этом платформа Due продолжает поддерживать протокол ADK 2.0, что делает ее совместимой с библиотеками Google и некоторыми кодами, написанными для ADK 2012.

Плата Due выполнена в стандартном для Arduino форм-факторе, поэтому с ней будут работать все платы расширения и схемы, например, датчики движения и матрицы светодиодов, что соответствует Ревизии 3 официальной плат. Однако, Due имеет напряжение питания 3.3 В, в то время как платы на микроконтроллерах AVR работают при напряжении питания 5 В. Это означает, что некоторые неофициальные платы расширения, не отвечающие спецификации Ревизии 3, могут быть несовместимы с Due. Разработчики, которые захотят использовать Arduino Due в существующих приложениях, должны учитывать это обстоятельство, чтобы избежать повреждения своей платы.

Команда Arduino также заверила, что изменения в интегрированной среде разработки Arduino IDE сделают ее кроссплатформенной. Коды (скетчи), написанные для платформы Uno, или другой платформы на микроконтроллере AVR, смогут выполняться и на Due. Конечно же, под капотом вы увидите различия в том, как компилируются Ваши программы, но снаружи все будет гладко и беспроблемно для пользователя.

Несмотря на увеличение вычислительной мощности и возможностей, начинающие пользователи, вероятно, будут отдавать предпочтение классической платформе Uno. «Базовые платы Arduino, такие как Uno и Leonardo, остаются самыми лучшими для изучения», – говорит Банци. – «Они исключительно просты, стабильны в работе, и для них уже есть огромное количество примеров и библиотек».

Что касается задержки с выпуском Arduino Due, Банци объясняет это болезнями роста. «В последние два года мы вынуждены были перейти от практики подключения к проектам свободных групп людей к формированию компаний, способных выполнять весь спектр деловых функций», – говорит он. – «Сейчас по всему миру вы можете встретить двери с табличкой «Arduino» (или «Officine Arduino»), за которыми выполняет свою работу профессиональная команда. Рост иногда замедляет работу».

Плата Arduino Due, официально представленная в октябре 2012 года, продается по цене $49. РЛ

Технические характеристики Arduino Due

Микроконтроллер	AT91SAM3X8E
Flash-память	512 Кбайт
ОЗУ	96 Кбайт (64 + 32 Кбайт)
Тактовая частота	84 МГц
Напряжение питания	3.3 В
Входное напряжение (рекомендованное)	7…12 В
Входное напряжение (мин./макс.)	6…20 В
Цифровые линии ввода/вывода	54 (из них 6 выходы ШИМ)
Аналоговые входы	12
Аналоговые выходы (ЦАП)	2
Общий постоянный выходной ток на всех линиях ввода/вывода	130 мА
Постоянный ток для вывода 3.3 В	800 мА
Постоянный ток для вывода 5 В	800 мА (теоретически до 1 А)
Отладочный интерфейс	JTAG/SWD

Ссылки по теме

1. «Как создавали и продвигали Arduino». РадиоЛоцман, 2011, ноябрь, стр. 13.
2. «Открытая платформа Arduino высвобождает творческий потенциал». РадиоЛоцман, 2011, декабрь, стр. 44.





РадиоЛоцман.Журнал

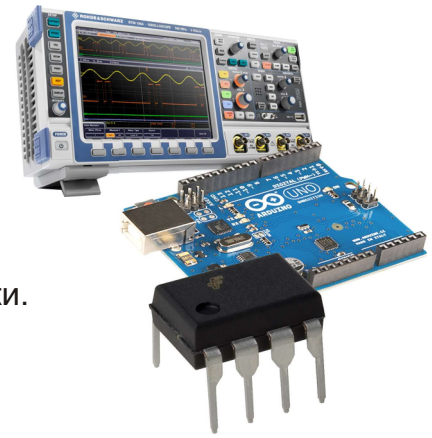
www.rlocman.ru/magazine

Официальные версии только здесь: без регистрации и назойливой рекламы, с высокой скоростью скачивания.

РадиоЛоцман.Форум

www.rlocman.ru/forum

Обсуждайте статьи и новости, общайтесь!
Приглашаем профессионалов.



РадиоЛоцман.Цены

www.rlocman.ru/price

Подберите оптимального поставщика электронных компонентов, измерительной и электронной техники.
Цены, условия поставки и доставки.

РадиоЛоцман.Соц.Сети

Обновления в привычном формате.

vk.com/rlocman

facebook.com/rlocman

odnoklassniki.ru/rlocman

twitter.com/rlocman

google.com/+RlocmanRuPlus

И еще многое другое на портале:

РадиоЛоцман. ...

www.rlocman.ru

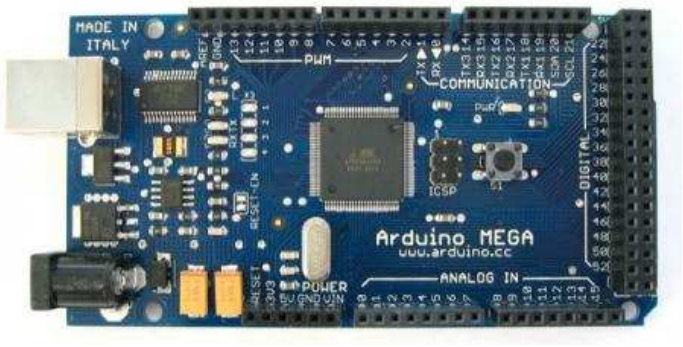


Arduino.ru http://arduino.ru/

Материалы по программированию Arduino переведены с официального сайта проекта Ардуино — **http://arduino.cc**
и представлены по лицензии Creative Commons Attribution-ShareAlike 3.0 License.

Arduino?

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.



Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

Микроконтроллер на плате программируется при помощи языка Arduino (основан на языке Wiring) и среды разработки Arduino (основана на среде Processing). Проекты устройств, основанные на Arduino, могут работать самостоятельно, либо же взаимодействовать с программным обеспечением на компьютере (напр.: Flash, Processing, MaxMSP). Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Программное обеспечение доступно для бесплатного скачивания. Исходные чертежи схем (файлы CAD) являются общедоступными, пользователи могут применять их по своему усмотрению.

Почему Arduino?

Существует множество микроконтроллеров и платформ для осуществления «physical computing». Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard и многие другие предлагают схожую функциональность. Все эти устройства объединяют разрозненную информацию о программировании и заключают ее в простую в использовании сборку. Arduino, в свою очередь, тоже упрощает процесс работы с микроконтроллерами, однако имеет ряд преимуществ перед другими устройствами для преподавателей, студентов и любителей:

Низкая стоимость – платы Arduino относительно дешевы по сравнению с другими платформами. Самая недорогая версия модуля Arduino может быть собрана в ручную, а некоторые даже готовые модули стоят меньше 50 долларов.

Кросс-платформенность – программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows.

Простая и понятная среда программирования – среда Arduino подходит как для начинающих пользователей, так и для опытных. Arduino основана на среде программирования Processing, что очень удобно для преподавателей , так как студенты работающие с данной средой будут знакомы и с Arduino.

Программное обеспечение с возможностью расширения и открытым исходным текстом – ПО Arduino выпускается как инструмент, который может быть дополнен опытными пользователями. Язык может дополняться библиотеками C++. Пользователи, желающие понять технические нюансы, имеют возможность перейти на язык AVR C на котором основан C++. Соответственно, имеется возможность добавить код из среды AVR-C в программу Arduino.

Аппаратные средства с возможностью расширения и открытыми принципиальными схемами – микроконтроллеры ATMEGA8 и ATMEGA168 являются основой Arduino. Схемы модулей выпускаются с лицензией Creative Commons, а значит, опытные инженеры имеют возможность создания собственных версий модулей, расширяя и дополняя их. Даже обычные пользователи могут разработать опытные образцы с целью экономии средств и понимания работы.

Справочник языка Ардуино

1. Операторы

```
setup()
loop()
```

Управляющие операторы

```
if
if...else
for
switch case
while
do... while
break
continue
return
goto
```

Синтаксис

```
; (semicolon)
{} (curly braces)
// (single line comment)
/* */ (multi-line comment)
```

Арифметические операторы

```
= (assignment)
+ (addition)
- (subtraction)
* (multiplication)
/ (division)
% (modulo)
```

Операторы сравнения

```
== (equal to)
!= (not equal to)
< (less than)
> (greater than)
<= (less than or equal to)
>= (greater than or equal to)
```

Логические операторы

```
&& (И)
|| (ИЛИ)
! (Отрицание)
```

Унарные операторы

```
++ (increment)
-- (decrement)
+= (compound addition)
-= (compound subtraction)
*= (compound multiplication)
/= (compound division)
```

2. Данные

Константы

```
HIGH | LOW
INPUT | OUTPUT
true | false
delayMicroseconds()
```

Математические функции

```
min()
max()
abs()
constrain()
map()
pow()
sq()
sqrt()
```

Тригонометрические функции

```
sin()
cos()
tan()
```

Генераторы случайных значений

```
randomSeed()
random()
```

Внешние прерывания

```
attachInterrupt()
detachInterrupt()
```

Функции передачи данных

```
Serial
```

Целочисленные константы
Константы с плавающей запятой

Типы данных

```
boolean
char
byte
int
unsigned int
word
long
unsigned long
float
double
string - массив символов
String - объект класса
массив (array)
void
```

Преобразование типов данных

```
char()
byte()
int()
long()
float()
```

Область видимости переменных и квалификаторы

Область видимости

```
static
volatile
const
```

3. Функции

Цифровой ввод/вывод

```
pinMode()
digitalWrite()
digitalRead()
```

Аналоговый ввод/вывод

```
analogRead()
analogReference()
analogWrite()
```

Дополнительные фунции ввода/вывода

```
tone()
noTone()
shiftOut()
pulseIn()
```

Работа со временем

```
millis()
micros()
delay()
```

4. Библиотеки Arduino

`Servo` — библиотека управления сервоприводами.
`EERPOM` — чтение и запись энергонезависимой памяти микроконтроллера.
`SPI` — библиотека, реализующая передачу данных через интерфейс SPI.
`Stepper` — библиотека управления шаговыми двигателями.

5. Примеры

Изменяем яркость светодиода — плавное изменение яркости светодиода функцией `analogWrite()`.

Мигаем светодиодом — пример подключения светодиода к Arduino и работы с ним

Тактовая кнопка — считывание состояния кнопки

Мигаем светодиодом без `delay()` - еще один, более практичный способ мигать светодиодом

Управление сервой фоторезистором - угол зависит от освещения

Начало работы с Arduino в Windows

Данный документ разъясняет, как подключить плату Arduino к компьютеру и загрузить ваш первый скетч.

Необходимое железо — Arduino и USB-кабель

Программа — среда разработки для Arduino

Подсоедините плату

Установите драйвера

Запустите среду разработки Arduino

Откройте готовый пример

Выберите вашу плату

Выберите ваш последовательный порт

Загрузите скетч в Arduino

Программирование Ардуино

Базовые и полезные знания, необходимые для успешного программирования под платформу Arduino:

Начало работы с Arduino в Windows

Работа с Arduino Mini

Цифровые выводы

Аналоговые входы

Широтно-импульсная модуляция

Память в Arduino

Использование аппаратных прерываний в Arduino

Перепрошивка контроллера Atmega8U2 для Arduino Uno и Mega2560

Переменные

Функции

Создание библиотек для Arduino

Использование сдвигового регистра 74HC595 для увеличения количества выходов

Прямое управления выходами через регистры микроконтроллера Atmega

Проекты на Arduino

OLED дисплей + ChipKit MAX32

О производстве Arduino

Arduino в руках художников. Простая и очень красивая инсталляция на базе Arduino

Оправка смс через Siemens S45 с помощью Arduino, а также Zyxel Keenetic 4G, Arduino и датчика температуры ds18b20

Робот, балансирующий на шаре

Двухколесный балансирующий робот на Arduino

Анализатор спектра звука на Arduino

Робот, управляемый по Wi-Fi, на базе роутера и Arduino

Оригами лампа с емкостным сенсором

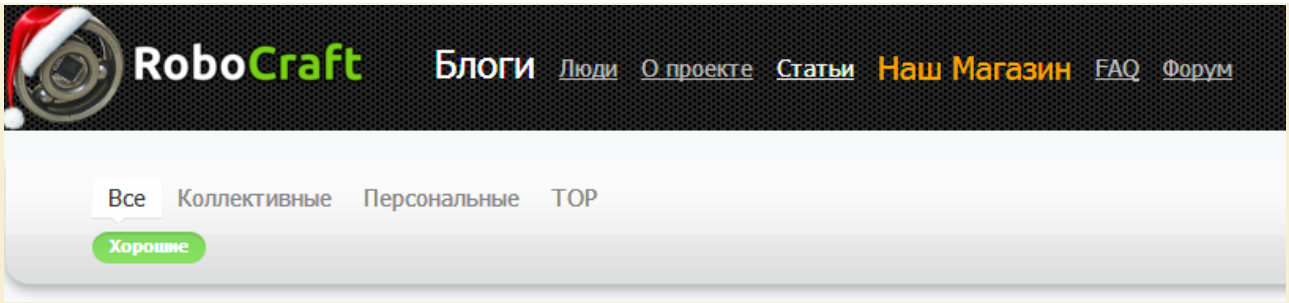
Роботизированная рука под управлением Arduino

Arduino и RGB светодиодная матрица 8x8

Пишем в воздухе светодиодами

Остановить мгновение. Фототриггер на Arduino

Управление фотовспышкой с помощью Arduino контроллера



RoboCraft.ru http://robocraft.ru/

Роботы? Это просто! RoboCraft - это сообщество/коллективный блог любителей робототехники, электроники и программирования (а также, команда для изучения и разработки робототехнических проектов).

Здесь Вы можете:
почитать новости робототехники;
ознакомиться с различными технологиями и примерами их использования;
опубликовать свои заметки, идеи и наработки;
задавать вопросы и искать ответы на форуме;
приобрести необходимые модули в нашем магазине.

Цели проекта:
делиться своими идеями и наработками друг с другом; найти единомышленников; сделать будущее ближе!

Arduino http://robocraft.ru/page/summary/ http://robocraft.ru/tag/Arduino/

СТАТЬИ и ПРОЕКТЫ

Введение

Arduino - введение

Ардуино - что это и зачем?

Что за язык программирования используется для Arduino?

ХоумМейд Arduino - как сделать Arduino своими руками

Arduino как фреймворк

КМБ для начинающих ардуинщиков

Состав стартера (точка входа для начинающих ардуинщиков)

Как адаптировать библиотеки для Arduino 1.0

Разновидности плат Arduino, а также про клоны, оригиналы и совместимость

Статьи по программированию Arduino/CraftDuino

1. Программирование Arduino - введение
2. Структура программы, константы
3. Цифровой ввод/вывод
4. Аналоговый ввод/вывод
5. Дополнительные функции ввода/вывода
6. Работа со временем
7. Математические функции
8. Псевдослучайные числа
9. Последовательная передача данных
10. Прерывания
11. EEPROM
12. Создание своей библиотеки

Blink без delay

Библиотеки

Ethernet library – библиотека для работы с Ethernet

Servo library – библиотека для работы с сервомашинками

Firmata library – библиотека, реализующая протокол Firmata

OneWireSlave - библиотека эмуляции устройства 1-Wire

TimerOne - библиотека для работы с таймером МК

Wire - библиотека для работы с I2C

GLCD - библиотека для работы с графическими LCD

Дополнительно

Arduino/CraftDuino и PROTEUS

Arduino/CraftDuino и WinAVR - программируем на чистом C

Статьи по практическому программированию Arduino/CraftDuino

Подготовка к работе с Arduino/CraftDuino

Ошибки Arduino

0. Начало
1. Цифровой ввод - кнопка
2. Аналоговый вывод - Fading
3. Аналоговый ввод – потенциометр
4. Аналоговый ввод – осциллограф
5. Генерация звука – пьезоизлучатель.
6. Фоторезистор
7. Сенсор на светодиоде
8. Общение с Arduino - программирование работы с COM-портом
9. Аналоговый датчик температуры – LM335
10. Подключаем к Arduino мышку PS/2

Протокол 1-Wire и iButton

Arduino и эмулятор iButton

Arduino и температурный 1-Wire датчик DS18S20

Arduino и драйвер двигателей L293D (Простой мотор-шилд)

Сборка мощного моторшилда (на базе L298)

Arduino и сервомашинка

Как с помощью Arduino/CraftDuino можно управлять устройствами на 220В

Подключаем LCD-дисплей на базе HD44780 к Arduino (Графический экран WG12864B (и ему подобные на ks0107/ks0108))

ИК-датчик препятствий для Arduino на базе фототранзистора

Датчик измерения расстояния SHARP-GP2Y0A02YK0F

Ультразвуковой датчик измерения расстояния HC-SR04

SPI и Arduino

Теория, вывод, ввод

Подключаем кучу устройств к Arduino по 5 проводам

Делаем ISP-программатор из Arduino

Каскад входных сдвиговых регистров

Bluetooth-модуль HC-05

Энкодер и шкала

Датчики Холла

TC15-11 - матрица сетодиодная 8x8

CraftDuino и RFID-модуль RC522

Сервы

Сервы

Сервы 2. Взгляд под корпус (SG-5010)

Подключение сервомашинок к Arduino

Хаки сервомашинок. Переделка в серву постоянного вращения

CraftDuino

Описание CraftDuino и её отличий от Arduino

Дополнительные материалы

Шпаргалка по Arduino.

Книги про Arduino

Выводы Arduino и Arduino pin mapping

Блок питания Arduino из зарядки сотового телефона

Программаторы

Прошивка бутлоадера

Программирование Arduino - Физика процесса.

Про фьюзы

Хаки Arduino - Экономим RAM

ATmega - использование памяти

Fritzing - САПР для Arduino.

VirtualBreadboard - эмулятор Arduino

Игра "Жизнь" на Arduino

Самодельный датчик цвета

Интерфейсы

1-Wire

SPI

Интерфейс I2C

Arduino и ...

Arduino и Processing

Arduino и Matlab

Arduino и LabVIEW

Arduino и Scratch (S4A)

Arduino и openFrameworks

Статьи

Управляемая веб-камера

Мозги для робота

Новогоднее исследование ADSL-роутера на базе Linux - введение

Робософт - обзор существующих решений

Управление роботом-пылесосом iRobot Roomba с помощью Arduino

Исследование Wi-Fi-роутера TP-LINK TL-MR3020

Нейронная сеть

Нейронная сеть - введение

Обучение ИНС с помощью алгоритма обратного распространения



Все проекты с Arduino http://arduino-projects.ru/

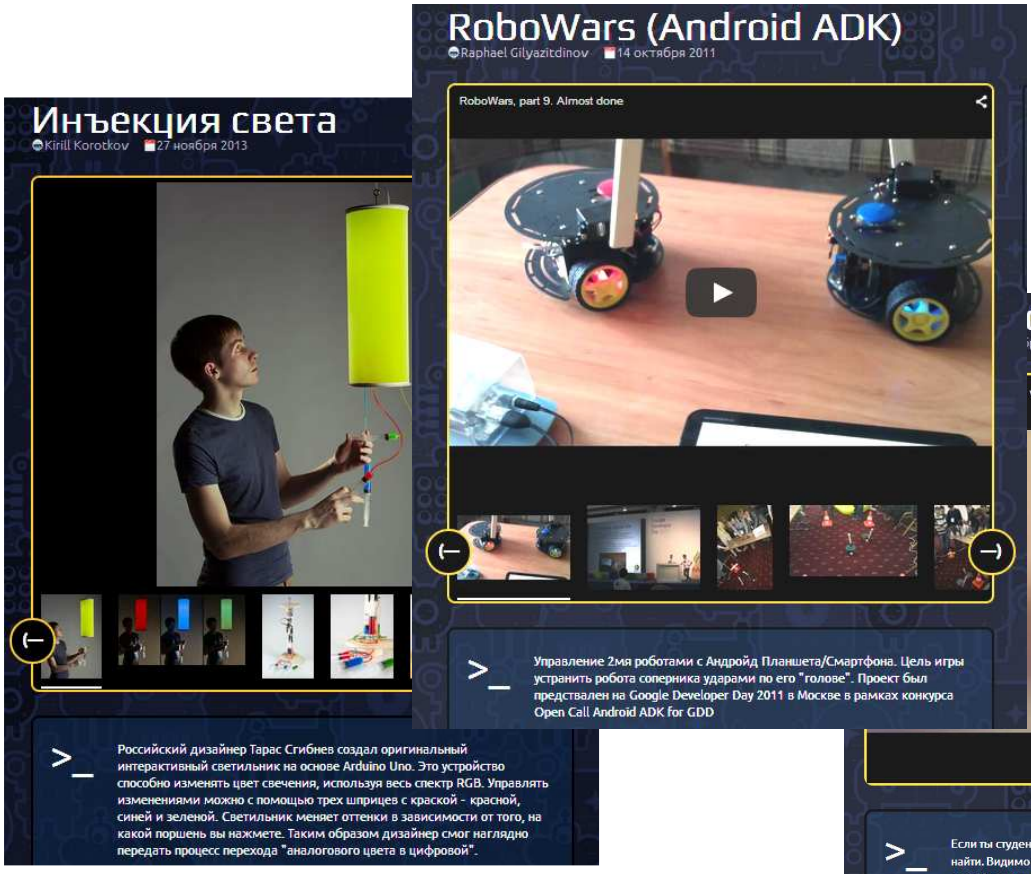
Из Arduino можно сделать много всего разного. Действительно много и действительно разного. Собрать все проекты вместе под одной крышей — это и есть цель каталога arduino-projects.ru. Здесь вы найдёте пополняемую подборку из более, чем 350 проектов. Сможете найти самые интересные устройства, получить вдохновение, перейти к подробностям сборки, добавить ссылку на свой собственный проект и поделиться им с друзьями.

ПРОЕКТЫ (отсортированы по «крутости»)

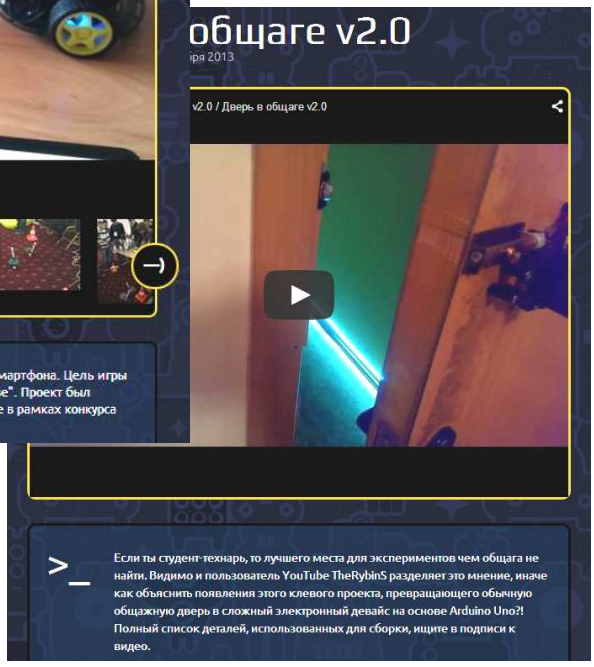
Светодиодный куб 8×8×8
Замок открывающийся на секретный стук
Управление отоплением в загородном доме через GSM
Лазерная арфа
Дверь в общаге v2.0
Радиоуправляемый вертолёт + Kinect + Arduino
Инъекция света
Jazari — музыкальная группа одного человека
Fire Hero
Автоматическая турель из Portal
Шарикоподшипниковый секвенсор
Простой Bluetooth Tank
LED-куб 16x16x16
Аэрохоккей с 3D-принтером
Подсветка лестницы с помощью Arduino
exTouch
Марио на светодиодной панели 8×8
Disco Planet
3D сканер своими руками.
Football Hero
Шкатулка с GPS-секретом
Часы на скетч-доске
R2D2 на Arduino
Программируемый танковый бой
Фольга и Arduino
Воздушная ударная установка
Открывай дверь одним пальцем!
Кормёжка собаки через Твиттер
Arduino-туча
Усовершенствование дверного звонка
Автоматический полив растения
IRIS
Автономная турель-часовой
Квадракоптер
Поворотный стол для 3D макросъемки на Arduino
Wifi Ardu Spy Tank
Spazzi: танцор на соленоидах
Plotclock - нарисованное время
Z-RoboDog - робопёс
Миниатюрный квадрокоптер
Магнитобот
Skube
Nebulophone — мини-синтезатор на Arduino
Рука-робот InMoov
Симулятор кокпита самолета Baron B58
Back to the Spectrum
3D mapping
Doodle Bot
Умная лампа
The DIWire — принтер, печатающий проволоку
RoboWars (Android ADK)
Кнопочные ковбои 2.0
Рисующий тостер
Èlka
Музыкальные дисководы
Старомодный принтер твитов
MaKey MaKey
Прокачай свой велик
Квадрокоптер на печатной плате
Робот-паук ArduSpider
Dingoo A320 с джойстиком SNES
Магнитная левитация
Обучающаяся роборука
Робот "Inchworm"
Высокоскоростная фотография
Многофункциональный кодовый замок на Arduino
Морской бой
Крылья Архангела
Резинки заряжай! Товсь!..
Электронная система управления подвеской
Музыка и желе
Система отслеживания движения глаз
Arduino-ороситель лужайки
NES Keytar. Восьмибитная гитара

Самоуничтожающийся стул
Аквариум контроллер
Hexy
Metatron
Терменвокс на Arduino
Particles
Biofeedback-контроллер
Автономная печатная машинка
Игра Ping Pong
Умный аквариум
Будильник поливающий водой
Контроллер для Angry Birds
Сборка LED-куба
Счётчик пчёл
Дверной молоточек, повторяющий стук
Термохромные часы на Arduino
Игра "Скалолаз"
Психоделичные коты от Random Studio
RFID-кофемашина Juraduino
Arduino тыква
Плавучий змий Sneel
Arduino + Социальные сети + Алкоголь
Face tracking
GPS-ботинки
ArduSat: Arduino в космосе
WiSpyBot
Moppy
Генератор мыльных пузырей
Yahoo-ферма
Модель тяжелого бластерного пистолета DL-44
Робот-древолаз
Элкетрогитара - клавиатура
Destroyer 3000

Pong часы
LED-змейка
Песчаный архитектор
ADK Arduino LedMatrix
Фрезерный станок на Arduino
Робо-лодка
Светофор на Arduino
Светодиодный куб 4x4x4
Робот-барабанщик
Автономный металлодетектор
Диско-шар
Счётчик расхода воды с выводом в интернет
Переливающийся столик
Погодная метеостанция на ардуино
Solar Tracker
Рисующая робо-рука
Lightplot — световой принтер
Трикоптер
Таблично-волновой синтезатор на Arduino
Artbot
SITWAY
Робот, рисующий граффити.
Tree-Climbing Robot
Пульт для составления музыкальных плэй-листов
Робот для игры в Rock Band на iPhone
GMail-нотификатор
Маяк, светильник и книжный шкаф
Дешифратор перфоленты (код ISO-7 Bit)
Робот-платформа с манипулятором
Динамическая фоновая подсветка для монитора
Рентгеновский сканер
Термостат
Матричный генератор звуков



amperka.ru
Амперка





Паяльник http://cxem.net/

СМИ "Сайт Паяльник" посвящен радиоэлектронике. Здесь вы найдете большое количество радиоэлектронных схем, статей для начинающих, программ, он-лайн калькуляторов, обзоры и адреса магазинов радиодеталей, сможете общаться на форуме и многое другое, т.е. все то, что необходимо радиолюбителю или профессионалу. Основной целью сайта является популяризация современной радиоэлектроники в мировом сообществе. На сайте любой желающий может опубликовать свою авторскую статью или добавить видео. За публикацию материалов мы выплачиваем гонорары и выдаем свидетельства о публикации в СМИ.

Arduino http://cxem.net/tags/Arduino/ http://cxem.net/arduino/arduino.php

СТАТЬИ и ПРОЕКТЫ

Управление светодиодной лентой с помощью ТВ-пульта и Ардуино

Розетка, активируемая звуком

Светодиодный диско-пол на Arduino

Простой индикатор радиации на Arduino

4-канальный вольтметр с ЖК-индикатором на базе Arduino

Запрограммируйте Arduino с помощью Android устройства!

Светодиодные часы на Ардуино

Музыкально-световой клавесин на Intel Galileo

Измеритель индуктивности на базе Arduino

EGYDuino – клон Arduino своими руками

Сигнализатор замерзших труб на Arduino

Умный дом с Arduino

Умный магнит для холодильника на базе Arduino

Вендинговый разменный автомат на Arduino

Сейф, который распознает цвета

Измерение освещенности на Arduino и вывод на дисплей Nokia 5110

Аудио спектроанализатор на RGB-ленте WS2812

Управляем любой ДУ техникой дома по ИК с веб-страницы

Контроль положения контактов 3-х позиционного переключателя с помощью Ардуино

GSM охранная система для дома на базе Arduino

Отладочная плата для устройств на МК Atmega8/48/88/168/328

Arduino Uno. Подключение ИК-приемника

Метеостанция + построение графика + C#

Тетрис на базе Arduino и двухцветных светодиодных матриц

Делаем сами Arduino Uno Mini

Подключение GPS приемника EB500 к Arduino UNO

Аргентина - Ямайка 5:0

Автоматический контроллер температуры и влажности для домашних питомцев на базе Arduino

Часы на Ардуино

Прием сигнала от нескольких передатчиков 433МГц на Arduino

Подключение модуля датчика газа MQ-2 и Arduino

Arduino, датчик движения и релейный модуль

Робот-платформа с манипулятором на ATmega

Nanino - самодельная Arduino

Управление iRobot Create с помощью беспроводного геймпада через Arduino

Емкостная сенсорная Midi клавиатура

Текстовой дисплей для удаленной связи с офисом на базе Arduino Uno

Настройка модуля часов реального времени RTC для Arduino

Проигрывание WAV-файлов при помощи Arduino

Светодиодная матрица 24x6

Поющее растение на базе Arduino

Метеостанция на Arduino и MR3020 для народного мониторинга

Радиочастотное (RFID) управление доступом с помощью Arduino UNO и модуля EM-18

Ошейник для кота с GPS навигацией на базе микроконтроллера TinyDuino

Необычный кодовый замок на Arduino

Объект String в Arduino и команды через последовательный порт

Голосовое управление радиорозетками UNIEL

Система контроля параметров комнатных растений

Пульт дистанционного управления ПК на микроконтроллере ATtiny85

Компания Microchip представила новую макетную плату, совместимую с Arduino™

Arduino Yun интегрирует архитектуру Arduino с открытым кодом с ОС Linux

Сервер домашней метеостанции на Arduino - виджет для OS X

Новая плата расширения для Raspberry Pi

Простой анализатор спектра на микросхеме MSGEQ7

Изготовление электронной нагрузки постоянного тока и мощности на Arduino

BASCOM-AVR и Arduino

Управление камерой, приборами и данные с датчиков на экране телевизора

Процессор Sitara AM335x ARM Cortex-A8 интегрирован на Arduino

Тюнер гитары на контроллере Arduino

CxemCAR на Arduino - Bluetooth управление машинкой с Android

CxemCAR 2 - управление роботом и просмотр real-time видео с Android-смартфона

ИК-термометр своими руками

Поворотное устройство для антенны/web-камеры на ATmega8/Arduino

Контроллер кнопок для игры Что? Где? Когда?

Измерение тока потребления устройств при помощи токовых датчиков серии ZXCT

Робот-паук на сервоприводах + ИК-управление

Подключение монетоприемника к Arduino

Мониторинг параметров компьютера на Arduino

LCD осциллограф на Arduino

Brute Force BIOS'a ПК при помощи Arduino

RGB-ночник управляемый руками

Голосовое управление освещением X10. Части 1 - 5

Сервер домашней метеостанции на Arduino + Виджет на Android. Добавление датчика BMP085

Новая платформа Embedded Pi включает в себя Raspberry Pi, Arduino и 32-битные встроенные ARM

Arduino представила Arduino GSM Shield

Сервер домашней метеостанции на Arduino + Виджет на Android для вывода данных

pcDuino - мощная плата разработчика на процессоре 1ГГц ARM Cortex A8 совместимая с Arduino шилдами

Игровая ТВ-приставка на Arduino. Части 1, 2

Робот-танк, управляемый акселерометром

Робот, следующий по линии на Arduino

Сенсорная панель управления

Информационный экран для iPod

Обзор FEZ Panda II

Немного о FEZ и .NET Micro Framework

Подключение I2C EEPROM к Arduino

Интервалометр с LCD для Sony NEX

Тахометр на Arduino

VGA на Arduino

Передача данных по Bluetooth между Android и Arduino

Arduino и Bluetooth

Генератор сигналов на Arduino

Arduino своими руками с USB портом

Велосипедный спидометр на Arduino

Arduino IDE - введение (видеоурок)

Arduino UNO урок 1 - Мигание LED

Arduino UNO урок 2 - Управление сервоприводом

Arduino UNO урок 3 - Тайминг

Arduino UNO урок 4 - Бегущий огонь

Arduino UNO урок 5 - Fade-эффект

Arduino UNO урок 6 - Энкодер

Arduino UNO урок 7 - Пьезоизлучатель

Arduino UNO урок 8 - Ночник

Arduino UNO урок 9 - Нагрузка

Arduino UNO урок 10 - LCD

Arduino UNO урок 11 - Serial LCD

Arduino UNO урок 12 - Джойстик

Arduino UNO урок 13 - драйвер двигателя L298N

Arduino UNO урок 14 - цифровой компас HMC5883L

Световой диммер управляемый Arduino

Android и Arduino. Обмен данными

Android и Arduino. Привет Android из Arduino

Android и Arduino. Программное обеспечение

Android и Arduino. Введение в ADK

Arduino: спасибо и прощай

Поворотная вебкамера на Arduino

ATX блок питания управляемый Arduino

Блок питания для Arduino из ATX

AC диммер на Arduino

Apple Remote Shield на Arduino

Светодиодный куб 4x4x4 на Arduino

Цифровой вольтметр на Arduino с подключением к ПК через последовательный порт

TV-выход на Arduino

Пробуждение ПК по сети на Arduino

Простой тестер емкости Li-ion аккумуляторов

Часы на Arduino с использованием стандартного индикатора

Arduino лазертаг

Устройство мониторинга сейсмоактивности

Измеритель ёмкости аккумуляторов (Li-Ion/NiMH/NiCD/Pb)

Анализатор спектра звука

USB MIDI контроллер на Arduino

3D интерфейс ввода на Arduino

Система распознавания и слежения за лицами на Arduino

Ambilight своими руками

Мониторинг электроэнергии на LabView

Динамическая подсветка ТВ

Подключение PS/2 клавиатуры

Shell на Arduino

Солнечный трекер

Запись/чтение на SD-карту с Arduino

Недорогой тепловизор своими руками

Передача MIDI данных от Arduino в компьютер

Считываем показания датчика DS18B20

DDS-генератор синусоидального сигнала

Инфракрасный датчик движения (PIR сенсор)

Электронный барометр своими руками

Парктроник своими руками

Счетчик расхода воды

RGB LED Pixels

Ножная Midi педаль на Arduino

Lo-Fi гитарная педаль на Arduino

Устройство вокальных звуковых эффектов на Arduino

Arduino-синтезатор

Шторы с электроприводом и ПДУ

Тестер AA аккумуляторов

Геркулес – машинка управляемая Android телефоном

Простой робот SPROT

Робот Боб 2.0

Автономный танк на Arduino

Робот, проходящий лабиринты

CARduino – самодельный робот на Arduino

Четвероногий робот на основе Arduino

Машинка 4x4 на Arduino с управлением по WiFi с Android

Плавающий робот из бутылок своими руками

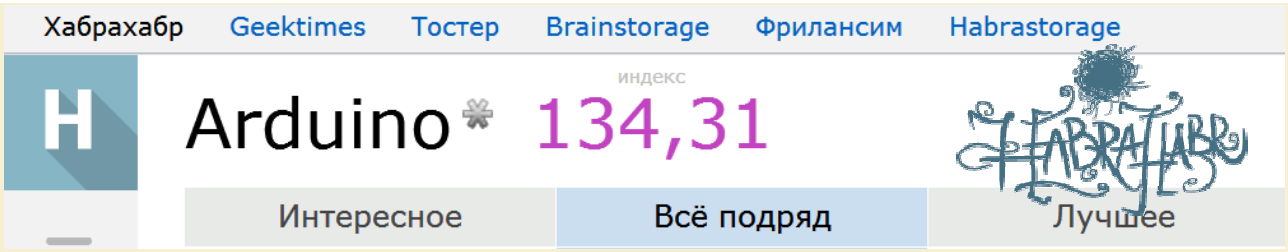
Лодка на Arduino с датчиками

Автономный робот с авто навигацией на Arduino

Drawbot

FijiBot

Робот, лазающий по деревьям



Хабрахабр http://habrahabr.ru/

«Хабрахабр» (или «Хабр») — крупнейший в Европе ресурс для IT-специалистов с ежемесячной аудиторией более 10 млн уникальных пользователей. В «Хабрахабр» заложена уникальная модель совместного творчества, позволяющая людям собирать и структурировать информацию, выделяя из неё наиболее полезную и ценную. Используя заложенные в проект механизмы, пользователи самостоятельно наделяют правами друг друга, давая или ограничивая возможности.

Глобальная цель проекта — помогать людям развивать профессиональные навыки и информировать друг друга о происходящем в индустрии информационных технологий.

Arduino http://habrahabr.ru/hub/Arduino/

СТАТЬИ

Знакомство с Arduino. Введение

Подготовка и запуск "Hello, World!"

Arduino Duemilanove

Как сделать библиотеку для Arduino? (пример прилагается)

Старые друзья шиворот-навыворот, или как светодиод может быть фотодатчиком

Chronulator: вторая версия на микроконтроллере Atmel ATmega168V

Панк-рок — на два метра левее

Шпаргалка по Arduino

Индикация о новом сообщении Gmail через Arduino

Arduino: первое знакомство

Знакомство с Arduino, часть 2. Морзе-клавиатура: альфа-версия

Знакомство с Arduino, часть 3. Морзе-клавиатура: бета-версия

Робот радуется сообщениям в твиттере :)

Удобный регулятор громкости для ПК

Ethernet термометр на основе Arduino

Touchscreen от Nintendo DS и Arduino

Получение и вывод GPS координат на Arduino

FTDI анонсировала платформу на основе Vinculum-II

Scratch и визуальное программирование Arduino

Обновления проекта Arduino

Самодельный стабилизатор для цифровых камер

Toolduino

Arduino-терминал

Квадрокоптер своими руками: Части I и II

ArduIMU — коптер на Arduino

Сверхточный электронный компас и датчик положения

yet another serial protocol, для людей

Работа с Arduino из AVR Studio 4

Добавление метода ввода к Arduino

Самодельный SD Card Shield для Arduino

Перчатка SudoGlove: управление машиной движениями пальцев

Настройка Eclipse для работы с Arduino Uno

Программируем на C/C++ в Visual Studio под Arduino

Ловим горизонт с Arduino

Трех-степенной манипулятор на Arduino

Непрерывный мониторинг радиационного фона в Москве

Робот «Inchworm» на базе Arduino Nano

Open Call или 10 Android/Arduino идей для Google Developer Day 2011

Универсальный триггер для фотокамеры

O'Bakas одночипный Arduino

Orbduino — механическая рука управляемая через интернет

Arduino? – Нет! IDuino – Да!

Контроллер для домашней пивоварни Mega Brewery

Siri отпирает дверь

Arduino в Visual Studio

Очередной проект на Arduino: «хищный» дисплей

Шести-степенная жизнь на хвосте питона

Простые бинарные часы с будильником на Arduino

«Бегущая строка» — динамическая индикация на примере светодиодной матрицы RL-M2388 и Arduino Mega

И снова Arduino: принтер, следящий за Twitter-каналом пользователя

Новые Arduino: апаратный USB в контроллере, ARM-архитектура и WiFi

Простой синтезатор на Arduino

Teagueduino — новая альтернатива Arduino «для чайников»

Самодельный расходомер для автомобиля

BlackBerry PlayBook защитит ваше молоко от пришельцев

Обзор книги «Программирование микроконтроллерных плат Arduino/Freeduino»

Релиз IDE Arduino 1.0!

Рисунки на Etch-a-Sketch с помощью игрового контроллера на основе Arduino

Самый нужный шилд для Arduino!

Arduino по-китайски или штангенциркуль по-Ардуински

Metaboard на проволочках, или Arduino Skeleton

«Камера» из светодиодов

Копируем данные с CD/DVD на жесткий диск при помощи LEGO и Arduino

Использование инерциальной навигационной системы (ИНС) с несколькими датчиками на примере задачи стабилизации высоты квадрокоптера

Механический дисплей на Arduino — работа над ошибками

Обновляем скетч по воздуху

Уведомление о непрочитанном входящем письме с помощью Arduino

Управление Arduino с телефона

Arduino в кофейном автомате

Duck Hunt на основе Arduino

Сверхдешёвая камера с управлением — своими руками

Механический дисплей из Лего и Arduino

Подключение светового пистолета через Arduino

Мозги моего робота — Arduino, первый опыт

Ну очень недорогой робот телеприсутствия на основе ноутбука и мото-тележки. Продолжение

Ускоряем свою Arduino

Автоматизированное копирование 3.5" дискет на основе Arduino

Шкатулка с музыкальным замком

AVRDudeR — встраиваем avrdude в IDE

Bluetooth термометр

Бюджетный Serial to Ethernet адаптер за один вечер

Arduino LCD Informer

Распознаём изображение с токена при помощи камеры

Автоматическая подсветка лестницы с помощью Arduino

Использование радио модулей RF-2400 от Inhaos

Наглядный мониторинг большого числа обьектов (например, каналы IPTV) с помощью Arduino

Простой робот с ультразвуковым дальномером

Панорамы: быстро или качественно? И то и другое!

Zyxel Keenetic 4G, Arduino и датчики температуры ds18b20

Практическое использование термистора с Arduino

Онлайн-шахматы на реальной доске на основе Arduino

Заклинатели «змейки» на Nokia 6110i и Arduino

Hexy — сборный робот на основе Arduino

Олли — автономный роводирижабль

Дистанционный пульт DSLR для ночной фотосъёмки

Трикоптер на Crius MultiWii контроллере

Подключение сенсора освещения от Mac Book Pro к Arduino

Использование Arduino UNO в качестве программатора

Оптический выключатель со звуковым эффектом на Arduino

Виноградный мультитач

Соединение Raspberry Pi и Arduino

Arduino в снековом автомате

Альтернатива Arduino от Texas Instruments за $4.30

Подключаем джойстики от Dendy к PC через Arduino

Google I/O — Презентация ADK 2012

Программируемый выключатель света с дистанционным управлением

Контроль температуры воды в газ. колонке

Я презираю Arduino

О полетном контроллере MultiWii (коптеры, самолеты и вертолеты)

Arduino в космосе. Осталось 23 дня, чтобы заказать 3 суток аренды спутника за 325$

Как считать сигнал с аналоговых входов Teensy++ и вывести их на экран

leOS — многозадачное ядро для Arduino

Arduino будильник управляемый из браузера

Как рождаются Arduino

Управляем самодельными железяками по воздуху при помощи Open Sound Control

Поддержка C++ на avr в gcc

Вышла новая версия Energia

Биометрическая идентификация по рисунку вен ладони (mini How To)

Оригинальный замок для двери на основе Arduino и ёмкостного датчика

Управляем чайником из браузера или как я интернет-розетку делал

Subaru и Arduino: протокол SSM1. Рукопожатие

Mozilla запускает Codebender — среду разработки для программинга микроконтроллеров

Счетчик скорости вращения колеса белкой

Sparkfun.com FREE day или *как вы относитесь к recaptcha*

Master-key для замков Onity на Arduino в обычном маркере

Многоканальный программный ШИМ в AVR

ИК пульт для ПК и розеток. Часть 1

Альтернатива Wiring для Arduino — BASCOM-AVR

Недорогая альтернатива Arduino. Дадим шанс?

Делаем сами Single-Side Arduino с COM-портом на борту

Автоматическая газель на Arduino

Создание робота телеприсутствия

Самодельная подсветка на квадрокоптер

Светодиодная новогодняя ёлка на Arduino за один вечер

Разработчик конструкторов Arduino названа предпринимателем года в США

Заказ производства печатных плат на заводе — step by step

Arduino Esplora

ЛУТ на виниле или домашняя Arduino Mini

Свежий взгляд на микроконтроллеры

Роботизированная лампа в стиле Pixar на основе Arduino

Роботизированный бас на Atmel ATMega328 и Arduino IDE

Работа с Arduino

Arduino <> Firmata <> Visual Basic .NET

Простое устройство мониторинга энергопотребления

Arduino: Робо-рука на двух сервоприводах

Теория и практика пассивных пироэлектрических датчиков или как сделать индикатор направления движения

Arduino: настройка при помощи DHCP

Оптимизация преобразования HSV в RGB для микроконтроллеров

Ford выпустил SDK для своих автомобилей

Управление декоративной подсветкой на Arduino с телефона

Arduino&Oregon или погодная станция своими руками

Радар на Arduino

Конструктор для сборки сотового телефона

Бейсболка-сонар для слабовидящих

Программирование Arduino из консоли, gentoo-way, ничего лишнего

Самодельный тепловизор на базе Arduino менее чем за 100$

Программирование Arduino из Linux, gentoo-way, быстрый старт

10 девайсов, которые вы можете проапгрейдить, немного поковырявшись с электроникой

Ethernet метеостанция

Включите мой комп!

Rainbow Cube — куб с RGB диодами

Проект GPS-автомаяка с открытым кодом

Домашний мини-климат-контроль своими руками

Хилый укроп и рукотворное солнце

Arduino: Автоматическая подсветка лестницы

Используем Twitter по назначению

Где разработчики настолько суровы, что в хакатоне побеждает проект массой в 100 кг?

Arduino Robot: открытая платформа для сборки роботов

Искусственный интеллект на базе Arduino

Из пункта А в пункт Б

CubieBoard дорого, Raspberry Pi слабовато, или где взять SoC с GPIO помощней да подешевле

Arduino Yún — Wi-Fi и Ethernet при нём

Замкнутая экосистема по-русски

Arduino watchdog или автоматический RESET в случае зависания

Как не переплачивать за Arduino и модули к нему

Беспроводной Arduino с радиусом действия более километра

STM32 vs Arduino

Разгон Arduino. Под жидким азотом. 20 ☐ 65.3Mhz @ -196 °C

Unreal LED, или управление нагрузкой из Unreal Tournament

Мигающая строка из светодиодов для стоп-сигнала на Arduino Pro Mini

Работа с сенсорным экраном на Arduino DUE

Scratch for Arduino делает программирование под Arduino более наглядным

Работа с TFT дисплеем на ARDUINO DUE

Опыт постройки коптера: WLtoys V949 + MultiWii 2.0

Биопринтер своими руками

Microduino — что если всё взять и поделить?

CodeBender.cc — программирование Arduino из браузера

Создание автономного робота Frank. Части 1 - 3

Делаем автоспуск почти для любого фотоаппарата

«Achtung!», или Мониторим состояние сборки проекта

Новогоднее поздравление от робота

Первая передача текстового сообщения с помощью запаха

Автомат для сортировки по цвету M&M's и Skittles на Arduino Uno

Делаем елочные игрушки из светодиодных лент

Здание превратили в работающий «Кубик Рубика» при помощи Arduino

Чертежи для самостоятельной сборки мобильного телефона

Беспроводные коммуникации «умного дома». Часть вторая, практическая

Galileo — первый Arduino-совместимый микрокомпьютер на платформе Intel. Уже в продаже!

Аналог ambilight из LED ленты WS2812, Arduino и киндер-сюрприза

Управляем веб-камерой с помощью джойстика

Аппарат кормления котов управляемый, пассивный АКК-УП-1

Контроллер центральный домашний, всемогущий КЦД-В-2-12

Делаем автополив комнатного цветка на Arduino за 15 минут

Мой удобный дом

Arduino ли единым

Визуализация с RGB-полоской

Робот на базе: Android, Arduino, Bluetooth. Начало

Робот на базе: Android, Arduino, Bluetooth. Рефлексный. Часть 2

Вывод звука на Arduino Due

x86-compatible, часть 3: «86Duino наносит ответный удар»

Опыт создания робота

Автомат света и музыки АСИМ-АУ-2-6

Радиоуправляемый выключатель своими руками. Часть 1 — Hardware

Радиоуправляемый выключатель своими руками. Часть 2 — Тестирование и подготовка

Радиоуправляемый выключатель своими руками. Часть 3 — Software выключателя

Радиоуправляемый выключатель своими руками. Часть 4 — «Центр управления»

Как я стал преподавать Arduino

Fish on Wheels: самоходный аквариум с золотой рыбкой на основе Arduino + Beagleboard

«Заводим» радиоуправляемые розетки без пульта

Робот для игры в воздушный хоккей из частей для 3D принтера

Зубочистка-детектив раскрывает секрет радиопротокола

Прибор измерительный температуры и влажности ПИ-ТВ-2

Arduino в руках дизайнеров

Визитка на Arduino с «Тетрисом» на борту

Светомузыка Processing + Arduino + RGB

Термостатирование в доме

Блок дистанционный сервисный многофункциональный БДС-М

Мощный лабораторный блок питания

Погодная станция с Ethernet и планшетом в качестве устройства отображения

Управляем Intel Galileo через облако

Погодная станция на Ethernet (HTTP+Modbus) с питанием по POE

Альтернативные аркады на GDC: галерея сумасбродных контроллеров домашнего производства

Управляем машинкой через Bluetooth с планшета или телефона под Android

Зрение для Arduino, или как работать с видео имея 2,5 кб оперативной памяти

Катаемся на флоппике и изучаем перспективный роутер. (В этой статье не пострадал ни один TP-Link MR3020 и TL-WR703N, и отделался легким испугом HAME MPR A100 (A2))

Arduino. Лучшая книга по мнению читателей

Arduino: ИК-управление бытовой техникой (применение девайса)

Отладка кода Ардуино в кристалле

Arduino: ИК-управление бытовой техникой

Arduino и аквариум

Как нас било током на 1 апреля

Краткий обзор 6 бюджетных колесных платформ для роботов с Ebay

Уменьшаем количество проводов в Arduino — I2C LCD экран и RTC часы по двум проводам

Йогуртница на основе Arduino

Установка и сборка образа Angstrom Linux для платы TechNexion Thunder

Настройка Intel Galileo с нуля и до установки полной версии Debian

Как гироскопом на руле машину измеряли

Автоматическая кормушка для птиц

Обзор роботов для настольных игр

Cubietruck & Raspberry Pi & Arduino

Измеритель радиоактивности (радиометр) с Глонасс приемником и передачей данных по Wi-Fi

DIY: Универсальный Ambilight для домашней мультимедиа системы — Атмосвет

Обзор домашнего робота ver 0.3

Испытания домашнего робота версия 0.3.1

Прошивка Arduino Pro Mini через Nano

Arduino управляет промышленным грузовым лифтом

Создание робота балансера на Arduino

Arduino UNO играет в 2048 с размером поля 8x8

Photopizza — открытый проект поворотного предметного стола для 3D-фото

Игрушка на Arduino: Тук-тук шкафчик

Работа с инфракрасным дальномером Sharp

Управление двигателями через Wifi Bee от Seeedstudio

Резистор изгиба своими руками

Управление RGB лентой с помощью Arduino и драйвера L298N

Основа для умной комнаты, или как Arduino в общежитии живёт

Как я сделал Wi-Fi весы, ниоткуда не уволился, а про жизнь вообще молчу

Как сделать робота за 24 часа

Модельно-ориентированное проектирование на коленке, идентификация систем в MATLAB/Simulink

Комнатная метеостанция на Arduino

Метеостанция: от идеи до реализации

Ethernet метеостанция 2 – Продолжение…

Автоматизация смыва унитаза на Arduino + Z-Wave

Педалируем Vim

Спидометр для скейта. Безысходность

Самодельный велокомпьютер, или как Arduino на улицу попал

Простое удалённое управление с компьютера роботом

Хитрая канистра

Программируемый робот Brownie версия 0.4

Астротрекер за два вечера

Программируем квадрокоптер на Arduino

Применение самодельного тепловизора на базе Arduino в исследованиях экономии электроэнергии

Диафильмы на шарманке: грабли, шишки и подводные камни

Загрузка скетчей в Arduino через Bluetooth

Как скрестить конструктор «Знаток» и Arduino своими руками

Расширяем интеграцию Arduino в детский конструктор Знаток

Музыкальный дверной звонок в стиле Star Wars на Arduino

Прошивка и программирование ATtiny13 при помощи Arduino

Управление яркостью подсветки 7-ми дюймового TFT LCD

Запустите Windows 8 на своем Intel Galileo

Светодиодный «смайлик». Технология мультиплексирования методом «Чарли»

Дверной замок. RFID

Автоматизация снятия показаний со счетчиков воды

Удаленное управление VLC player'ом при помощи Arduino и Python

Управляем розеткой по SMS

Модифицирование преобразователя USB-UART на чипе CP2102 для использования в качестве программатора Arduino

Станция слежения OpenDNS дата центров

Как я логгер собирал на SD card Shield V2.0

Идет сбор средств на закупку тепловизионного модуля Flir Lepton

Обзор Makeblock Starter Robot Kit V2.0. Распаковываем

Как мы заняли второе место на соревнованиях MATEC ROV 2014 в США

Пара костылей для Notepad++ Arduino IDE (для Чайников)

Выставка 3D PrintShow в Лондоне. Индустрия — золотая жила или охота за привидениями?

FTDI наносит ответный удар

Ввод пароля или похитители времени

Автоматический перебор комбинаций в механических сейфовых замках

PcDuino3 Nano, серьезный конкурент Raspberry Pi B+, за $39

Программирование Arduino с помощью ArduBlock на примере робота, движущегося по полосе

Проект Dronecode: свободная платформа для беспилотных аппаратов

Брутфорсим EFI с Arduino

Перепрошивка сетевого хранилища NS-2502 от Edimax через консольный кабель

Маленький Hello World для маленького микроконтроллера — в 24 байта (и чужое решение в 12 байт)

Подключаем Arduino к счетчику электроэнергии

Термокоса под управлением Arduino и LabVIEW

Электронный «шар судьбы» на ATtiny13

Как программисты решают проблемы

Хочешь вкусного пива, Arduino в помощь

STM32 Nucleo. Подключаем TFT LCD на базе чипа ILI9341

Воскрешаем FTDI в картинках

Подключаем датчики к Raspberry Pi без проводов и с Arduino

Ардуино на службе здоровья

Arduino на службе здоровья — 2

Безынерционное измерение температуры воздуха ультразвуком

Arduino, модуль Nokia 5110 LCD и кириллица

Безопасное использование ножки RESET на Arduino

Питание Raspberry Pi с использованием Arduino

Спутник на балконе

Бинарные часы с будильником и таймером на Arduino Uno

Слежение за статистикой в WoT с помощью Arduino

Строим роботанк с управлением по Wifi, камерой, пушкой, блекджеком и т.д.

Реле с дистанционным ИК управлением на ATtiny13A

Управление микроклиматом на Arduino

Контроль температуры в серверных шкафах с помощью Ардуино

Белый Куб на страже чистоты воздуха, части 1, 2

Простейший кардиограф на Arduino

Приставка MIDI для синтезатора (taptempo)

Электронный тортик в кармане: дневник разработки

Зимовка кактусов с онлайн контролем температуры

Разработка Электрофокусера на базе отладочной платы Arduino Uno, части 1 - 3

Простой встраиваемый усилитель НЧ на микросхеме с FM приемником на основе Arduino

Альтернативная среда программирования для Arduino — FLProg

FLProg – альтернативная среда программирования Arduino. Описание проекта

FLProg.ru – лицо программы FLProg в интернете

FLProg и пингвины

Бесплатная отправка SMS из Arduino без GSM модуля

Светильник декоративный бытовой СДБ-3 «Евлампия»

Arduino за 1$

Программирование Arduino с помощью ISP программатора

Robotale: радиоуправляемая машинка с Arduino и Bluetooth, которая поможет изучить основы работы с Arduino и не только

Новый Год, гирлянда, Arduino

Плоттер на основе конструктора Makeblock

Превращаем Arduino в полноценный AVRISP программатор

Попытка подружиться с STM32

Считывание данных с помощью Arduino и Processing. Обработка с помощью R

Программно регулируемая люстра на Arduino

Программирование Arduino в CLion

YACTL: Ещё одна ёлочная гирлянда

Школьный звонок на .NET Micro Framework с удаленным управлением

SuperDuino: миниатюрная плата с цветным дисплеем и акселерометром, совместимая с Arduino

Домашний тир на Raspberry

Домашняя автоматизация с openHAB: освещение и удаленное управление обогревателями. Часть 1Maker — ваш DIY-справочник по продуктам Intel

Как подружить OpenHAB и Arduino



KAZUS.RU http://kazus.ru/

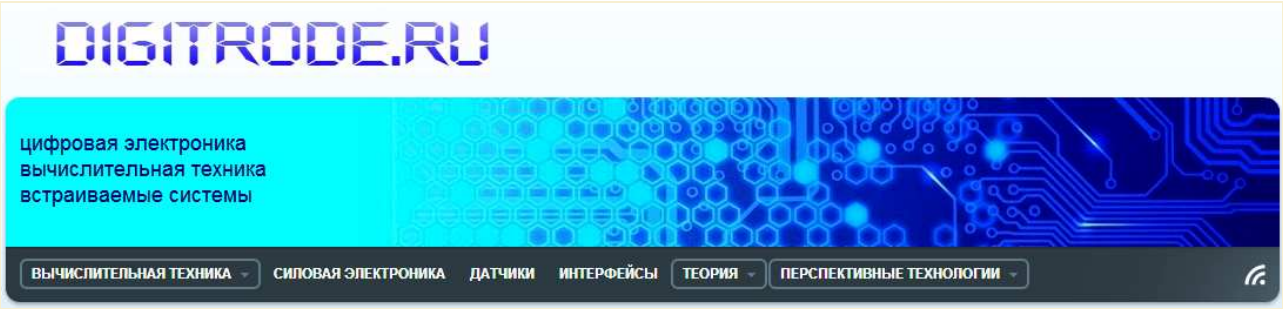
С октября 2003 года наш сайт предоставляет своим читателям объективную, полную, оперативную и самую интересную информацию о мире электроники и электронных компонентов. Благодаря своей независимости, сайт KAZUS.RU является одним из авторитетных сетевых изданий в области электроники и IT-индустрии.

Arduino http://kazus.ru/shemes/list_pages_categories/1/0/1.html

СТАТЬИ и ПРОЕКТЫ

Arduino UNO урок 1 - Мигание светодиодом
Arduino UNO урок 2 - Управление сервоприводом
Arduino UNO урок 3 - Тайминг
Arduino UNO урок 4 - Бегущий огонь
Arduino UNO урок 5 - Fade-эффект
Arduino UNO урок 6 - Энкодер
Arduino UNO урок 7 - Пьезоизлучатель
Arduino UNO урок 8 - Ночник
Arduino UNO урок 9 - Нагрузка
Arduino UNO урок 10 - Подключаем LCD
Arduino UNO урок 11 - Serial LCD
Arduino UNO урок 12 - Подключение джойстика
Arduino UNO урок 13 - драйвер двигателя L298N
Arduino UNO урок 14 - цифровой компас HMC5883L
Arduino своими руками с USB портом
ATX блок питания управляемый Arduino

Блок питания для Arduino из ATX
Запись/чтение на SD-карту с Arduino
Калибровка аналогового датчика для Arduino
Подключаем датчик температуры DS18S20/DS18B20 к Arduino
Подключение 12 светодиодов к Arduino Mega
Подключение I2C EEPROM к Arduino
Подключение LCD HD44780 к Arduino Uno
Подключение PS/2 клавиатуры к Arduino
Подключение монетоприемника к Arduino
Пробуждение ПК по сети на Arduino
Светодиодный куб 3x3x3 на Arduino
Светодиодный куб 4x4x4 на Arduino
Светодиодный куб 5x5x5 на Arduino
Старинные часы-метеостанция
Часы DS1302 для Arduino Uno с ЖК экраном BTHQ 22005VSS
Часы реального времени для Arduino



digitrode.ru http://digitrode.ru/

Привет всем интересующимся современной электроникой, цифровой техникой, встраиваемыми системами и их программированием!

На этом сайте выкладывается познавательный и обучающий материал, который поможет новичкам в освоении электронной техники, а опытные инженеры, возможно, подчерпнут из него что-то новое для себя.

Arduino – это потрясающая электронная платформа для создания радиолюбительских проектов. Она проста в использовании и намного дешевле профессиональных отладочных плат.

Arduino http://digitrode.ru/tags/Arduino/

СТАТЬИ и ПРОЕКТЫ

Arduino своими руками

Музыкальный инструмент на основе Arduino и желе

Управляем реле с помощью Arduino

Библиотеки Arduino: подключение

Датчик температуры TMP36 и его подключение к Arduino

Питание Arduino от источника напряжения 3.3 В

Arduino и четырехразрядный семисегментный индикатор

Расширяем количество портов Arduino за счет чарлиплексирования

Arduino и шаговый двигатель BYJ48

Система слежения за солнцем на Arduino

Ardubracelet – браслет на Arduino, позволяющий играть в Tetris

Погодная станция на основе Arduino и Wi-Fi модуля ESP8266

Подключение оптического энкодера к Arduino

Домашняя система безопасности с GSM на основе Arduino

Arduino и RFID-считыватель

Arduino и светодиодная матрица 8x8

Управляем светодиодами WS2812 с помощью Arduino

Подключаем к Arduino акселерометр MMA7361

Эмулятор процессора 6502 и Commodore 64 на Arduino

Управляем вентилятором с помощью Arduino

MiniPirate – управляем Arduino с помощью командной строки

Простой таймер на Arduino

Робот на основе Arduino:

- Часть I – управление двигателями
- Часть II – шасси, средства передвижения и питание
- Часть III – подключение, программирование и пробный пуск
- Часть IV – ультразвуковой датчик HC-SR04

Внешний монитор последовательной передачи данных ESM

Разблокировка компьютера с помощью NFC

Простой Arduino-счетчик на 7-сегментном индикаторе

Звуковая сигнализация на Arduino

Очередной клон Flappy Bird. Теперь на Arduino!

Вольтметр на Arduino

JTAG-адаптер для Arduino Micro

Дистанционное управление для Arduino

Сигнализация на Arduino, оповещающая об исчезновении напряжения сети

Управляем квадрокоптером с помощью Arduino

elektor labs

Sharing Electronics Projects

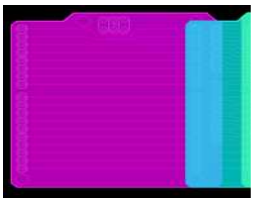
Search

Home | Proposals ➡ | In Progress ➡ | Finished | Log in

Elektor Labs http://www.elektor-labs.com/

Arduino site:www.elektor-labs.com
https://duckduckgo.com/?q=Arduino+site%3Awww.elektor-labs.com&ia

PROJECTS



Arduino Expander
Connect 2 non stackable Arduino shields to one arduino. Attachments: 0 photos | 1 schematics | 1 PCB(s) | 0 software files | 0 other You have to be logged in to open attachments . 9130102822 . Arduino Expander. Please ...



An Arduino'd Radiation Meter
These days no one marvels at seeing an Arduino board in just about any application. Arduino's ease of use makes it perfect to quickly develop DIY measuring instruments. This article shows how to build a simple meter for radioactive levels, ...



Tacho/RPM for CNC etc using Arduino Micro and OLED Display
I needed a tacho/RPM display for a CNC spindle. I decided to use an Arduino Micro and OLED display with a reflective LED/Photo transistor as a sensor.



Wi-Fi / Bluetooth / USB shield for Arduino & Platino
Hello community, I am working on an Arduino shield on which you can either mount a Wi-Fi module or a Bluetooth module controlled through a serial port.



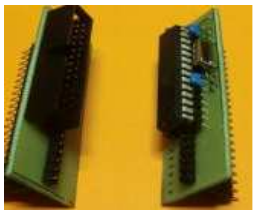
PIC dev board (with programmer)
An arduino is a little board with a microcontroller on it (usualy 2, 1 for programming and 1 for for learning /developing/etc). Most pic development board need an external programmer,



CC2A humidity sensor BOB
We decided to use the more affordable Chipcap2 humidity sensors from Amphenol for this. ... this is why its VCC power supply pin must be connected to an Arduino digital output (PB0) pin, which is controlled by software. In this sketch two constants are defined:



3D Pad - Control Without Touching
Arduino shield for controlling something without touching the controller. Great for Touchless Gesture User Interface experiments, artistic expression and game controllers. Based on capacitive sensing techniques.

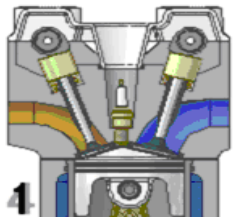


I-Board - poor man's T-board

Wiring the Arduino prototype board to the flatcableconnector is straight forward, with 1 exception: the wires for I/O 9 .. 13 are "crossed". Because there is no software you will not find anything here.



Banc de test pour sondes de température
Banc de test pour sondes de température. Proposal; Tweet . Author: Castor01 | Project status: Proposal | 0 contributions | 1 members | 0 ... Arduino's duties: PID regulation of the temperature's plate measuring of: reference temperature, plate temperature, device under test sending on USB ...

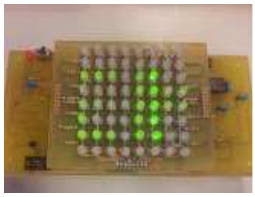


LP Gas (propane) Fuel Injection for Single Cylinder Engines

This is a finished project and comes with details of hardware, software and simple Arduino based electronics to convert a single cylinder 4 stroke engine to a gas fuel injection system.



T-Board - uController Prototyping Simplified
As hobbyists and enthusiasts start hitting the physical constraints of boards like the Arduino, we need take the giant leap into developing our own microcontroller-based boards. The easiest way to transition is onto a breadboard, ...



8x8 2-Color Led Matrix with ATmega328P
8x8 2-Color Led Matrix with ATmega328P (Arduino compatible) [130146-I] arduino; LED; matrix; arduino; Finished; prototyping; laboratory & development; Tweet . Author: Ruben van Leeuwen |



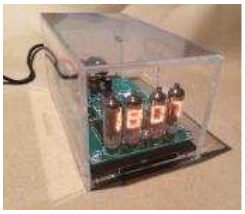
Elektor Wifi RGB Ledstrip

Since Koen used an ATMega328 I ported the code to the Arduino Uno. When I have fully working Arduino code I will de-Arduino it. For the moment it only relies on the Arduino Serial port class and the analogWrite function.



Voltage-level shifter for 3-state output
Fortunately, Arduino boards have a 3.3V regulated output that can be used to power the 23K256 chip. It is also quite easy to adapt the SS, SCLK and MOSI signals with simple 2-resistor voltage dividers, as they are 2-state Arduino -> 23K256 signals.

Yet Another Button Cell Charger
That controller should be a ATMEGA328P, the heart of the Arduino UNO. Eight: if possible the whole proposal is to be realized on a holes- or strips-print, avoiding high costs for a dedicated PCB. Hardware First I discuss the universal button cell placeholder.

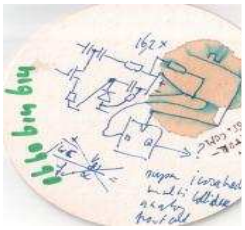


Numitron Arduino Clock and Thermometer

If the concept of Arduino powered proper projects as a next step for Arduino fans (and AVR fans) is of interest, I have 5 or 6 other projects that follow the same format (Proper project but Arduino IDE programmed) and I'll be happy to publish those too.



Home automation with Arduino and Raspberry Pi
I'm again working on a project for school. But this time I'm not uploading it when it's finished, we're still working on it and for some parts we could still use some help.



Arduino Experimenting, Evaluation and Expansion Shield
This shield can be used with an Arduino Yun board, but there are connectors on the Yun that require the shield to be mounted a bit higher. The most simple solution is to buy an extra set of Arduino Stackable headers (e.g. Sparkfun order code PRT-11417) ...



3D Printer Head and Mat Temperature controller - using Arduino

Affordable and easy to build - Using standard Arduino, piggy backed onto PCB negates the need for SMD skills and tools. Easy to use - turn a knob and temperature is set. Visual feedback on parameters.



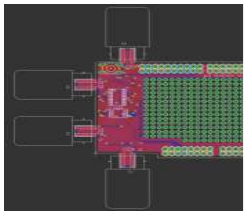
3D printed Arduino housing

Yes, my Arduino is always in danger to catch fire because of so many cables and wires on the bench. What I need is a proper housing to make it safe. So here it is, my new Arduino Safeguard. The Arduino Uno is in use for my beginner's course.



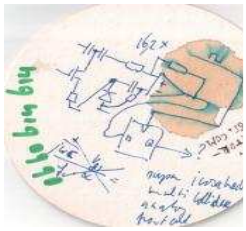
IV-3 VFD shield for Arduino

This Arduino shield is capable of driving 4 Russian IV-3 7 segment VFD tubes. 4 3mm LEDs provide background lighting for the tubes. The design is completely based on through hole components, no SMD components were used.



Arduino Nunchuk Shield

An Arduino shield to connect 4 Nintendo (TM) WII Nunchuks . In V0.8 there was a Bug. The supply for the Nunchuk must not be connected to 5V!



12V DC Model Train Arduino controller simulating momentum
I found some research papers about real world train controller maths, but the maths would be too much for the Arduino I guess ;-) I have not yet hooked up an Oscilloscope but I have seen other clever clogs complaining the the H-Bridge LM298 output might not be the best for a 12V DC motor.



IoT Learning Platforms

Arduino Uno + cellphone shield code for triggering vibration motor. Go here. *Note that the cellphone code works with an Arduino 0022 dev environment but has not been ported to 1.0 as of yet! Arduino Uno + Sparkfun geophone code. Go here.



Usb modules - Arduino like board & soft
My projet : - Boards with arduino like connector for FTDI modules (FT2232, FT232 ....) - Netbeans templates projets for the programming of those modules in C++.



Hack n Mod http://hacknmod.com/

Arduino Hacks http://hacknmod.com/tag/arduino/
http://hacknmod.com/hack/top-40-arduino-projects-of-the-web/

IDEAS and PROJECTS



How to: Super Simple Self-Balancing Robot Tutorial

Since the introduction of the segway, DIYers have been building their own self-balancing transportation devices. Before you go off and build a full-size version, here's a simple project to help you learn the basics of control system design (the...



Epic Doomsday Device Build

Inspired by science fiction movies, this DIYer created a doomsday device prop complete with pull-pin, epic LED lighting, and countdown timer. When his microwave broke, he salvaged its innards to build this device. Overall, this project is a...



iPad App Blends Virtual Warfare & Real-World Characters

This project is a basic proof of concept of creating a "physical" video game using real-world characters. For this project, a tank and robot duke it out with the help of an Arduino, iPad, and wifi. Essentially, when the tank's targeting system...



Epic DIY Replica of Disneyland's Electrical Light Parade

Every night at Disneyland, an Electrical Light Parade illuminates the darkness by featuring vehicles draped in a fantastic array of multicolored lights. One enthusiastic fan (who had already...



DIY Doorbell Video Security System

Ever wonder who comes to your door during the day while you're gone? Here's a simple DIY project to answer that question. Every time someone rings your doorbell, you'll receive a text message containing a photo of the person. The system...



Incredible RGB LED Cube Tutorial

We've seen our fair share of LED light cubes before, but what makes this one different is its incredibly detailed, step-by-step tutorial. The creator of this light show previously made a one color LED light cube and shared the trials,...



Interactive Magic Mirror Controlled via Android

We've seen some neat interactive mirror projects in the past, and here's one which integrates bluetooth and multiple external devices. It's control-led by IOIO (pronounced yo-yo), an Arduino-type microprocessor. It runs several different apps...



Tour of the Arduino Manufacturing Facility

Have you ever wondered where all those magical Arduinos come from? Surprise! They come from Torino, Italy. The video below will walk you through the PCB manufacuring and assembly process. A quick history lesson for you: this revolutionary...



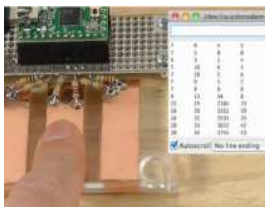
Start your Car from Afar using Arduino

This DIYer had a 1st generation iPhone lying around, so he decided to put it to good use. You can use any phone to accomplish this, even a $10 prepaid phone. An SMS to the iPhone goes through an Arduino ...



DIY Dazzling Laser Light Party Ball

Here's a challenge for you: stuff over a dozen red lasers and a rechargeable lithium ion battery inside a tennis ball. Oh, and also a microcontroller and infrared port to control patterns from afar. To make...



DIY Capacitive Touch Made Easy

Capacitive touch allows electronics to sense when your finger is within a few millimeters of a surface to simulate a button "press" without the use of an actual mechanical button. You'll notice the feature on the new Xbox 360 Slim. Here's a...



Psychedelic Sphere of LEDs

The friend of a performer created this optical light display of 256 LEDs that can be programmed to create some fascinating displays. The ball has 16 panels, which are not only the circuit boards, but the structural support of the ball as well....



How to: Build a Giant POV Light Tube

With this POV light painter, you can paint the town red or any other color that suits your fancy. The LightScythe is a 2-meter long POV tube made up of multi-color LED strips. POV stands for describes...



Electric Mountain Board with Wireless Control

As though an electric mountain board isn't cool enough for zipping around, this industrious inventor just added wireless control. We previously featured Andres Guzman's mountain board which, at the time, was controlled via a PlayStation...



Impressive Tree Climbing Robot

This DIY build uses pointed feet and moves up the tree something like an inchworm. Two, separated platforms are connected toa retractable spine that alternately grip the tree to move the device upward. The device has an Arduino Uno for a brain...



DIY Sonar for the Blind

This is one of those simple concepts which makes you think "Why didn't I think of that?" The entire sonar unit is attached to a wearable glove and as the user sweeps their arm back and forth, the device can detect anything from one inch to 10 feet...



Fully Autonomous Arduino Airsoft Tank

This fully autonomous airsoft tank uses an Arduino along with a Lego NXT module to control motor and detect targets. It navigates its surroundings using an ultrasonic sensor to avoid obstacles. Fitted...



Arduino Device Mutes Celebrities on TV

Learn how to program the Arduino to turn it into an intelligent "celebrity screener" using some keywords and Arduino. The appropriately named "Enough Already" mutes the TV by decoding the closed captioning transcription track, which is ...



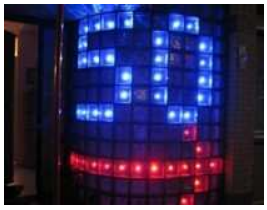
Use Google Voice Search through Arduino & Bluetooth

With the launch of Google's new Voice Search feature on the Chrome browser, you can trust the DIY community to come up with some cool hacks. The Verbalizer is an Open Source development board based on Arduino which wirelessly communicates with...



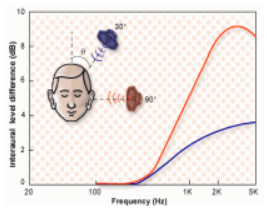
DIY Bluetooth Camera Shutter Release

If you're into photography, you may be wishing for a way to activate your camera shutter from a distance. Making use of an Arduino Uno bluetooth transceiver and an Android smartphone, this DIYerallace created a fairly simple way to send signals...



DIY LED Glass Wall

BlinkM is a smart LED that displays various shades of light effects by fading or changing color. You can buy them for about $15, but this DIYer assembled his own for about $4 to build an LED wall. The result is this 11x11 matrix of light to an I2C...



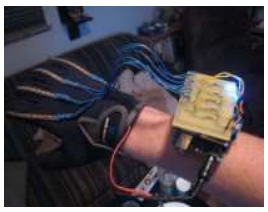
DIY Sound Localization Sensor with Arduino

We have covered loads of motion detection projects, but what about sound? A sound localization sensor helps you to identify the location or origin of a detected sound and find out its direction and distance. Our ears do it for us, but building an...



Incredible Light Painting with an Arduino

Long exposure shots can create some pretty interesting effects. Light painting takes this idea and morphs it into innovative examples of 'illuminated' art. If you experiment with different light sources and treat them as 'brushes', the...



Arduino Wireless Animatronic Hand Project

This DIY Robotic hand mimics the motions of a human counterpart. The controls lie in a special glove which has flex sensors wired to an Arduino which is connected to an XBee radio which wirelessly interfaces with a computer and controls the...



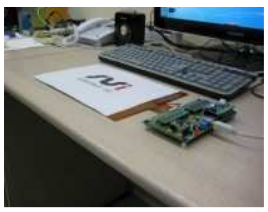
Arduino Powered, Chess Playing Robot

Arduino DIY-ers have built another superb project. This chess board looks like any chess board except the chess pieces are metallic and moved by a magnet below the board controlled by a robotic arm. The microcontroller controls stepper motors and...



DIY LED Matrix Pong Clock

Don't Miss: Dirt Cheap, Epic DIY Pong Clock This neat LED clock mimics the display of the Pong video game to show the time as two paddles play back and forth. The game plays on automatically and updates the time by scoring a point. It also...



DIY Open Source 'Paper Thin' Multitouch Trackpad

OpenMT project is an Open Source initiative that has the noble aim of helping out DIYers who want to create their own thin and flexible pressure sensitive multi-touch panel. Unlike lot of camera based multi-touch projects we have seen, The Open...



Customize your Car's Electronics using Arduino

Vehicles these days come with a lot of electronics but you can learn from our collection of car hacks to add your own custom features. For example, take the Jeep Wrangler and put in a couple of Arduinos, an LCD display, shift s,...



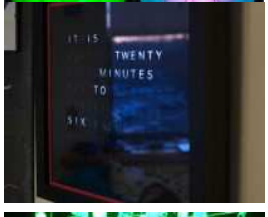
Arduino-Based Snail Mail Alert for Your Smartphone

Since we're so focused on our email and text nowadays, we often neglect to check our snail mail. Now, with the help of an Arduino, you can easily setup notifications and alerts for just about anything. This project uses an Arduino, a...



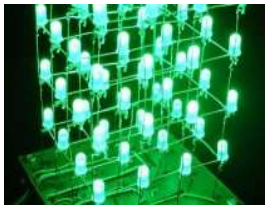
DIY New Year's Eve LED Ball

With the New Year rapidly approaching, learn how to build this LED New Year's ball to add some unique and superb decor to your party. It uses six high power RGB LED arrays, six LED controllers, and an Arduino. The ball itself is bade from about...



Build a Glossy Black Word Clock with Arduino

For a unique addition to your living space, the word clock is a novel new way to tell the time. The inspiration for this black and sleek clock comes the previously mentioned DIY Arduino Word Clock. Yes, it's more difficult...



Build an LED Cube Using an Arduino

Learn how to make a stunning LED cube using an Arduino. Although the image above looks a bit plain, you can control the colors and patterns of the 64 LEDs. Also, if you're looking for a unique soldering challenge, this is a great projec to start...



DIY Dorm Room Keypad Lock with Arduino

Who doesn't want a unique way to open their dorm room? We've seen loads of different ways, including RFID locks and even laser surveillance security systems. Don't Miss: Our collection of DIY Security projects We've seen a similar example of...



Back to the Future 'Auto-Lace' Sneakers

If the Nike Wiis aren't an extreme enough shoe mod for you, check out these self-tying Power Laces. Inspired by the Back to the Future films, the Power Laces Prototype features a motor and an Arduino microcontroller. When your foot hits the...



DIY Fiber Optic Fabric using RGB LEDs

By weaving fiber optic strands as if they were yarn, this team of DIYers created a fiber optic "fabric" of sorts which features a few dozen RGB LEDs and reacts to the outside world via Twitter. Currently, they're raising...



Secret Knock Detecting Door + Mechcanical Iris = Awesome!

Here's a superb DIY project which comes two outstanding hacks we've covered previously. It's a steampunk mechanical iris combined with the secret knock detecting door to create this unique garage door opener. The brassy mechanical iris hides...



Laser Tripwire takes a Photo, Uploads it to Twitter

The laser tripwire project is superb project for both experienced and brand new Arduino enthusiasts. It combines both hardware, software, and communicating between the two. The simple project captures a snapshot of an unauthorized entry and...



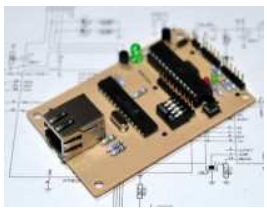
Incredible DIY Daft Punk Helmet Project

We've covered some downright superb daft punk projects in the past, but this helmet is one of the best so far. It took 17 long months to build and was documented fairly well in the project's build notes. The video below is a great walkthrough of...



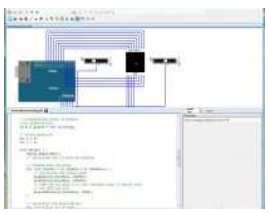
Giant LED Matrix Controlled via the Web

What do you get when you combine five LED enthusiats, a free weekend, an Arduino and hundreds of LEDs? The LED wall, which is a massive pegboard (size - 53" x 31") fitted with a matrix of 350 LEDs. The matrix of LEDs can be controlled by...



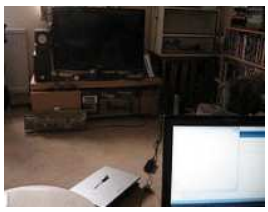
Add Internet to your Arduino (for half the price!)

Imagine being able to control your Arduino projects from anywhere... Yes, this is possible by simply adding an ethernet shield to your Arduino, but you'd have to sacrifice the programming board. Instead of sacrificing the entire Arduino, just...



100% Software-Based Arduino Simulator!

Some software developers don't get into hardware because they enjoy the instant gratification of seeing their work displayed on screen. They don't want to fuss with soldering or hooking up components. Now, however, they can't complain since...



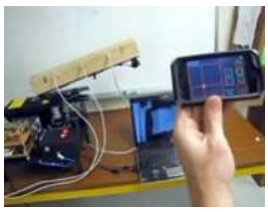
Control Electronics using an Arduino and Infrared LEDs

Learn how to use infrared LEDs to send signals to your TV and other electronic devices via an Arduino. First, the Arduino will interpret IR pulses sent out by the TV remote, save them to memory, then "replay" them upon the user's command. ...



Superb DIY Retro Lighting Design

Sometimes, your inventions don't have to have a practical function. Take for example, this epic looking decorative lamp. Dubbed 'Coachella', this eye-catching device was named for an annual music and arts festival. Although it has a steampunk...



Wi-Fi Enabled Coil Gun with iPhone App

We've covered loads of airsoft, nerf, and gun projects, and here's another superb pr0ject to add to our collection. It's a coil gun placed on a turret which is triggered via Wi-Fi. Additionally, it uses remote targeting and shooting through...



Control a Robot Arm over the Web using Arduino

We've seen some neat internet-based remote control projects including the Ethernet Enabled Christmas Bells project. Here's another neat project to add to our collection. The Orbduino is a robotic arm which can can be controlled entirely...



DIY Arduino Helicopter with Obstacle Avoidance

Here's another neat Arduino helicopter project featuring four infrared sensors and one ultrasonic sensor for obstacle avoidance. We've seen an Arduino powered helicopter with four rotors and even one with six rotors too. This quadcopter,...



Twitter Based Mood Light using Arduino

By monitoring a stream of tweets coming through Twitter, this mood light changes color based upon the frequency of words such as "angry, happy, or sad." The system uses an Arduino which collects data from Twitter using a WiFly module. After...



How To: DIY Arduino Balancing Surface

Here's another superb Arduino project using just a few simple components (which are available in our DIY projects store). It's a pretty simple setup using a laptop, webcam, servos, and an Arduino to monitor and balance a ball bearing on a flat...



Build a Self-Balancing Segway Skateboard

This low cost self-balancing skateboard project is a mix between the Segway-like scooter and Electric Segway Skateboard we've covered before. Add an Arduino microcontroller, a motor, and some sensors to a standard skateboard and you get a...



Arduino Powered Remote Control Lawnmower

We've seen loads of great Arduino projects and even a few RC lawn mowers, but we've never seen the two combined until now. This project walks you through the entire build process step by step and includes a thorough guide for creating remote...



Arduino Based LED Alarm Clock

Making an accurate microcontroller clock can be a little tricky but you can learn how it's down by building your own Arduino alarm clock using LED Matrices and a coffee can housing. All of the electronics are based on Wisedino (Arduino with timing...



Twitter Controlled LED IKEA Table using Bluetooth

An ever-changing LED display makes for an interesting show. An LED table powered by tweets makes the light show that much more exciting. This Twitter-enabled device builds on a previous project which had uses 81 RGB LED modules in a 9x9 grid...



War Torn District 9 PC Case Mod

District 9 was a popular film which has spawned numerous DIY project such as the District 9 rifle we featured previously. The latest we have come across is the superb PC case mod above which started with a stock ATX case which was fabricated with...



How to: LED Hard Drive Clock using Arduino

Ah...the Arduino. The super powerful, easy to program microcontroller brings us another fantastic DIY project. We've seen a few hard drive LED clocks in the past, but this one looks outstanding and is easier to create. It uses POV...



Interactive, Portable "Magic Mirror" Arduino Project

Using an old suitcase, a laptop, X10 home automation parts and an Arduino set-up, this guy managed to build a fully-functional magic mirror suitcase. Basically, it's an interactive mirror which reacts to it's environment using various...



Superb Reverse-Geocaching Gift Idea

Geo-Caching works by finding a "treasure" at a certain longitude and latitude. This box on the other hand opens only when it is brought to a certain location so it's perfect to give to someone as a surprise gift or romantic occasion. We've...



Arduino Powered Helicopter with Four Rotors

We've seen loads of superb Arduino projects, but few which involve complex aeronautics. Here we have the "quadrocopter" - an arduino powered, remote control helicopter with four rotor blades. It's a superb build overall with the finished...



Exercise Bike Xbox 360 Controller

Need help with your New Year's resolution? Burn off the calories while playing your favorite racing games. The faster you pedal, the faster your vehicle travels. Learn how to build your own Excite Bike video game controller. The...



Touchpad Built With an Arduino and Infrared

We've seen a few multitouch projects including a 5 minute dirt cheap pad and the more complex flexible multitouch display. This multi-touch device (the TRKBRD) is basically is a finger tracking sensor. By the looks of it, it does...



HacknMod's Most Popular DIY Projects of 2009

Another year behind us! HacknMod will turn 4 this June and we're still growing strong. We've nearly trippled our traffic in the last year and we're glad to have you along for the ride. Oh! You can show your HacknMod...



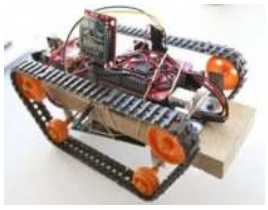
Ethernet Enabled Christmas Bells - Create a Tune in Real Time!

Here's a superb project which allows you to create a Christmas jingle using a flash interface in your browser. Then, you're redirected to a webcam where you can see your tune played in real time! Awesome! The project uses the Arduino...



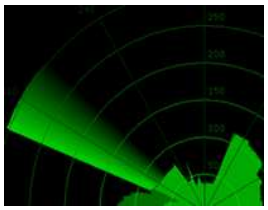
DIY Batman Signal Spotlight using Arduino

We've covered tons of batman projects on Hack N Mod previously and here's another superb build. The final result looks fantastic, allowing you to project any image you'd like using an Arduino and RGB LEDs. Full project build...



iPhone Controlled Tank using Arduino

Here's a neat project featuring an iPhone controlled tank which has an Arduino for a brain. The data is transferred wirelessly to Processing and then to the Xbee wireless serial communication module. Oh, and did we mention it's solar...



Make a DIY Sonar System using Arduino

Learn how to build a robot with built in visual sonar using sensors, an arduino, and Processing. The author supplies the code for Processing, an easy to learn programming tool for visualization. Processing gathers the sonar data from the...



Make a DIY Arduino Word Clock

Find out how to build your own 'Word Clock' using the power of Arduino. It's definitely unique and eye-catching clock which literally tells you the time unlike your standard digital clock. The instructions are really easy to understand and well...



Secret Knock Detecting Door Lock

Here's a fantastic project using an arduino, a motor, and a piezo sensor which allows you to both record and detect knocking patterns to unlock a door. The video below does a fantastic job of demonstrating how it works: It's completely mobile...



GhostBusters DIY Halloween Costume (Arduino Powered)

For your DIY Halloween costume this year, Learn how to make your own Ghostbusters Proton Pack equipt with red and green lasers (above). There is a full tutorial on how to build your own along with a detailed photogallery of the proton...



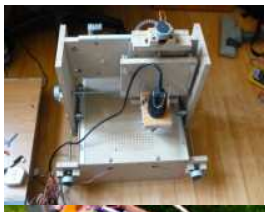
DIY Football Hero Using the Arduino

We've seen plenty of guitar hero projects and even a laser harp hero, but now we have a football hero (Soccer for us Americans). Made for the music industry, the director pulled together a team of DIYers and football players to make the game a...



Top 5 Custom LED Beer Pong Tables

Beer Pong...The sport of choice of college students. While we would never, ever support over consumption of alcohol, we must admire the engineering and creativity of these outstanding illuminated tables. If you're new to LEDs, check out...



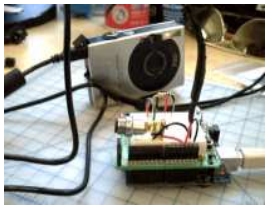
Incredible $100 Arduino-Based CNC Machine

CNC machines allow you to mill just about any shape or design you'd like using X, Y and Z axis. Typcailly they cost serveral thousand dollars, but this man built his own CNC machine and wrote up a tutorial on how you can ...



The Webcycle for the Internet Junkie

Is your desire/addiction the web strong enough to pull you away from the information superhighway onto an exercise bike in order to keep you plugged in? The webcycle is a bicycle which controls your internet speed ...



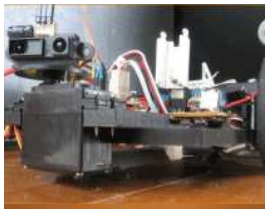
Remote Controlled Camera Trigger Arduino Project

Controlling the shutter of a camera remotely has several uses. The most important catching a photo at exactly the right time. Many top of the line digital cameras come with remotes. But what if you wanted to build one for your average...



Make a DIY Open Source PSP with an Arduino

Here's another fantastic tutorial from our Open Source Arduino master, Matt. See all of our Arduino projects. In the spirit of the open source gameboy (#5 on the Top 40 Arduino Projects of the Web), Matt's latest project uses a few Arduino...



Build a Sensory Robot with Arduino and Processing

An infrared proximity sensor on a robot can give it eyesight to make sense of the real world. Cory Barton has used an inexpensive Sharp distance sensor, an Arduino and the open source Processing language to build his robotics...



RFID Enabled Mp3 Player Hack using Arduino

This is a fantastic, unique project which uses RFID tagged pictures to select and play album. The system utilizes the easy to use arduino microcontroller platform along with an RFID reader and is also ethernet enabled. See more ...



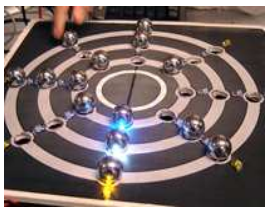
Incredible, Colorful RGB LED Light Strip

This incredible, RGB LED light strip allows you to address each and every LED individually, creating splendid color patters. If you were challenged by the Open Source POV Bicycle Wheel Animation project we covered last week or were thinking of...



DIY Internet Enabled Arduino Messenger

Is there anything that the Arduino microcontroller can't do? Just look at the vast assortment of applications in Top 40 Arduino Projects of the Web we've covered earlier. Here's another awesome arduino project by Hans Scharler. Hans uses an...



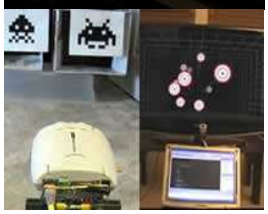
Cornell's Newest Microcontroller Projects

Cornell has released their latest batch of microcontroller electronics projects ranging from a 3D scanner to infrared laser harps. All of the projects use Atmel ATmega chips (the same chips Arduino uses) to accomplish their tasks. All 42...



DIY Midi Drums using an Arduino

Using some piezo sensors, this DIYer created some DIY midi drums for the Arduino. All it takes is a handful of the sensors and a few resistors to put together. He's already assembled the soruce code, parts kit and detailed instructions so you...



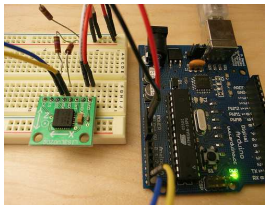
Top 30 Wiimote Hacks of the Web

Nintendo's Wii remote (or wiimote) has proven to be an incredibly hackable device. This is mainly because of its bluetooth connectivity and assortment of sensors which make it a great interface for controlling DIY electronics projects. So, for...



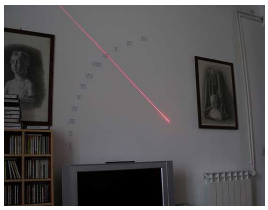
New Arduino Mega & Interview with Tom Igoe

As you should know by now, Arduino is a fantastic microcontroller platform featuring a simple data input and outputs along with an easy to learn programming environment. , Now the boys over at tinker.it have come out with the Arduino's big...



How to Connect Multiple Arduino Microcontrollers with I2C

Use I2c to connect several Arduinos together to combine their power. , I2C stands for the Inter Integrated Circuits and , is one of the best Multi Master Serial Computer Bus technologies used in different systems to connect the...



Make your Own 21st Century Sundial

If you thought sundials were a thing of the past, think again. The only difference is that this time around things are much more high tech. Over at 5volt there is a fantastic tutorial demonstrating how to build your own laser sundial. The great...



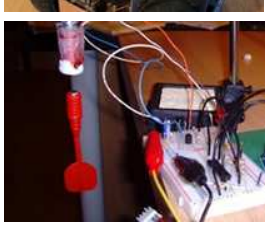
Arduino & NunChuck create Real Time DIY Pan & Tilt Camera

Learn to create your own DIY pan and tilt camera which is controllable (in basically real time) by using a Wii nunchuck and the Arduino. By moving and rotating the nunchuck, two servo motors match your movements allowing you to control your...



Twitter Robot makes Popcorn, starts a Movie

Imagine this... On your way home from work, you send the Twitter command. Your robot reads the command, waits for a bit, pops some corn and starts a movie. , Build your own twitter powered...



Magnetic Levitation using the Arduino

This great project uses a soleniod's magnetic coil, a Hall sensor, and Arduino to perfectly levitate a magnet. The Hall sensor detects the field of the permanent magnet and uses that information to modulate the magnetic field of the...



Hack Twitter to Control Electronics

Twitter has become one of the most popular social media sites, but its uses extend far beyond the world of the internet. By, interfacing physical hardware with Twitter these projects have created fantastic hacks allowing electronics to react to...



Automatic Head Tracking with Arduino

We've seen some infrared head tracking before and now it has been accomplished using the Arduino. The demo video above shows its interactivity with a flight simulator. Some modification to the code...



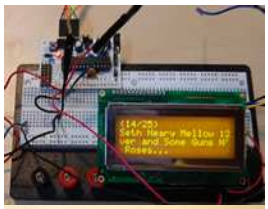
Use the Wii and iPhone to Control an RC car

This fantastic video demonstrates the use of the wiimote, wii balance board, and iPhone to control an RC car. For data transmission, an Arduino is used to communicate between devices. Learn how to do this yourself in a ...



Make a UAV Spyplane Using the Arduino

With a handful of servos, battery pack, GPS unit, and the Arduino programmer you can build your own unmanned aerial vehicle (UAV) drone. It's an absolutely outstanding project complete with a detailed, 22 page tutorial to guide you through the...



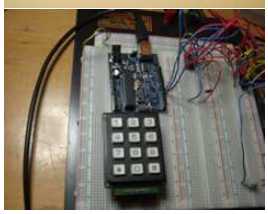
Gmail and RSS Notifiers using the Arduino

By programming the Arduino and having it communicate to the web through your computer's serial port, you can create your own physical notifier which alerts you when you receive email or when an RSS news...



Electronics Filled, Arduino Powered Room

Using sensors, an old PC, LCD screen, lots of sensors and few X10 home automation parts, one man built a mini-house in his backyard packed full of electronics. The house also features a flickering LED fireplace and a...



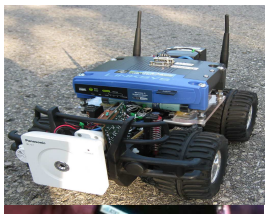
Arduino Keypad Security System

Using the endless powers of the Arduino, one man built a keypad entry security system using the Arduino and some servo motors. The project also uses a speaker to verify the door is locked. , Luckily, the project has detailed build...



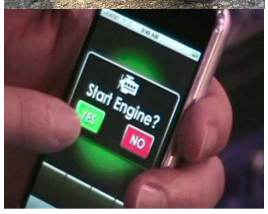
Make a Mini Segway Using the Arduino

Using the Arduino, and a couple LEGO motors and sensors, build your own self balancing segway-like robot. We recently saw a similar balancing robot using a far more simple design. There aren't too many details posted about the...



RC Car Controlled Via the Web

Strap on a standard Linksys router to an RC car and you can wirelessly control it through the web from up to 1640 feet away. It's also equipt with a webcam and operates through a brain comprised of PIC microcontrollers or the Arduino's...



DIY iPhone Controlled RC Car

Combine the power of the Arduino with the iPhone and you're bound to end up with an outstanding project. Learn how to control an RC car using the multitouch and tilt sensing features of the iPhone. The data is basically sent from the iPhone...



DIY Magnetic Strip Card Decoder

Using a magnetic strip reader incorporated with, of course, an Arduino, you can scan what those strips and see the data stored on it. , A bit more hacking and you could also write data back onto the card and create duplicates as well. If...



Website Visitor Blinker

I already wrote my first Arduino program. It's really simple but fun :) What it does is that every time somebody visits my website, a LED light blinks once in my room, and I get a warm feeling of a visitor ;) You can test it by opening a new...



Create Techno with a Laser Harp

Ever wanted to make your own techno music, but are tired of the standard midi controller? Well, you could build a harp (of all instruments) out of lasers and then play it by breaking the beams of light. See how it's done. The harp is connected...



Make an LED Gmail Notifier

Using an Arduino, you can program any device to turn on, illuminate, spin, fly, whatever you want when you receive an email. In this tutorial, the author used an LED Cube with an embedded RBG color cycle. If you don't happen to have a glowing...



Рюмик С. М.

Микроконтроллеры DUINO. Сборка 1...7

— ж. "Радиоаматор", 2010, № 2…9, 11. — Цикл статей.

Считается, что каждые 5-7 лет радиолюбитель должен проходить заочные курсы повышения квалификации. Зачем? Чтобы не отставать от жизни и следить за новостями. Чтобы держать себя в тонусе и быть компетентным. Чтобы разбираться в свежих технологиях и не пасовать перед знатоками. Наконец, чтобы не ослабевал интерес к любимому хобби.

Очередной цикл статей будет посвящен одному из новомодных течений в микроконтроллерной технике, получившему большую популярность благодаря проекту «Arduino».

Листинги, HEX-коды и другие дополнительные материалы размещены на сайте журнала "Радиоаматор" http://www.ra-publish.com.ua/

СОДЕРЖАНИЕ

Сборка 1
Постановка задачи
Философия платформы «Open»
Проект «Arduino»
Особенности «Arduino»
Структурные схемы
Электрические схемы
Если внимательно приглядеться...

Сборка 2
Функциональная схема базового модуля «Arduino»
Состав сигналов «Arduino»
Конструкция «Arduino»
Разводка печатной платы
KiCad: порядок действий

Сборка 3
История короля Ардуина
Arduino-совместимые проекты
Среда программирования «Arduino-0018»
«Arduino» и WinAVR

Сборка 4
«Беспаечные» платы
Программа «Fritzing»
Логика работы управляемого светофора
Алгоритмический язык Wiring
Структурная схема типовой Wiring-программы

Сборка 5
Тонкости тестирования и поиска неисправностей
Полный отказ «Arduino»
Частичный отказ «Arduino»
Тестовый стенд для проверки «Arduino»
Программное обеспечение стенда
Среда визуального программирования HiAsm

Сборка 6
Философия разработок на основе «Arduino»
«Гаджеты»
Дверные звонки
Внешний вид дверных кнопок
Электрические схемы
Пояснения к листингам
Важный нюанс

Сборка 7
Измеритель времени «дребезга» контактов
Алгоритм функционирования программы «Knopka.exe»
Программа для прошивки МК ATmega168
Калибровка измерителя времени
Подведение итогов



Рюмик С. М.

Конструкции с Arduino. Часть 1...5

— ж. "Радиоаматор", 2011, № 3, 6, 8, 10, 12. — Цикл статей.

«В одну реку нельзя войти дважды» - так древнегреческий философ Гераклит образно охарактеризовал простую житейскую мудрость. В нашем случае первый подход к тематике «Arduino» состоялся в 2010 г. Второй подход (нынешний) будет отличаться от первого большей практичностью. Иными словами, теоретический курс пройден, наступило время для разработки и показа конкретных конструкций.

Платформа Arduino находится в постоянном развитии. Появляются новые разновидности плат, совершенствуется программное обеспечение, генерируются свежие идеи относительно применения Arduino в повседневной практике.

Но для начала следует вспомнить о «средствах производства». На сегодняшний день официальная линейка «Arduino» состоит из 12 фирменных модулей: Uno, Uno SMD, Nano, Mini, LilyPad, Fio, Mega-2560, BX Pro, Pro Mini, Serial, Serial Single-Sided. Любой из них можно использовать для экспериментов, хотя желательно, чтобы микроконтроллер (МК) был посвежее, в частности, ATmega328P.

Собрать конструкцию с применением микроконтроллерного модуля Arduino - это полдела. Не менее важно разобраться в методике получения результата, исследовать разные сферы применения, научиться не останавливаться на одном варианте решения проблемы.

Листинги, HEX-коды и другие дополнительные материалы размещены на сайте журнала "Радиоаматор" http://www.ra-publish.com.ua/

СОДЕРЖАНИЕ

Часть 1
Минимум миниморум
Минимальный «Arduino»
Оптоизолированный «Arduino»

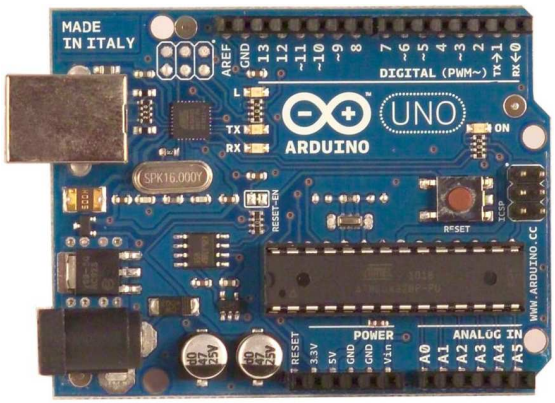
Часть 2
Arduino UNO
Разновидности UNO
Аналоги UNO
Порядок установки программного обеспечения
Тестирование UNO
Arduino и терменвокс

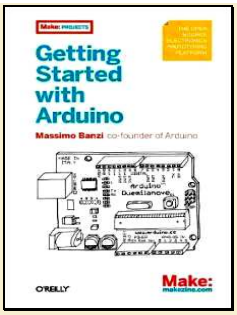
Часть 3
Сигнализатор землетрясений
Проект «The Seismic Reflector»
Модельные эксперименты с программой «Quake»
Управляющая программа на базе HiAsm
GSM-сигнализатор землетрясений

Часть 4
Что такое Japanino?
Arduino и резистивные датчики
Знакопеременное напряжение

Часть 5
Ассемблер, Си или Wiring?
Среда программирования Scratch для Arduino
Порядок действий
Электромузыкальные инструменты на основе Arduino
Список использованных источников и литературы







Massimo Banzi. Getting Started with Arduino

— O'Reilly Media, Inc., 2008. — 128 с.: ил.
ISBN 978-0-596-15551-3

Массимо Банци. Знакомство с Arduino

Пер. с англ.
Интернет-ресурс: http://arduino.shopium.ua/pages/arduino-getting-started/

От переводчика:

- Если вы доселе не держали в руках Arduino, вам стоит это прочесть!
- Текст взят из файла *Getting Started with Arduino.chm*, гуляющего по интернету.
- В книге описана работа с платой Arduino Duemilanove, но вы вожете применять примеры с любой из плат Arduino, просто внимательно читайте описание выводов вашей платы.
- Перевод не литературный, не было стилистической правки, возможно шероховатости. Если вы хотите улучшить текст, присылайте исправления и замечания, буду рад!

От автора:

Эта книга была создана для того чтобы начинающие поняли преимущества, которые они могут получить от изучения возможностей применения, а также для того, чтобы они поняли её философию.

Эта книга была написана для "настоящих" пользователей Arduino: дизайнеров и художников. Поэтому она пытается объяснять вещи таким образом, который может свести некоторых инженеров с ума. Вообще-то, один из них назвал вступительные главы моего первого проекта отстоем. В этом-то и дело. Посмотрим правде в глаза: большинство инженеров не в состоянии объяснить то, что они делают другим инженерам, не говоря уже об обычных людях. После того, как Arduino начала становиться популярной, я понял что экспериментаторы, люди с хобби и хакеры всех видов начали использовать её для создания прекрасных и безумных вещей. Я понял, что вы все художники и создатели в собственном праве, так что эта книга для вас.

СОДЕРЖАНИЕ

Глава1. Введение
1.1 Целевая аудитория
1.1.1 Дизайн взаимодействия - это дизайн любого опыта взаимодействия
1.2 Что такое физические вычисления?

Глава 2. Путь Arduino
2.1 Прототипирование
2.2 Самоделкины
2.3 Смешивание
2.4 Искажение схем
2.5 Хаки клавиатуры
2.6 Мы любим мусор!
2.7 Хакайте игрушки
2.8 Сотрудничество

Глава 3. Платформа Arduino
3.1. Аппаратное обеспечение Arduino
3.1.1 14 контактов цифрового ввода-вывода (контакты 0-13)
3.1.2 6 контактов аналогового входа (контакты 0-5)
3.1.3 Контакты аналогового выхода (контакты 3, 5, 6, 9, 10 и 11)
3.2 Интегрированная среда разработки (IDE)
3.3 Установка Arduino на ваш компьютер
3.4 Установка драйверов: Macintosh
3.5 Установка драйверов: Windows
3.6 Идентификация порта: Macintosh
3.7 Идентификация порта: Windows

Глава 4. Знакомство с Arduino - теперь точно!
4.1 Анатомия интерактивного устройства
4.2 Сенсоры и актюаторы
4.3 Мигание светодиодом
4.4 Передайте мне пармезан
4.5 Arduino не остановить
4.6 Настоящие самоделкины пишут комментарии
4.7 Код, шаг за шагом
4.8 Что мы будем создавать
4.9 Что такое электричество?
4.10 Использование кнопки для управления светодиодом
4.11 Как это работает?
4.12 Одна схема, тысяча применений

5. Продвинутый ввод-вывод
5.1 Пробуем другие датчики включения-выключения
5.1.1 Выключатели
5.1.2 Термостаты
5.1.3 Магнитные переключатели, также известные как "герконы"
5.1.4 Ковровые переключатели
5.1.5 Датчики наклона
5.2 Управление светом при помощи ШИМ
5.3 Использование фотодатчика вместо кнопки
5.4 Аналоговый ввод
5.5 Попробуйте другие аналоговые датчики
5.6 Последовательная связь
5.7 Управление большими нагрузками (электродвигатели, лампы и тому подобное)
5.8 Сложные сенсоры

Глава 6. Разговоры с облаками
6.1 Цифровой вывод
6.1.1 Цифровой вывод
6.1.2 Аналоговый вывод
6.1.3 Цифровой ввод
6.1.4 Аналоговый ввод
6.1.5 Последовательная связь
6.2 Планирование
6.3 Программирование
6.4 Сборка схемы
6.5 Как собрать лампу

Глава 7. Устранение неполадок
7.1 Понимание
7.1.1 Понимание
7.1.2 Упрощение и разделение
7.1.3 Исключение и уверенность
7.2 Проверка платы
7.3 Проверка схемы на макетной плате
7.4 Выделение проблемы
7.5 Проблемы с IDE
7.6 Как получить помощь онлайн

Приложение A. Макетная плата
Приложение B. Маркировка резисторов и конденсаторов



Brian W. Evans

Arduino Programming Notebook
Блокнот программиста Arduino

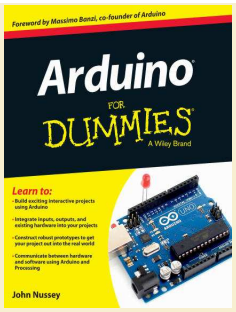
— перевод с англ. В.Н. Гололобова — 40 с.

Интернет-ресурс: http://robocraft.ru/files/books/arduino_notebook_rus_v1-1.pdf

Этот блокнот следует рассматривать, как удобное, лёгкое в использовании руководство по структуре команд и синтаксису языка программирования контроллера Arduino. Для сохранения простоты, были сделаны некоторые исключения, что улучшает руководство при использовании начинающими в качестве дополнительного источника информации - наряду с другими web-сайтами, книгами, семинарами и классами. Подобное решение, призвано акцентировать внимание на использовании Arduino для автономных задач и, например, исключает более сложное использование массивов или использование последовательного соединения.

CONTENTS

Предисловие
структура
setup()
loop()
функции
{} фигурные скобки
; точка с запятой
* ... */ блок комментария
однострочный комментарий
переменные
объявление переменных
границы переменных
byte
int
long
float
массивы
арифметика
смешанное присваивание
операторы сравнения
логические операторы
константы
true/false
high/low
input/output
управление программой
if
if...else
for
while
do...while
цифровой ввод/вывод
pinMode (pin, mode)
digitalRead (pin)
digitalWrite (pin, value)
analogRead (pin)
analogWtite (pin, value)
время и математика
delay (ms)
millis()
min (x, y)
max (x, y)
случайные числа
randomSeed (seed)
random (max)
random (min, max)
последовательный обмен
Serial.begin (rate)
Serial.println (data)
приложение
цифровой выход
цифровой ввод
сильноточный выход
pwm выход
вход с потенциометра
вход от переменного резистора
серво вывод



John Nussey

Arduino For Dummies

Arduino для «чайников»

— Publisher: John Wiley & Sons, Ltd., 2013. — 446 pages
— Series: For Dummies

ISBN-10: 1118446372
ISBN-13: 978-1118446379

Быстрый и простой способ, чтобы прыгнуть в захватывающий мир физических явлений. Начните создание очередной крутой штуковины с Arduino и этой книги.

CONTENTS

Foreword
Introduction
About This Book
Foolish Assumptions
How This Book Is Organized
Icons Used In This Book
Where to Go from Here

Part I: Getting to Know Arduino

Chapter 1: What Is Arduino and Where Did It Come From?
Where Did Arduino Come From?
Learning by Doing
Patching
Hacking
Circuit bending
Electronics
Inputs
Outputs
Open Source

Chapter 2: Finding Your Board and Your Way Around It
Getting to Know the Arduino Uno R3
The Brains: ATmega328 microcontroller chip
Header sockets
Digital pins
Analog in pins
What about analog out?
Power pins
USB socket
External power jack
Reset button
Discovering Other Arduino Boards
Official Arduino boards
Contributed (Approved) Arduinos
Shopping for Arduino
Official Arduino Store
Distributors in the United Kingdom
Distributors in the United States
Amazon
Electronics distributors
Kitted Out: Starting with a Beginner's Kit
Preparing a Workspace

Chapter 3: Downloading and Installing Arduino
Installing Arduino
Installing Arduino for Windows
Installing Arduino for Mac OS X
Installing Arduino for Linux
Surveying the Arduino Environment

Chapter 4: Blinking an LED
Working with Your First Arduino Sketch
Finding the Blink Sketch
Identifying your board
Configuring the software
Uploading the sketch
Congratulate yourself!
What just happened?
Looking Closer at the Sketch
Comments
Declarations
Variables
Functions
Setup
Loop
Blinking Brighter
Tweaking the Sketch

Part II: Getting Physical with Arduino

Chapter 5: Tools of the Trade
Finding the Right Tools for the Job
Breadboard
Jump wires
Needle-nose pliers
Multimeter
Using the Multimeter to Measure Voltage, Current, and Resistance
Measuring voltage (in volts) in a circuit
Measuring current (in amps) in a circuit
Measuring resistance (in ohms) of a resistor
Measuring resistance (in ohms) of a variable resistor
Checking the continuity (in bleeps) of your circuit

Chapter 6: A Primer on Electricity and Circuitry
Understanding Electricity
Using Equations to Build Your Circuits
Ohm's Law
Calculating power
Joule's Law
Working with Circuit Diagrams
A simple circuit diagram
Using a circuit diagram with an Arduino
Color Coding
Datasheets
Resistor Color Charts

Chapter 7: Basic Sketches: Inputs, Outputs, and Communication
Uploading a Sketch
Using Pulse Width Modulation (PWM)
The LED Fade Sketch
Understanding the fade sketch
Tweaking the fade sketch
The Button Sketch
Understanding the Button sketch
Tweaking the Button sketch
The AnalogInput Sketch
Understanding the AnalogInput sketch
Tweaking the AnalogInput sketch
Talking Serial
The DigitalReadSerial Sketch
Understanding the DigitalReadSerial sketch
The AnalogInOutSerial Sketch
Understanding the AnalogInOutSerial sketch

Chapter 8: More Basic Sketches: Motion and Sound
Working with Electric Motors
Discovering Diodes
Spinning a DC Motor
The Motor sketch
Understanding the Motor sketch
Changing the Speed of Your Motor

The MotorSpeed sketch
Understanding the MotorSpeed sketch
Controlling the Speed of Your Motor
The MotorControl sketch
Understanding the MotorControl Sketch
Tweaking the MotorControl sketch
Getting to Know Servo Motors
Creating Sweeping Movements
The Sweep sketch
Understanding the Sweep sketch
Controlling Your Servo
The Knob sketch
Understanding the Knob sketch
Making Noises
Piezo buzzer
The toneMelody sketch.
Understanding the sketch
Making an Instrument
The PitchFollower sketch
Understanding the sketch

Part III: Building on the Basics

Chapter 9: Learning by Example
Skube
How it works
Further reading
Chorus
How it works
Further reading
Push Snowboarding
How it works
Further reading
Baker Tweet
How it works
Further reading
The National Maritime Museum's Compass Lounge and Compass Card
How it works
Further reading
The Good Night Lamp
How it works
Further reading
Little Printer
How it works
Further reading
Flap to Freedom
How it works
Further reading

Chapter 10: Soldering On
Understanding Soldering
Gathering What You Need for Soldering
Creating a workspace
Choosing a soldering iron
Solder
Third hand (helping hand)
Adhesive putty
Wire cutters
Wire strippers
Needle-nosed pliers
Multimeter
Solder sucker
Solder wick
Equipment wire
Staying Safe while Soldering
Handling your soldering iron
Keeping your eyes protected
Working in a ventilated environment
Cleaning your iron
Don't eat the solder!
Assembling a Shield
Laying out all the pieces of the circuit
Assembly
Header pins
Acquiring Your Soldering Technique
Building Your Circuit
Knowing your circuit
Laying out your circuit
Preparing your wire
Soldering your circuit
Cleaning up
Testing your shield
Packaging Your Project
Enclosures
Wiring
Securing the board and other elements

Chapter 11: Get ting Clever with Code
Blinking Better
Setting up the BlinkWithoutDelay sketch
Understanding the BlinkWithoutDelay sketch
Taking the Bounce Out of Your Button
Setting up the Debounce sketch
Understanding the Debounce sketch
Making a Better Button
Setting up the StateChangeDetection sketch
Understanding the StateChangeDetection sketch
Smoothing Your Sensors
Setting up the Smoothing sketch
Understanding the Smoothing sketch
Calibrating Your Inputs
Setting up the Calibration sketch
Understanding the Calibration sketch

Chapter 12: Common Sense with Common Sensors
Making Buttons Easier
Implementing the DigitalInputPullup sketch
Understanding the DigitalInputPullup sketch
Exploring Piezo Sensors
Implementing the Knock sketch
Understanding the Knock sketch
Utilizing Pressure, Force, and Load Sensors
Implementing the toneKeyboard sketch.
Understanding the toneKeyboard sketch
Sensing with Style
Implementing the CapPinSketch sketch
Understanding the CapPinSketch sketch
Tripping Along with Lasers
Implementing the AnalogInOutSerial sketch
Understanding the AnalogInOutSerial sketch
Detecting Movement
Implementing the DigitalReadSerial sketch
Understanding the DigitalReadSerial sketch
Measuring Distance
Implementing the MaxSonar sketch
Understanding the MaxSonar sketch
Testing, Testing . Can Anybody Hear This?
Implementing the AnalogInOutSerial sketch
Understanding the AnalogInOutSerial sketch

Part IV: Unlocking Your Arduino's Potential

Chapter 13: Becoming a Specialist with Shields and Libraries
Looking at Shields
Considering combinations
Reviewing the field
Staying current
Browsing the Libraries
Reviewing the standard libraries
Installing additional libraries
Obtaining contributed libraries

Chapter 14: Sensing More Inputs and Controlling More Outputs
Controlling Multiple LEDs
Implementing the AnalogWriteMega sketch
Understanding the AnalogWriteMega Sketch
Tweaking the AnalogWriteMega sketch
Controlling Lots of LEDs by Shifting Out
Implementing the shiftOutCode, Hello World sketch
Understanding the shiftOutCode, Hello World sketch
Tweaking the shiftOutCode, Hello World sketch
Doing more with the same cirduit

Chapter 15: Multiplying Your Outputs with I2C
What Is I2C?
Assembling the I2C PWM/Servo Driver
Using the I2C PWM/Servo Driver
Understanding the I2C PWM/Servo Driver Sketch
Buying Servo Motors
Other Uses for I2C

Part V: Sussing Out Software

Chapter 16: Getting to Know Processing
Looking Under the Hood
Installing Processing
Taking a look at Processing
Trying Your First Processing Sketch
Drawing shapes
Changing color and opacity
Playing with interaction

Chapter 17: Processing the Physical World

Making a Virtual Button
Setting up the Arduino code
Setting up the Processing code
Understanding the Processing PhysicalPixel sketch
Understanding the Arduino Physical Pixel sketch
Drawing a Graph
Setting up the Arduino code
Setting up the Processing code
Understanding the Arduino Graph sketch
Understanding the Processing Graph sketch
Sending Multiple Signals
Setting up the Arduino code
Setting up the Processing code
Understanding the Arduino SerialCallResponse sketch
Understanding the Processing SerialCallResponse sketch

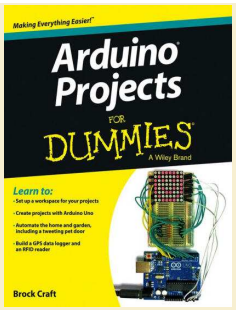
Part VI: The Part of Tens .

Chapter 18: Ten Places to Learn More about Arduino
Arduino Blog
Hack a Day
SparkFun
MAKE
Adafruit
Bildr
Instructables
YouTube
Hackerspaces
Forum
Friends, Colleagues, and Workshops

Chapter 19: Ten Great Shops to Know
Shops in the United Kingdom
SK Pang
Technobots
Proto-PIC
Oomlout
RoboSavvy
Active Robots
Shops around the World
Adafruit (U.S.)
Arduino Store (Italy)
Seeed Studio (China)
SparkFun (U.S.)

Chapter 20: Ten Places to Find Parts and Components
RS Components (World)
Farnell (World)
Rapid (World)
Digi-Key (World)
eBay (World)
Maplin (U.K.)
RadioShack (U.S.)
Ultraleds (U.K.)
EnvironmentalLights.com (U.S.)
Skip/Dumpster Diving (World)

Index



Brock Craft

Arduino Projects For Dummies

Arduino проекты для «чайников»

— Publisher: John Wiley & Sons, Ltd., 2013. — 404 pages
— Series: For Dummies

ISBN-10: 1118551478
ISBN-13: 978-1118551479

Узнайте все удивительные вещи, которые вы можете сделать с Arduino.

Arduino – это плата, которую используют все: от ученых, программистов и аппаратных хакеров, до художников, дизайнеров, любителей и инженеров для того, чтобы добавить интерактивности объектам и проектам и экспериментировать с программированием и электроникой. Эта легкая для понимания книга – идеальна для того, чтобы узнать больше об обширных возможностях Arduino.

CONTENTS

Introduction

Why Arduino?
Foolish Assumptions
Safety and Arduino Projects
How This Book Is Organized
Part I: Getting Started with Arduino Projects
Part II: Basic Arduino Projects
Part III: The Interactive Home and Garden
Part IV: Advanced Arduino Projects
Part V: The Part of Tens
The Companion Website
Icons Used in This Book

Part I: Getting Started with Arduino Projects

Chapter 1: Exploring the World of Arduino
About Arduino
Discovering Who Uses Arduino
Arduino in education
Arduino in the corporate world
Making and hacking communities
Understanding Microcontrollers
Using tiny computers to do useful stuff
Getting Started

Chapter 2: Setting Up Your Workspace and Tools
Preparing to Bui ld
Setting up your workspace
Selecting Basic Tools
Selecting and using your multimeter
Selecting and using a power supply
Understanding electricity and safety
Working with breadboards, stripboards, and perfboards
Choosing Your Soldering Iron and Accessories
Selecting Project Boxes and Housings
Choosing Your Arduino or Arduino Kit
Getting to know Arduino shields
Setting Up Your Arduino on Your Computer
Installing the Arduino IDE
Installing drivers on Windows computers
Installing Arduino drivers on Linux

Chapter 3: Understanding the Basics
Understanding Key Concepts
Connecting your Arduino
Programming your Arduino using the IDE
Extending your reach with libraries
Powering your Arduino
Understanding Basic Electronics
Voltage (V)
Current (I)
Resistance (R)
Ohm's Law
So what?
Identifying Electronic Components
Reading schematic diagrams
Reading parts placement diagrams
Understanding Sensing and Actuating
Reading datasheets
Understanding and using sensors
Understanding and using actuators
Making Projects Work
Moving from your breadboard to your project box
Learning soldering basics

Part II: Basic Arduino Projects

Chapter 4: The All-Seeing Eye
Selecting Your Parts
Building the Circuit
Understanding How the Code Works
All in good time
Setting up the code
The main event
Stepping up and stepping down
Understanding How the Hardware Works
The Potential of a Potentiometer

Chapter 5: Making a Light Pet
Selecting Your Parts
Building the Circuit
Understanding How the Code Works
Making moods
Cranking out the code
Blending light
Fooling your eyes with pulse-width modulation
Testing the code
Upload and go
Tweaking it!
Understanding How the Hardware Works

Chapter 6: Making a Scrolling Sign
Selecting Parts
Building the Circuit
Understanding How the Code Works
Summoning a sprite
Animating sprites
Displaying scrolling text
Understanding How the Hardware Works
Troubleshooting
Getting creative

Chapter 7: Building an Arduino Clock
It's About Time!
Selecting and Preparing Your Parts
Assembling your RTC module
Adding and testing your LCD display
Displaying the time
Adding your input buttons and a switch
Adding your alarm

Part III: The Interactive Home and Garden

Chapter 8: Building a Keypad Entry System
Selecting and Preparing Your Parts
Selecting an electric lock mechanism
Prototyping your keypad and display
Coding and testing your keypad
Adding and testing your relay
Assembling and Installing Your System

Chapter 9: Building an RFID Tag Reader
Understanding RFID
About passive RFID
About active RFID
RFID frequencies and protocols
Building an ID-Innovations RFID Reader
Selecting your parts
Assembling your RFID reader
Programming your RFID reader
Testing and Setting Your RFID Reader

Chapter 10: Building an Automated Garden
Creating a Watering System
Selecting Your Parts
Building Your System
Building your moisture sensor
Building your reservoir
Running the water supply
Building the breadboard circuit
Coding, Calibrating, and Testing
Defining the setup
Running the main loop
Calibrating the sensor and flow rate
Adding more valves

Chapter 11 : Building a Tweeting Pet Door
Selecting Your Parts
Testing Your Circuit
Preparing Your Twitter Account
Crafting Your Code
Specifying your tweets
Adding libraries for Ethernet and Twitter
Adding your program logic
Modifying Your Pet Door

Chapter 12: Building a Home Sensing Station
Building Your Sensor Probes
Selecting your parts
Building and testing your circuit
Building your sensor probes
Building your sensor shield
Creating a Xively Account
Programming Your Sensing Station
Understanding the code
Understanding the main loop
Making sense of your sensor readings

Part IV: Advanced Arduino Projects

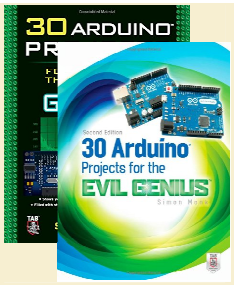
Chapter 13: Building a GPS Data Logger
Understanding GPS
Selecting Your Parts
Building Your Project
Assembling and testing the GPS shield
Programming your data logger
Testing your data logger
Making the enclosure
Collecting and Plotting GPS Data
Tracking your path
Plotting your data

Chapter 14: Building a Remote-Controlled Car
Selecting and Preparing Your Parts
Building Your Detector and Drive
Building your circuit on the breadboard
Coding the detector
Reading your remote control codes
Coding the drive motors
Testing the drive motors
Building Your Chassis

Chapter 15: Building an LED Cube
Selecting Your Parts
Building Your Cube
Assembling the LED matrix
Fabricating the enclosure
Programming Your Cube
Variable declarations
Setup
The main loop
Using the LED Cube Pattern Designer
Part V: The Part of Tens

Chapter 16: Ten Great Arduino Resources
Websites
Arduino.cc and related forums
Fritzing
Hack-a-day
Instructables
Learn.adafruit.com
Make:
element14
YouTube
Books and eBooks
Arduino For Dummies
The Arduino Cookbook
Making Things Talk

Chapter 17: Ten Troubleshooting Tips
Troubleshooting Your Hardware
Checking Your Connections
Confirming Your Power Is Correct
Hunting for Odors and Hot Components
Test Your Outputs on External Devices
Testing Your Digital Pins
Troubleshooting Your Software
Checking Your Syntax
Using the Serial Monitor
Checking Your Inputs and Outputs
Using a Simulator or an Emulator
When All Else Fails ...



Simon Monk

30 Arduino Projects for the Evil Genius

30 проектов на Arduino для Злого Гения

— Publisher: McGraw-Hill/TAB Electronics — Series: Evil Genius
— 1 edition 2010 — 208 p. ISBN-10: 007174133X ISBN-13: 978-0071741330
— 2 edition 2013 — 224 p. ISBN-10: 0071817727 ISBN-13: 978-0071817721

Это руководство покажет, как программировать и собирать разнообразные проекты с платой Arduino. Вы узнаете, как подключить плату Arduino к компьютеру, запрограммировать её, чтобы создать свои собственные хитрые устройства.

CONTENTS

Introduction

1. Quickstart
Powering Up
Installing the Software
Configuring Your Arduino Environment
Downloading the Project Software
Project 1 Flashing LED
Breadboard
Summary

2. A Tour of Arduino
Microcontrollers
What's on an Arduino Board?
The Arduino Family
The C Language
Summary

3. LED Projects
Project 2 Morse Code S.O.S. Flasher
Loops
Arrays
Project 3 Morse Code Translator
Project 4 High-Brightness Morse Code Translator
Summary

4. More LED Projects
Digital Inputs and Outputs
Project 5 Model Traffic Signal
Project 6 Strobe Light
Project 7 S.A.D. Light
Project 8 High-Powered Strobe Light
Random Number Generation
Project 9 LED Dice
Summary

5. Sensor Projects
Project 10 Keypad Security Code
Rotary Encoders
Project 11 Model Traffic Signal Using a Rotary Encoder
Sensing Light
Project 12 Pulse Rate Monitor
Measuring Temperature
Project 13 USB Temperature Logger
Summary

6. Light Projects
Project 14 Multicolor Light Display
Seven-Segment LEDs
Project 15 Seven-Segment LED Double Dice
Project 16 LED Array
LCD Displays
Project 17 USB Message Board
Summary

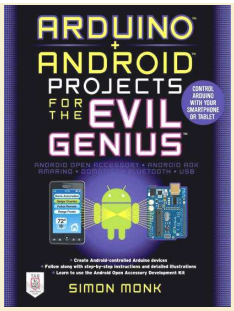
7. Sound Projects
Project 18 Oscilloscope
Sound Generation
Project 19 Tune Player
Project 20 Light Harp
Project 21 VU Meter
Summary

8. Power Projects
Project 22 LCD Thermostat
Project 23 Computer-Controlled Fan
H-Bridge Controllers
Project 24 Hypnotizer
Servo Motors
Project 25 Servo-Controlled Laser
Summary

9. Miscellaneous Projects
Project 26 Lie Detector
Project 27 Magnetic Door Lock
Project 28 Infrared Remote
Project 29 Lilypad Clock
Project 30 Evil Genius Countdown Timer
Summary

10. Your Projects
Circuits
Components
Tools
Project Ideas

Appendix: Components and Supplies
Suppliers
Starter Kit of Components



Simon Monk

Arduino + Android Projects for the Evil Genius: Control Arduino With Your Smartphone or Tablet
Arduino + Android проекты для Злого Гения:
управление Arduino с помощью смартфона или планшета

— Publisher: Tab Books, 2011. —197 pages
— Series: Evil Genius

ISBN-10: 007177596X
ISBN-13: 978-0071775960

Эта книга покажет, как создать устройства Ардуино и управлять ими со смартфонов и планшетов на базе Android. Даётся информация по Android (ADK), USB и объясняется, как использовать их с основной платформой Arduino. А также методы связи между Android и Arduino, которые не требуют ADK, включая звук, Bluetooth и Wi-Fi / Ethernet.

CONTENTS

Acknowledgments
Introduction

PART ONE. Android Peripherals

1. Bluetooth Robot
Construction
Theory
Summary

2. Android Geiger Counter
Google Open Accessory
Construction
Theory
Summary

3. Android Light Show
Construction: The Droid Accessory Base
Construction: The Light Show Project
Using the Project
Theory
Summary

4. TV Remote
Construction
Using the Project
Theory
Summary

5. Temperature Logger
Construction
Using the Project
Theory
Summary

6. Ultrasonic Range Finder
Construction
Using the Project
Theory
Summary

PART TWO. Home Automation

7. Home Automation Controller
The Sound Link Module
Android Software
Internet Access
Theory
Summary

8. Power Control
Power Control Electronics
Constructing the Power Control Module
Adding It to the Home Automation Controller
Setting Up Your Home
Theory
Summary

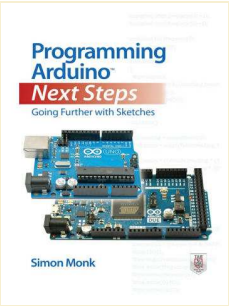
9. Smart Thermostat
Construction
Using the System
Theory
Summary

10. RFID Door Lock
Construction
Using the System
Theory
Summary

11. Signaling Flags
Construction
Theory
Summary

12. Delay Timer
Construction
Theory
Summary

Appendix: Open Accessory Primer
Learning Android Programming
Arduino Programming
The Example
On the Arduino
Android
Conclusion



Simon Monk

Programming Arduino Next Steps: Going Further with Sketches

Следующие шаги в программировании Arduino: погружаемся в скетчи

— Publisher: McGraw-Hill/TAB Electronics, 2013. — 288 pages
— Series: Tab

ISBN-10: 0071830251
ISBN-13: 978-0071830256

В этом практическом руководстве автор раскрывает профессиональные секреты программирования Arduino: использование прерываний, управление памятью, программирование для Интернета, применение интерфейсов I2C, 1-Wire, SPI и многое другое.

CONTENTS

Introduction

1. Programming Arduino
What Is Arduino?
Installation and the IDE
Installing the IDE
Blink
A Tour of Arduino
Power Supply
Power Connections
Analog Inputs
Digital Connections
Arduino Boards
Uno and Similar
Big Arduino Boards
Small Arduino Boards
LilyPad and LilyPad USB Boards
Unofficial Arduinos
Programming Language
Modifying the Blink Sketch
Variables
If
Loops
Functions
Digital Inputs
Digital Outputs
The Serial Monitor
Arrays and Strings
Analog Inputs
Analog Outputs
Using Libraries
Arduino Data Types
Arduino Commands

2. Under the Hood
A Brief History of Arduino
Anatomy of an Arduino
AVR Processors
ATmega328
ATmega32u4
ATmega2560
AT91SAM3X8E
Arduino and Wiring
From Sketch to Arduino
AVR Studio
Installing a Bootloader
Burning a Bootloader with AVR Studio and a Programmer
Burning a Bootloader with the Arduino IDE and a Second Arduino

3. Interrupts and Timers
Hardware Interrupts
Interrupt Pins
Interrupt Modes
Enabling Internal Pull-Up
Interrupt Service Routines
Volatile Variables
ISR Enabling and Disabling Interrupts
Timer Interrupts

4. Making Arduino Faster
How Fast Is an Arduino?
Comparing Arduino Boards
Speeding Up Arithmetic
Do You Really Need to Use a Float?
Lookup vs. Calculate
Fast I/O
Basic Code Optimization
Bytes and Bits
ATmega328 Ports
Very Fast Digital Output
Fast Digital Input
Speeding Up Analog Inputs

5. Low Power Arduino
Power Consumption of Arduino Boards
Current and Batteries
Reducing the Clock Speed
Turning Things Off
Sleeping
Narcoleptic
Waking on External Interrupts
Use Digital Outputs to Control Power

6. Memory
Arduino Memory
Minimizing RAM Usage
Use the Right Data Structures
Be Careful with Recursion
Store String Constants in Flash Memory
Common Misconceptions
Measure Free Memory
Minimizing Flash Usage
Use Constants
Remove Unwanted Trace
Bypass the Bootloader
Static vs. Dynamic Memory Allocation
Strings
C char Arrays
The Arduino String Object Library
Using EEPROM
EEPROM Example
Using the avr/eeprom.h Library
EEPROM Limitations
Using Flash
Using SD Card Storage

7. Using I2C
I2C Hardware
The I2C Protocol
The Wire Library
Initializing I2C
Master Sending Data
Master Receiving Data
I2C Examples
TEA5767 FM Radio
Arduino-to-Arduino Communication
LED Backpack Boards
DS1307 Real-Time Clock

8. Interfacing with 1-Wire Devices
1-Wire Hardware
The 1-Wire Protocol
The OneWire Library
Initializing 1-Wire
Scanning the Bus
Using the DS18B20

9. Interfacing with SPI Devices
Bit Manipulation
Binary and Hex
Masking Bits
Shifting Bits
SPI Hardware
The SPI Protocol
The SPI Library
SPI Example

10. Serial UART Programming
Serial Hardware
Serial Protocol
The Serial Commands
The SoftwareSerial Library
Serial Examples
Computer to Arduino over USB
Arduino to Arduino
GPS Module

11. USB Programming
Keyboard and Mouse Emulation
Keyboard Emulation
Keyboard Emulation Example
Mouse Emulation
Mouse Emulation Example
USB Host Programming
USB Host Shield and Library
USB Host on the Arduino Due

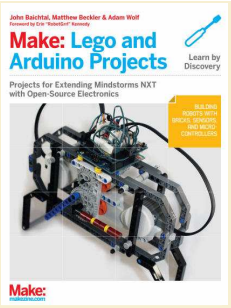
12. Network Programming
Networking Hardware
Ethernet Shield
Arduino Ethernet/EtherTen
Arduino and WiFi
The Ethernet Library
Making a Connection
Setting Up a Web Server
Making Requests
Ethernet Examples
Physical Web Server
Using a JSON Web Service
The WiFi Library
Making a Connection
WiFi Specific Functions
WiFi Example

13. Digital Signal Processing
Introducing Digital Signal Processing
Averaging Readings
An Introduction to Filtering
Creating a Simple Low-Pass Filter
Arduino Uno DSP
Arduino Due DSP
Filter Code Generation
The Fourier Transform
Spectrum Analyzer Example
Frequency Measurement Example

14. Managing with One Process
Making the Transition from Big Programming
Why You Don't Need Threads
Setup and Loop
Sense Then Act
Pause Without Blocking
The Timer Library

15. Writing Libraries
When to Make a Library
Using Classes and Methods
Library Example (TEA5767 Radio)
Define the API
Write the Header File
Write the Implementation File
Write the Keywords File
Make the Examples Folder
Testing the Library
Releasing the Library
A Parts
Arduino Boards
Shields
Components and Modules
Suppliers

Index



John Baichtal, Matthew Beckler, Adam Wolf

LEGO and Arduino Projects: Projects for extending MINDSTORMS NXT with open-source electronics

LEGO и проекты Arduino: проекты для расширения «MINDSTORMS NXT» с open-source электроникой

— Publisher: Maker Media, Inc, 2012. — 328 pages

ISBN-10: 1449321062
ISBN-13: 978-1449321062

Делайте удивительных роботов и невероятные гаджеты, с двумя самыми распространенными «сделай сам» технологиями: LEGO и Arduino. С этим руководством Вы изучите, как создавать устройства с Lego Mindstorms NXT 2.0, Arduino и некоторыми другими компонентами.

CONTENTS

Foreword.

Preface

1 Project: Drawbot
Parts List
Assembly Instructions
Program the Robot
The Next Chapter

2 Anatomy of Lego Robotics
Mindstorms
Expanding on the Mindstorms Set
Programming Your Brick
The Next Chapter

3 Arduino Interlude
History of the Arduino Project
What Is OSHW?
Anatomy of the Uno
The Arduino Ecosystem
Arduino Resources
The Next Chapter

4 Project: Clock
Parts List
Assembly Instructions
Program the Robot
Setting the Clock
The Next Chapter

5 Project: Chocolate Milk Maker.
Parts List
Assembly Instructions
Program the Robot
The Next Chapter

6 Basic Electronic Theory
Basic Concepts in Electricity
Know Your Electronic Components
Sensors 101
Further Study

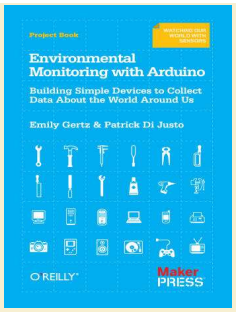
7 Gripperbot
Parts List
Building Instructions
Assembling the Gripperbot's Electronics
Programming the Gripperbot
The Next Chapter

8 Project: Keytar
Parts List
Assembly Instructions
Program the Keytar
Play Some Music!
The Next Chapter

9 Project: Lamp
Parts List
Assembly Instructions
Program the Lamp
The Next Chapter

10 Advanced Techniques
Wiring Up Bricktronics Equivalents
Mounting PCBs to Legos
Adding Molex Connectors to Lego Wires
All About Motors
Powering Your Robot
NXT to Arduino Communication
XBee Wireless Modules
Epilogue

Index



Emily Gertz, Patrick Di Justo

Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us
Экологический мониторинг с Arduino: создание простых устройств для сбора данных о мире вокруг нас

— Publisher: Maker Media, 2012. — 100 pages

ISBN-10: 1449310567
ISBN-13: 978-1449310561

После разрушительного цунами в 2011г. самодельщики Японии создали свои собственные устройства для мониторинга уровня радиации, а затем выложили их конструкции в Интернете. Прямо сейчас тысячи людей во всём мире отслеживают условия окружающей среды с устройствами контроля, которые они создали сами. Вы можете сами повторить их конструкцию с платой Arduino.

CONTENTS

Preface

1 - The World's Shortest Electronics Primer
What Is Arduino?
Electronic Circuits and Components
Programming Arduino
First Sketch: Make an LED Blink
Parts
Install the IDE6
Breadboard the Circuit6
Write the Code
Things to Try

2 - Project: Noise Monitor - LED Bar Output
Measuring Noise: The Microphone
The LED Bar
Make the Gadget
Parts
Breadboard the Circuit14
Write the Code
Things to Try

3 - New Component: 4Char Display
Test Project20
Parts
Breadboard the Circuit
Write the Code
Things to Try

4 - Detecting Electromagnetic Interference (and making bad music)
Detecting EMI Sources in the Environment
Make the Gadget27
Parts
The 8-Ohm Speaker
Construct the EMI Monitor
Write the Code
Run the Sketch31
Powering the Gadget in Mobile Mode
What Are We Measuring with This Gadget?
Things to Try

5 - Project: Water Conductivity - Numerical Output
What Is Conductivity, and Why Do I Care?
Make the Gadget35
Parts
Construct the Probe36
Breadboard the Circuit
Write the Code
How to Take a Reading
Things to Try

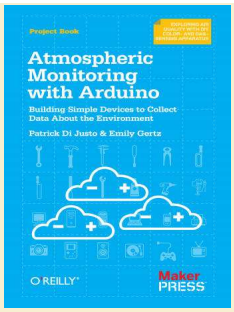
6 - New Component: Ethernet Shield
Using the Ethernet Shield
The Ethernet Port44
The MAC Address44
The IP Address
The SD Card Slot
Testing the Ethernet Shield
Parts49
Assembly
Testing the SD Card Slot50
Parts
Assembly
Things to Try

7 - Project: Humidity, Temperature & Dew Point - 4Char Display
You Don't Have to Be a Weatherman to Measure the Weather
Getting Usable Measurements
First Electronic Sensor: The DHT-2254
Using Code Libraries
Make the Gadget55
Parts
Breadboard the Circuit
Write the Code
Things to Try

8 - Real-Time, Geo-Tagged Data Sharing with Pachube
Test Project: Connecting and Uploading Data to Pachube
Parts64
Open a Pachube Account
Write the Code
Things To Try

9 - Project: Radiation Counter - Sharing Data on the Internet
What's a Geiger Counter?
Make the Gadget70
Parts
Breadboard the Circuit
Write the Code72
What Are We Measuring with This Gadget?
Failure Mode Analysis77
Things to Try

10 - Casing the Gadget



Emily Gertz, Patrick Di Justo

Atmospheric Monitoring with Arduino: Building Simple Devices to Collect Data About the Environment
Атмосферный контроль с Arduino: создание простых устройств для сбора данных об окружающей среде

— Publisher: Maker Media, 2012. — 90 pages

ISBN-10: 1449338143
ISBN-13: 978-1449338145

Производители во всём мире создают недорогие устройства для контроля окружающей среды. С этим практическим руководством вы тоже сможете это сделать. Вы изучите, как создать гаджеты для исследования качества нашей атмосферы, используя Arduino и несколько недорогих датчиков.

CONTENTS

Preface

1 - The World's Shortest Electronics Primer
What Is Arduino?
Electronic Circuits and Components
Programming Arduino
First Sketch: Make an LED Blink
Parts
Install the IDE
Breadboard the Circuit
Write the Code
Things to Try

2 - Gadget: Tropospheric Gas Detector
How Gas Sensors Work
Which Gases Can We Monitor?
How This Gadget Works
Transistorized!
Build the Gadget
Load the Sketch
Displaying and Storing Your Data
Liquid Crystal Displays
Reading Data Off EEPROM
Reading Data from an SD Card
Things to Try
Other Sensors
Solar Powered
GSM
Do Not Deploy Your Gadget in Public Without Official Permission
Get Official Permission
Get Your Community Involved

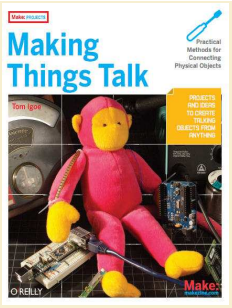
3 - A Brief Introduction to LEDs
What Is a Diode?
What Is a Light Emitting Diode?
How Are We Using LEDs in the LED Photometer?

4 - Gadget: LED Sensitivity Tester
Mission: Inputtable
Build the Gadget

5 - Gadget: LED Photometer
Build the Gadget
Load the Sketch
Calibrate the Gadget: Air Mass, Atmospheric Optical Thickness, and
Extraterrestrial Constant
Calculating Atmospheric Optical Thickness
Things to Try
Detecting "Ozone Holes": Measuring the Ozone Layer
Add an Accelerometer

6 - Using the LED Photometer
Atmospheric Aerosols
Photosynthetically Active Radiation (PAR)
Water Vapor (WV)
Extracting Data from the LED Photometer
Graphing Data in a Spreadsheet
Sending Data to COSM

7 - Doing Science: How to Learn More from Your Atmospheric Data
The Scientific Method
Steps in the Scientific Method
Observe Something in the World
Ask an Answerable Question
Formulate a Hypothesis
Compare the Predicted to Actual Results, Considering the Results
Ask Another Question



Tom Igoe

Making Things Talk: Practical Methods for Connecting Physical Objects
Устанавливаем связь:
практические методы подключения физических объектов

— Publisher: Maker Media, 2007. — 432 pages
— 1 edition

ISBN-10: 0596510519
ISBN-13: 978-0596510510

Эта книга идеально подходит для людей с незначительной технической подготовкой, но большим интересом к теме. На практике показано как связать между собой различные датчики и исполнительные механизмы используя ZigBee, Bluetooth и Интернет.

CONTENTS

Preface
Who This Book Is For
What You Need To Know
Contents of This Book
On Buying Parts
Using Code Examples
Using Circuit Examples
Acknowledgments
We'd Like to Hear from You

Chapter 1: The Tools
It Starts with the Stuff You Touch
It's About Pulses
Computers of All Shapes and Sizes
Good Habits
Tools
Using the Command Line
It Ends with the Stuff You Touch

Chapter 2: The Simplest Network
Layers of Agreement
Making the Connection: The Lower Layers
Saying Something: The Application Layers
Project 1: Monski Pong
Flow Control
Project 2: Wireless Monski Pong
Project 3: Negotiating in Bluetooth
Conclusion

Chapter 3: A More Complex Network
Network Maps and Addresses
Clients, Servers, and Message Protocols
Project 4: A Networked Cat
Conclusion

Chapter 4: Look Ma! No Computer
Introducing Network Modules
Project 5: Hello Internet!
An Embedded Network Client Application
Project 6: Networked Air Quality Meter
Serial-to-Ethernet Modules: Programming and Troubleshooting Tools
Conclusion

Chapter 5: Communicating in (Near) Real Time
Interactive Systems and Feedback Loops
Transmission Control Protocol: Sockets & Sessions
Project 7: A Networked Game
Conclusion

Chapter 6: Wireless Communication
Why Isn't Everything Wireless?
Two Flavors of Wireless: Infrared and Radio
Project 8: Infrared Transmitter-Receiver Pair
Project 9: Radio Transmitter-Receiver Pair
Project 10: Duplex Radio Transmission
An XBee Serial Terminal
Project 11: Bluetooth Transceivers
What About Wi-Fi?
Buying Radios
Conclusion

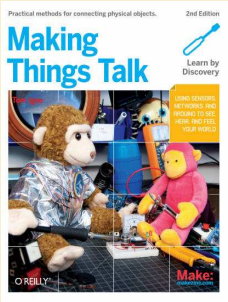
Chapter 7: The Tools
Look, Ma: No Microcontroller!
Who's Out There? Broadcast Messages
Project 12: Reporting Toxic Chemicals in the Shop
Directed Messages
Project 13: Relaying Solar Cell Data Wirelessly
Conclusion

Chapter 8: How to Locate (Almost) Anything
Network Location and Physical Location
Determining Distance
Project 14: Infrared Distance Ranger Example
Project 15: Ultrasonic Distance Ranger Example
Project 16: Reading Received Signal Strength Using XBee Radios
Project 17: Reading Received Signal Strength Using Bluetooth Radios
Determining Position Through Trilateration
Project 18: Reading the GPS Serial Protocol
Determining Orientation
Project 19: Determining Heading Using a Digital Compass
Project 20: Determining Attitude Using an Accelerometer
Conclusion

Chapter 9: Identification
Physical Identification
Project 21: Color Recognition Using a Webcam
Project 22: 2D Barcode Recognition Using a Webcam
Project 23: Reading RFID Tags in Processing
Project 24: RFID Meets Home Automation
Network Identification
Project 25: IP Geocoding
Project 26: Email from RFID
Conclusion
Appendix A: And Another Thing
Other Useful Protocols
Proxies of All Kinds
Mobile Phone Application Development
Other Microcontrollers
New Tools

Appendix B: Where to Get Stuff
Hardware
Software
Appendix C: Program Listings

Index



Tom Igoe

Making Things Talk: Using Sensors, Networks, and Arduino to see, hear, and feel your world

Устанавливаем связь: использование датчиков, сетей и Arduino, чтобы видеть, слышать и чувствовать окружающий мир

— Publisher: Maker Media, 2011. — 496 pages

ISBN-10: 1449392431
ISBN-13: 978-1449392437

На 33-х простых проектах будет показано, как соединить различные датчики и исполнительные механизмы с платой Arduino. Вы научитесь обрабатывать принятые данные, отображать их и т.п.

CONTENTS

Who This Book Is For
Using Code Examples
Using Circuit Examples
Acknowledgments for the First Edition

Chapter 1: The Tools
It Starts with the Stuff You Touch
It's About Pulses
Computers of All Shapes and Sizes
Good Habits
Tools
Using the Command Line
Using an Oscilloscope
It Ends with the Stuff You Touch

Chapter 2: The Simplest Network
Supplies for Chapter 2
Layers of Agreement
Making the Connection: The Lower Layers
Project 1: Type Brighter
Project 2: Monski Pong
Flow Control
Project 3: Wireless Monski Pong
Project 4: Negotiating in Bluetooth
Conclusion

Chapter 3: A More Complex Network
Supplies for Chapter 3
Network Maps and Addresses
Project 5: Networked Cat
Conclusion

Chapter 4: Look, Ma, No Computer! Microcontrollers on the Internet
Supplies for Chapter 4
Introducing Network Modules
Project 6: Hello Internet!
An Embedded Network Client Application
Project 7: Networked Air-Quality Meter
Programming and Troubleshooting Tools for Embedded Modules
Conclusion

Chapter 5: Communicating in (Near) Real Time
Supplies for Chapter 5
Interactive Systems and Feedback Loops
Transmission Control Protocol: Sockets & Sessions
Project 8: Networked Pong
The Clients
Conclusion

Chapter 6: Wireless Communication
Supplies for Chapter 6
Why Isn't Everything Wireless?
Two Flavors of Wireless: Infrared and Radio
Project 9: Infrared Control of a Digital Camera
How Radio Works
Project 10: Duplex Radio Transmission
Project 11: Bluetooth Transceivers
Buying Radios
What About WiFi?
Project 12: Hello WiFi!
Conclusion

Chapter 7: Sessionless Networks
Supplies for Chapter 7
Sessions vs. Messages
Who's Out There? Broadcast Messages
Project 13: Reporting Toxic Chemicals in the Shop
Directed Messages
Project 14: Relaying Solar Cell Data Wirelessly
Conclusion

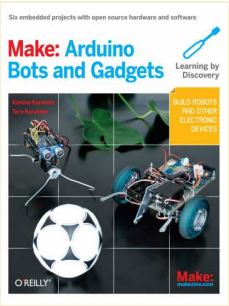
Chapter 8: How to Locate (Almost) Anything
Supplies for Chapter 8
Network Location and Physical Location
Determining Distance
Project 15: Infrared Distance Ranger Example
Project 16: Ultrasonic Distance Ranger Example
Project 17: Reading Received Signal Strength Using XBee Radios
Project 18: Reading Received Signal Strength Using Bluetooth Radios
Determining Position Through Trilateration
Project 19: Reading the GPS Serial Protocol
Determining Orientation
Project 20: Determining Heading Using a Digital Compass
Project 21: Determining Attitude Using an Accelerometer
Conclusion

Chapter 9: Identification
Supplies for Chapter 9
Physical Identification
Project 22: Color Recognition Using a Webcam
Project 23: Face Detection Using a Webcam
Project 24: 2D Barcode Recognition Using a Webcam
Project 25: Reading RFID Tags in Processing
Project 26: RFID Meets Home Automation
Project 27: Tweets from RFID
Network Identification
Project 28: IP Geocoding
Conclusion

Chapter 10: Mobile Phone Networks and the Physical World
Supplies for Chapter 10
One Big Network
Project 29: CatCam Redux
Project 30: Phoning the Thermostat
Text-Messaging Interfaces
Native Applications for Mobile Phones
Project 31: Personal Mobile Datalogger
Conclusion

Chapter 11: Protocols Revisited
Supplies for Chapter 11
Make the Connections
Text or Binary?
MIDI
Project 32: Fun with MIDI
Representational State Transfer
Project 33: Fun with REST
Conclusion

Appendix: Where to Get Stuff
Supplies
Hardware
Software



Tero Karvinen, Kimmo Karvinen

Arduino Bots and Gadgets: Six Embedded Projects with Open Source Hardware and Software

Arduino роботы и гаджеты: шесть проектов для встраиваемых систем с открытым аппаратным и программным обеспечением

— Publisher: Maker Media, Inc, 2011. — 296 pages
— Series: Learning by Discovery

ISBN-10: 1430238828
ISBN-13: 978-1430238829

Создавать с Arduino собственные полезные устройства, роботы, гаджеты легко даже для новичков. Встраиваемые системы есть везде - в автомобилях, детских игрушках и мобильных телефонах. Эта книга научит вас основам встраиваемых систем и поможет построить свой первый гаджет всего за несколько дней.

CONTENTS

1. Introduction
Building Philosophy
Reusing Parts
Buying Components
Useful Tools
Electronic Circuit Theory Review

2. Arduino: The Brains of an Embedded System
Why Arduino?
Starting with Arduino
Hello World with Arduino
Structure of "Hello World"
Arduino Uno
Arduino Nano

3. Stalker Guard
What You'll Learn
Tools and Parts
Solderless Breadboard
Jumper Wire
Ping Ultrasonic Sensor
Vibration Motor
Combining Components to Make the Stalker Guard
Making the Motor Vibrate
Providing Power from a Battery
What's Next?
Making an Enclosure

4. Insect Robot
What You'll Learn
Tools and Parts
Servo Motors
Constructing the Frame
Programming the Walk
Avoiding Obstacles Using Ultrasound
What's Next?

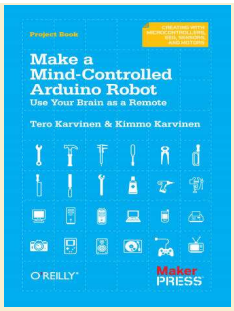
5. Interactive Painting
What You'll Learn
Tools and Parts
Resistors
LEDs
Detecting Motion Using Ultrasonic Sensors
Moving Images
Installing Python
Hello World in Python
Communicating over the Serial Port
Displaying a Picture
Scaling an Image to Full Screen
Changing Images with Button Control
Gesture-Controlled Painting in Full Screen
Animating the Sliding Image
Connecting Arduino with Processing
Processing Code for the Painting
The Finished Painting
Creating an Enclosure
Building a Frame

6. Boxing Clock
What You'll Learn
Tools and Parts
Android Software Installation
Creating a Boxing Clock in Android
What's Next?

7. Remote for a Smart Home
What You'll Learn
Tools and Parts
The Relay: A Controllable Switch
Hacking the Remote Control
Controlling the Arduino from the Computer
Creating a Graphical User Interface
The Finished Remote Control Interface
Creating an Enclosure

8. Soccer Robot
What You Will Learn
Tools and Parts
Continuous Rotation Servos
Modding a Standard Servo into a Continuous Rotation Servo
Connecting the Arduino to the Bluetooth Mate
Testing the Bluetooth Connection
Building a Frame for the Robot
Programming the Movements
Controlling Movement from a Computer
Steering with an Android Cell Phone
The Accelerometer
An Easier Approach to Bluetooth
Controlling the Robot with Cell Phone Motion
Completing the Soccer Robot
What's Next?

A. tBlue Library for Android



Tero Karvinen, Kimmo Karvinen

Make a Mind-Controlled Arduino Robot: Use Your Brain as a Remote
Создайте разумного робота на Arduino: используйте свой мозг удалённо

— Publisher: aker Media, 2011. — 90 pages
— Series: Creating With Microcontrollers Eeg, Sensors, and Motors

ISBN-10: 1449311547
ISBN-13: 978-1449311544

Создание робота, который реагирует на электрические возмущения в вашем мозге — это просто и забавно. Эта книга покажет, как построить робота, который издаёт звуки, мигает светом и реагирует на сигналы от электроэнцефалограммы головного мозга. Вы найдете полные инструкции для создания простого шасси робота с сервомоторами, колесами, датчиками, светодиодами и динамиком.

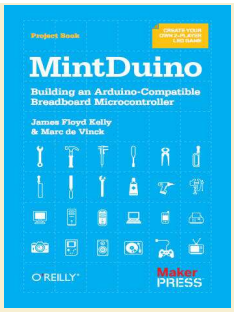
CONTENTS

Preface

1 - Building the Chassis
Tools and Parts
Parts
Tools
Servo Motors
Attaching Servos
Chassis
Making Holes
Painting the Chassis
Attaching Servos to the Chassis
Attaching the Line-Detecting Sensor
Wheels
Attaching the RGB LED to Chassis
Attaching the Power Switch to the Chassis
Attaching Arduino
Battery Holder
Attaching Solderless Breadboard
ScrewShield Holds Wires in Place

2 - Coding
Moving
Connect Servos
Hello Servo
Calibrate Stopping Point
Full Speed Forward
Other Ways to Control Servos
Line Avoidance
Connect the Reflection Sensor
Hello Reflection
Don't Cross the Black Line
Battery, No Strings Attached
Choosing Rechargeable Batteries
Connecting the Battery and Power Key
Bells and Whistles
Red, Green, and Blue LED
Beeping Piezo
Setting Threshold with a Potentiometer
Everything But Your Mind
Code Structure
Measuring Your Brains with MindWave
Hack MindWave Dongle
Level Conversion with Resistors
Hello Attention!
NeuroSky Protocol
Complete Mind-Controlled Robot

Appendix: Building the ScrewShield



James Floyd Kelly, Marc de Vinck

MintDuino: Building an Arduino-Compatible Breadboard Microcontroller

MintDuino: создание Arduino-совместимой макетной платы микроконтроллера

— Publisher: Maker Media, 2011. — 58 pages

ISBN-10: 1449307663
ISBN-13: 978-1449307660

Эта практическая книга покажет вам как разработать полный проект MintDuino от начала до конца. Изучите как собрать микроконтроллер на «макетке» (пайка не требуется) и сразу начать программировать его.

CONTENTS

Preface

1 - Build a Mintronics: MintDuino
Build the Power Supply
Power the Microcontroller
Getting Ready to Program

2 - The MintDuino Reflex Game

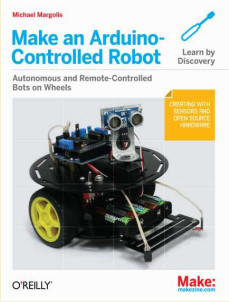
3 - Subtask 1: Light an LED
Start Building
Upload Your First Sketch

4 - Subtask 2: Randomly Light an LED
The Game Takes Shape

5 - Subtask 3: Light an LED with a Pushbutton
Light the LED

6 - Subtask 4: Add Buttons and LEDs
Light LEDs for Each Player

7 - Subtask 5: Program the Game
The Final Sketch
Conclusion



Dale Wheat

Make an Arduino–Controlled Robot

Создаём Arduino–управляемого робота

— Publisher: Maker Media, Inc, 2012. — 256 pages
— Series: Make: Projects (Book 2314)

ISBN-10: 1449344372
ISBN-13: 978-1449344375

С помощью этой книги, платы Arduino и программной среды вы узнаете как построить и запрограммировать робота, который может ходить, ощущать окружающую среду, а также выполнять широкий спектр задач.

CONTENTS

Who This Book Is For
How This Book Is Organized
What Was Left Out
Code Style (About the Code)
Arduino Hardware and Software
Conventions Used in This Book
Using Code Examples
Safari® Books Online
How to Contact Us
Acknowledgments

Chapter 1. Introduction to Robot Building
Why Build a Robot?
How Robots Move
Tools

Chapter 2. Building the Electronics
Hardware Required
Construction Techniques
Soldering
Building the Motor Controller
Soldering the Reflectance Sensors
Making a Line Sensor Mount
Next Steps

Chapter 3. Building the Two-Wheeled Mobile Platform
Hardware Required
Mechanical Assembly
Lay Out the Chassis Parts
Motor Assembly
Assemble the Chassis Components
Attaching the Control Electronics
Mounting the IR sensors
Mounting the IR Sensors for Edge Detection
Mounting the IR Sensors for Line Following
Next Steps

Chapter 4. Building the Four-Wheeled Mobile Platform
Hardware Required
Mechanical Assembly
Lay Out the Chassis Parts
Motor Assembly
Assemble the Chassis Components
Solder the Power and Motor Connections
Connecting the Battery Pack and Power Switch
Building the Optional Trickle Charger
Assemble the Chassis
Mounting Arduino and Connecting Wires to the Shield
Mounting the IR sensors
Mounting the IR Sensors for Edge Detection
Mounting the IR Sensors for Line Following
Next Steps

Chapter 5. Tutorial: Getting Started with Arduino
Hardware Required
Arduino Software
Arduino Hardware
Installing the Integrated Development Environment (IDE)
Installing Arduino on Windows
Installing Arduino on OS X
Installing Arduino on Linux
Driver Installation
Connecting the Arduino Board
Using the IDE
Uploading and Running the Blink Sketch
Using Tabs
Installing Third-Party Libraries

Chapter 6. Testing the Robot's Basic Functions
Hardware Required
Software Prerequisites
Sketches Used in This Chapter
Load and Run helloRobot.ino
About the Sketch
Troubleshooting
Making the Sketch Easy to Enhance

Chapter 7. Controlling Speed and Direction
Hardware Required
Sketches Used in This Chapter
Types of Motors
Motor Controllers
Controlling Motor Speed
How Motor Speed Is Controlled
Code for Motor Control
Calibrating Rotation and Tracking
Software Architecture for Robot Mobility
Functions to Encapsulate Robot Movements
Core Movement Code
Additional Core Functions
Functions to Rotate the Robot
Higher-Level Movement Functions

Chapter 8. Tutorial: Introduction to Sensors
Hardware Discussed
Software
Infrared Reflectance Sensors
Sonar Distance Sensors
Maxbotix EZ1 Sonar Distance Sensor
Sharp IR Distance Sensor
Proximity Sensor
Sound Sensor
Arduino Cookbook

Chapter 9. Modifying the Robot to React to Edges and Lines
Hardware Required
Sketches Used in This Chapter
The Look Code
Edge Detection
Line Following
Seeing Sketch Data

Chapter 10. Autonomous Movement
Hardware Required
Sketches Used in This Chapter
Mounting a Ping Distance Sensor
Making a Mount for the Ping Sensor
Mounting the Ping Sensor in a Fixed Position
Mounting the Ping Sensor on a Servo
Letting the Robot Wander
Adding Scanning

Chapter 11. Remote Control
Hardware Required
Sketches Used in This Chapter
Design of the Remote Control Code
Controlling the Robot with a TV Type IR Remote
Installing the IR Decoder Chip
The IR Remote Software

Appendix A. Enhancing Your Robot
Planning
Think Before You Code
Avoid Feature Bloat
Don't Reinvent the Wheel
Structure to Reflect Functionality
Use Clear Names for Functions and Variables
Implementing a Complex Project
Test Often
Simplify
If It Is Awkward, Start Over
Don't Confuse Activity with Progress
Experiment
Be Tenacious
Have Fun

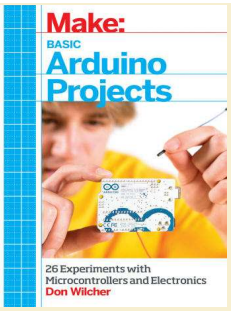
Appendix B. Using Other Hardware with Your Robot
Alternative Motor Controllers
Ardumoto
Continuous Rotation Servos

Appendix C. Debugging Your Robot
Identify the Symptoms and Localize the problem
Seeing What the Robot Is Doing

Appendix D. Power Sources
Monitoring Battery Voltage
Trickle Charging

Appendix E. Programming Constructs
Digital I/O
Analog I/O
Math functions
Other Functions and Constructs

Appendix F. Arduino Pin and Timer Usage
Handling Resource Conflicts
Modifying a Library to Change Timer Allocation
Writing Code That Avoids the Use of a Timer
Pin and Timer Tables



Don Wilcher

Basic Arduino Projects: 26 Experiments with Microcontrollers and Electronics

Простые проекты на Arduino:
26 экспериментов с микроконтроллерами и электроникой

— Publisher: Maker Media, Inc, 2014. — 260 pages

ISBN-10: 1449360661
ISBN-13: 978-1449360665

В этой книге представлено 26 четко описанных проектов, которые Вы можете собрать с Arduino - включая многокрасочные сигнальные огни, таймеры, инструменты для тестирования схем, звуковых эффектов, блока управления приводом и устройств на датчиках.

CONTENTS

Preface

1. The Trick Switch
Parts List
Let's Build a Trick Switch
Trick Switch with On/Off Indicators
Something to Think About

2. Sunrise-Sunset Light Switch
Parts List
Let's Build a Sunrise-Sunset Light Switch
Circuit Theory
Sunrise-Sunset Detector with Serial Monitor
Something to Think About

3. Tilt Sensing Servo Motor Controller
Parts List
Let's Build a Tilt Sensing Servo Motor Controller
Upload the Tilt Sensor Sketch
A Simple Animatronic Controller Using the Serial Monitor
Circuit Theory
Something to Think About

4. Twin LEDs
Parts List
Circuit Theory
Twin LED Flasher
Build the Adjustable Twin LED Flasher
It's Alive! Build a FrankenBot Toy
Something to Think About

5. The Opposite Switch
Parts List
Circuit Theory
The Opposite Switch (aka the NOT Logic Gate)
Build an Arduino NOT Logic Gate
Upload the Arduino NOT Logic Gate Sketch
Something to Think About

6. The AND Logic Gate
Parts List
Circuit Theory
The Arduino AND Logic Gate
Upload the Arduino AND Logic Gate Sketch
Something to Think About

7. The OR Logic Gate
Parts List
Circuit Theory
The Arduino OR Logic Gate
Upload the Arduino OR Logic Gate Sketch
Something to Think About

8. Tilt Flasher
Parts List
Circuit Theory
The Up-Down Sensor
Something to Think About

9. Multicolor RGB Flasher
Parts List
Circuit Theory
The RGB Flasher
Something to Think About

10. The Magic Light Bulb
Parts List
Let's Build a Magic Light Bulb
Upload the Magic Light Bulb Sketch
Circuit Theory
Something to Think About

11. Metal Checker: The Electronic Switch
Parts List
Let's Build a Metal Checker
Upload the Metal Checker Sketch
Circuit Theory
Something to Think About

12. The Theremin
Parts List
Let's Build a Theremin
Upload the Theremin Sketch
Circuit Theory
Something to Think About

13. An Arduino Ohmmeter
Parts List
Let's Build an Arduino Ohmmeter
Upload the Arduino Ohmmeter Sketch
Circuit Theory
Something to Think About

14. The LCD News Reader
Parts List
Let's Build the LCD
Upload the LCD News Reader Sketch
Circuit Theory
Something to Think About

15. A Logic Tester (with an RGB LED)
Parts List
Let's Build a Logic Tester
Upload the Logic Tester Sketch
Circuit Theory
Something to Think About

16. A Logic Tester (with an LCD)
Parts List
Let's Build a Logic Tester
Upload the Logic Tester Sketch
Circuit Theory
Something to Think About

17. The Amazing Pushbutton (with Processing)
Parts List
Let's Build an Amazing Pushbutton
Upload the Amazing Pushbutton Sketch

Download and Install Processing Notes
Let's Visualize Digital Data with Processing
Troubleshooting Tips for Processing
Something to Think About

18. The Terrific Tilt Switch (with Processing)
Parts List
Let's Build a Terrific Tilt Switch
Upload the Terrific Tilt Switch Sketch
Let's Visualize Digital Data with Processing
Something to Think About

19. The Rocket Launching Game (with Processing)
Parts List
Let's Build a Rocket Game
Upload the MultiDigital4 Sketch
The Rocket Launcher with Processing
Something to Think About

20. Temperature Indicator (with Processing)
Parts List
Let's Build a Temperature Indicator
Upload the Temperature Indicator Sketch
The Negative Temperature Coefficient (NTC) Sensor with Processing
Something to Think About

21. Sweeping Servo
Parts List
Let's Build a Servo Motor Tester
Upload the Sweeping Sketch
Something to Think About

22. Electronic Cricket
Parts List
Let's Build an Electronic Cricket
Upload the Electronic Cricket Sketch
Something to Think About

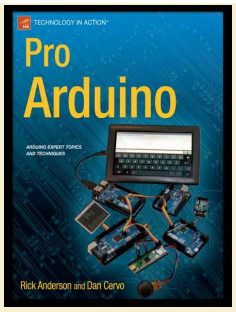
23. A Pocket Stage Light
Parts List
Let's Build a Pocket Stage Light
Upload the Pocket Stage Light Sketch
Something to Think About

24. Electronic Pixel
Parts List
Let's Build an Electronic Pixel
Upload the Electronic Pixel Sketch
Something to Think About

25. The Metronome
Parts List
Let's Build a Metronome
Upload the Metronome Sketch
Something to Think About

26. The Secret Word Game
Parts List
Let's Build a Secret Word Game
Upload the Secret Word Game Sketch
Rules for the Secret Word Game
Something to Think About

Index



Rick Anderson, Dan Cervo

Pro Arduino
Arduino профи

— Publisher: Apress, 2013. — 316 pages
— Series: Technology in Action

ISBN-10: 1430239395
ISBN-13: 978-1430239390

Итак, вы создали несколько проектов на Arduino, и теперь пришло время, чтобы шагнуть на ступеньку выше. Куда вы идёте дальше? С "Arduino Профи" Вы узнаете о новых инструментах, методах и механизмах, чтобы создавать еще более продвинутые, новаторские проекты.

CONTENTS

About the Authors
About the Technical Reviewer
Acknowledgments
Introduction

Chapter 1: Arduino 1.0.4 Core Changes
Changes to the Arduino IDE
Changes to Sketches
API Updates
pinMode
Return Types
uint_8
Arduino API Core 1.0.4
Arduino.h
Updated Serial Object
Updated Stream Class
Constructor
Member Functions
Print
New Printable Class
Updated String Library
Wire Library Updates
HardwareSerial Updates
Physical Board Updates and USB Compatibility
Avrdude Update
The New Arduino Leonardo Board
Board Variants
Variants Files
Variant Types and Naming
Uploader Options Renamed to Programmers
New Bootloaders
USB Firmware for 16u2
Summary

Chapter 2: Arduino Development and Social Coding
Components of Social Coding and Project Management
What Is a Project and How Is It Organized?
Overview of Version Control
Overview of Issue Tracking
Documentation
Project Management for Social Coding
Version Control with Git and GitHub
What Is Git ?
Installing Git
GitHub Tools
Version Control, Basic Workflow
Creating Your Own Project
Editing Code and Checking for Changes
Work process
View changes
Saving and committing changes
Workflow Summary: Creating Your Own Project
Workflow Summary: Forking Another Project
Creating a Pull Request
Creating a Pull Request
How To Merge a Pull Request
What is issue management ?
Issue management with Github
Connecting Version Control with Issue Management
Documentation
Github wiki
Creating Pages
Using Markdown
Code Blocks
Linking to files.
Headings
Lists
Ordered lists
Unordered lists
Linking to Images
Normal Text
Contributing to Arduino Development
Forking Your Own Copy of Arduino
How to build the Arduino IDE from source
Community Resources
Summary

Chapter 3: openFrameworks and Arduino
Getting Started
Arduino Code
Verifying the Code
Arduino Serial Functions
openFrameworks Setup
Connecting to the Arduino from openFrameworks
Verifying the Code
openFrameworks Serial Functions
Coding Once Using Firmata and ofArduino
Setting Up Firmata
Controlling the Arduino with openFrameworks
Verifying the Code
Key Constants Used by ofArduino
ofArduino Reference of Class Functions
Expanding on the Idea
Changing Code
Verifying the Code
More Ideas to Work With
Summary

Chapter 4: Android ADK
Android Devices
What to Check For
Known Working Devices
Modding
Arduino IDE Setup
Android Application Creation
The Arduino Sketch
The Android ADK Application
AndroidManifest.xml
res/xml/accessory_filter.xml
res/layout/main.xml
res/values/strings.xml
src/CH4.example.proArduino/CH4ExamplesActivity.java
Verifying the Code
Completing the Framework
Completing the Application
Arduino
Verifying the Code
SPI and ADK
Summary

Chapter 5: XBees
Buying XBees

Simple Setup
Transparent (AT Command) Mode
Module Configuration
Arduino Setup
Verifying the Code
API Mode
Module Configuration
API Packet Construction
Sending Commands
Sending Data
Request Packets
Reply Packets
Arduino Data Echo
Endpoint Firmware
Summary

Chapter 6: Simulating Sensors
Analog Sensors
Analog Sensor Reader
RC Low-Pass Filter
Verifying the Code
Resistor Ladder
Verifying the Code
Digital Sensors
PWM
Gray Code
Outputting Gray Code
Verifying the Code
Serial Sensors
Outputting Serial Data
Verifying the Code
I2C
The TWCR Register
The TWAR Register
The TWDR Register
The TWSR Register
Outputting I2C Data
Verifying the Code
Summary

Chapter 7: PID Controllers
The Mathematics
The Proportional Statement
The Integral Statement
The Derivative Statement
Adding It All Up
Time
PID Controller Setup
Wiring the Hardware
Verifying the Code
PID Tuner
Comparing PID, DEAD BAND, and ON/OFF Controllers
PID Can Control
Tuning
PID Library
PID Library Functions
Other Resources
Summary

Chapter 8: Android Sensor Networks
Setting Up a Sensor Network
openFrameworks
The Arduino
The Android Application
Summary

Chapter 9: Using Arduino with PIC32 and ATtiny Atmel Chips
Arduino and Nonstandard Environments
The MPIDE and chipKIT PIC32
Example: Object Detection using the Task Manager service
Arduino Support for the ATtiny Family
ATtiny 85/45/25
ATtiny 84/44/24
ATtiny 4313 and 2313
Using the Arduino as an ISP Programmer
Project: Secret Knock Box
What the Device Does
Bill of Materials
Summary

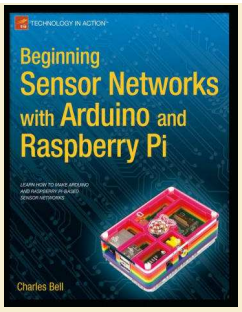
Chapter 10: Multiprocessing: Linking the Arduino for More Power
I2C
Serial Peripheral Interface
Connecting Two Devices
Setting Up a Master SPI Device
Verifying the Code
Interrupting Vectors
SPI by the Registers
Verifying the Code
Multiple Slaves
Master in Register
Verifying the Code
Symmetric Architecture Bipolar Bus
SABB by the Code
Verifying the Code
Connecting SABB to SPI
Conversion to Mega
Physical Best Practices
Summary

Chapter 11: Game Development with Arduino
Games Suitable for the Arduino
A Simple Game
Proof of Concept
Coding Stop It
Verifying the Code
Dirty Little Tricks,
Adding Better Displays and Graphics
Gameduino Library
A New Stop It
Art
Coding Stack It
Verifying the Code
Making Sounds
Adding a Bit of Splash
Programming the Game to Play Itself
The Finishing Polish
Arcade and Game Resources
Summary

Chapter 12: Writing Your Own Arduino Libraries
What you need to know to write your own libraries
Creating a simple library
Making a Motor Library
The anatomy of an Arduino library folder
Examples Folder
License
keywords.txt
Installing Arduino Libraries
Using Arduino Libraries
Arduino Objects and Library Conventions
Summary

Chapter 13: Arduino Test Suite
Installing the Arduino Test Suite
Getting Started with Testing
Arduino Test Result Format
Test Result Section Format Details
Test-Naming Structure
Test Status Options
Test Summary
Arduino Test Suite Basic Functions
ATS_begin
ATS_PrintTestStatus
ATS_end
Using the Basic Functions
Arduino Test Suite Built-In Tests
Strategies for Testing Your Own Arduino Derivative
Memory Testing
Example: Testing for a Memory Leak
Testing Libraries
SPI.transfer() Test
setBitOrder() Test
setClockDivider() Test
setDataMode() Test
SPI Test Results

Summary



Charles Bell

Beginning Sensor Networks with Arduino and Raspberry Pi

Введение в сенсорные сети с Arduino и Raspberry Pi

— Publisher: Apress, 2013. — 372 pages
— Series: Technology in Action

ISBN-10: 1430258241
ISBN-13: 978-1430258247

Эта книга научит вас, как создавать сенсорные сети с Arduino, Raspberry Pi и радиомодулями XBee, а также покажет, как превратить Ваш Raspberry Pi в сервер баз данных MySQL, чтобы хранить данные принятые с датчиков.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments

Introduction

Chapter 1: Introduction to Sensor Networks
Anatomy of a Sensor Network
Examples of Sensor Networks
The Topology of a Sensor Network
Communication Media
Wired Networks
Wireless Networks
Hybrid Networks
Types of Sensor Nodes
Basic Sensor Nodes
Data Nodes
Aggregator Nodes
Sensors
How Sensors Measure
Storing Sensor Data
Examples of Sensors

Chapter 2: Tiny Talking Modules: An Introduction to XBee
Wireless Modules
What Is an XBee?
XBee Primer
Choosing XBee Modules
Interacting with an XBee-ZB Module
Pin Layout
Configuring Modules
For More Information
An XBee Wireless Chat Room
Loading the Firmware for the Modules
Capturing Serial Numbers
Configuring the Coordinator
Configuring the Router
Let the Chat Begin
For More Fun
Building an XBee-ZB Mesh Network
Loading the Firmware for the Modules
Configuring the XBee Modules
Forming Test Messages
Testing the Network
For More Fun
Component Shopping List
Troubleshooting Tips and Common Issues
Things to Check
Common Issues

Chapter 3: Arduino-Based Sensor Nodes
What Is an Arduino?
Arduino Models
Arduino Clones
So, Which Do I Buy?
Where to Buy
Arduino Tutorial
Learning Resources
The Arduino IDE
Project: Hardware "Hello, World!"
Hosting Sensors with Arduino
Project: Building an Arduino Temperature Sensor
Hardware Setup
Software Setup
Writing the Sketch
Test Execution
Project: Using an Arduino as a Data Collector for XBee Sensor Nodes
XBee Sensor Node
Arduino with XBee Shield
Testing the Final Project
For More Fun
Component Shopping List

Chapter 4: Raspberry Pi-based Sensor Nodes
What Is a Raspberry Pi?
Noble Origins
Models
A Tour of the Board
Required Accessories
Recommended Accessories
Where to Buy
Raspberry Pi Tutorial
Getting Started
Installing a Boot Image
Booting Up
GPIO Pin Mapping
Required Software
Project: Hardware "Hello, World!"
Hosting Sensors with Raspberry Pi
Project: Building a Raspberry Temperature Sensor Node
Hardware Setup
Testing the Hardware
Software Setup
Testing the Sensor
For More Fun
Project: Building a Raspberry Barometric Pressure Sensor Node
Hardware Setup
Testing the Hardware
Software Setup
Testing the Sensor
For More Fun
Project: Creating a Raspberry Pi Data Collector for XBee Sensor Nodes
XBee Sensor Node
Hardware
Software
Testing the Final Project
For More Fun
Component Shopping List

Chapter 5: Where to Put It All: Storing Sensor Data
Storage Methods
Local Storage Options for the Arduino
Nonvolatile Memory
SD Card
Project: Saving Data in Nonvolatile Memory
Project: Writing Data to an SD Card
Local Storage Options for the Raspberry Pi
Project: Writing Data to Files
Remote Storage Options
Storing Data in the Cloud

Storing Sensor Data in a Database
Component Shopping List

Chapter 6: Turning Your Raspberry Pi into a Database Server
What Is MySQL?
Getting Started with MySQL
How and Where MySQL Stores Data
The MySQL Configuration File
How to Start, Stop, and Restart MySQL
Creating Users and Granting Access
MySQL and Python—MySQL Utilities
Building a Raspberry Pi MySQL Server
Partitioning and Formatting the Drive
Setting Up Automatic Drive Mounting
Project: Installing MySQL Server on a Raspberry Pi
Advanced Project: Using MySQL Replication to Back Up Your Sensor Data
Component Shopping List

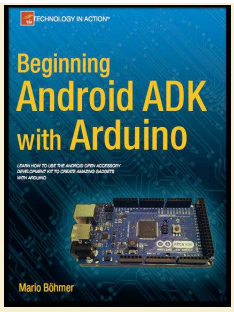
Chapter 7: MySQL and Arduino: United at Last!
Introducing Connector/Arduino
Hardware Requirements
What About Memory?
How to Get MySQL Connector/Arduino
Limitations
Building Connector/Arduino-Enabled Sketches
Database Setup
Setting Up the Arduino
Starting a New Sketch
Testing the Sketch
Troubleshooting Connector/Arduino
MySQL Server Configuration
MySQL User Account Problems
Networking Configuration
Connector Installation
Other
None of These Solved My Problem—What Next?
A Tour of the MySQL Connector/Arduino Code
Library Files
Field Structure
Public Methods
Example Uses
Project: Building a MySQL Arduino Client
Hardware Setup
Software Setup
Setting Up the Sensor Database
Writing the Code
Test Execution
For More Fun
Project Example: Inserting Data from Variables
Project Example: How to Perform SELECT Queries
Displaying a Result Set in the Serial Monitor
Writing Your Own Display Method
Example: Getting a Lookup Value from the Database
Component Shopping List

Chapter 8: Building Your Network: Arduino Wireless Aggregator + Wireless
Sensor Node + Raspberry Pi Server
Data-Aggregate Nodes
Local-Storage Data Aggregator
Project: Data-Aggregate Node with Local Storage
Remote-Storage Data Aggregator
Project: Arduino Data-Aggregate Node with Database Storage
Project: Raspberry Pi Data-Aggregate Node with Database Storage
Component Shopping List

Chapter 9: Planning Wireless Sensor Networks
Sensor Networks Best Practices
Considerations for Data-Aggregate Nodes
Considerations for Sensor Network Databases
Other Considerations
Choosing Sensor Nodes
Wired Or Wireless?
Arduino or Raspberry Pi?
Alternative Hosts
Project: Home Temperature-Monitoring Network
Planning Considerations
Planning the Nodes
Cost Considerations
What About Implementation?
Conclusion
For More Fun
Optional Component Shopping List

Appendix: Shopping List

Index



Mario Bohmer

Beginning Android ADK with Arduino
Введение в Android ADK с Arduino

— Publisher: Apress, 2012. — 316 pages
— Series: Technology in Action

ISBN-10: 1430241977
ISBN-13: 978-1430241973

Если Вы новичок в Arduino и разработчик приложений для Android, эта книга для Вас. Теперь, когда Google выпустила комплект для Android (Android ADK), сопряжение Android с Arduino для создания собственных гаджетов стало еще проще.

Вы изучите: как различные платы работают с ADK; как создать ваш первый проект; как работать со светом и звуком; как работать с сервомоторами и электродвигателями постоянного тока; как работать с фоторезисторами и терморезисторами; и многое другое.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments
Preface

Chapter 1: Introduction
What Is the ADK?
Hardware Development Boards
The Google ADK
The Arduino ADK
The IOIO
The Seeeduino ADK Main Board
More ADK Possibilities
Which Board Should You Use?
Supported Android Devices
Setting Up the Development Environment
The Java Development Kit
The Android SDK
The Eclipse IDE
The Arduino IDE
Installing Hardware Drivers
The ADK Reference Package
Fritzing (Optional Software)
Ready, Set, Go

Chapter 2: Android and Arduino: Getting to Know Each Other
Hello Arduino
Hello Android
Getting To Know Each Other
Extending Hello World for Arduino
Extending Hello World for Android
Let's Talk
Processing Commands for Arduino
Processing Commands for Android

Chapter 3: Outputs
Project 1: Toggle an LED
The Parts
The Setup
The Software
Project 2: Dim an LED
The Parts
The Setup
The Software

Chapter 4: Inputs
Project 3: Reading the State of a Button
The Parts
The Setup
The Software
Project 4: Adjusting Analog Input with a Potentiometer
The Parts
The Setup
The Software

Chapter 5: Sounds.
Project 5: Generating Sound with a Piezo Buzzer.
The Parts
The Setup
The Software
Project 6: Sensing Sound with a Piezo Buzzer
The Parts
The Setup
The Software

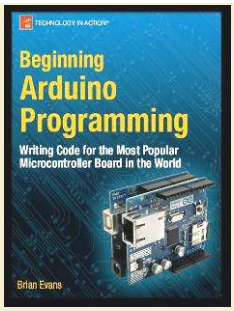
Chapter 6: Light Intensity Sensing.
Project 7: Sensing Light Intensity with a Photoresistor
The Parts
The Setup
The Software
Bonus: Measure Illuminance in Lux with Android

Chapter 7: Temperature Sensing
Project 8: Sensing the Temperature with a Thermistor
The Parts
The Setup
The Software
n Chapter 8: A Sense of Touch
Project 9: DIY Capacitive Touch Game Show Buzzer
The Parts
The Setup
The Software
Bonus Practical Example: The ADK Paper Piano

Chapter 9: Making Things Move
Project 10: Controlling Servos
The Parts
The Setup
The Software
Project 11: Controlling DC Motors
The Parts
The Setup
The Software

Chapter 10: Alarm System
Project 12: SMS Alarm System with Tilt-Switch
The Parts
The Setup
The Software
Project 13: Camera Alarm System with IR Light Barrier
The Parts
The Setup
The Software

Index



Brian Evans

Beginning Arduino Programming: Writing Code for the Most Popular Microcontroller Board in the World

Введение в программирование Arduino: пишем код для самых популярных микроконтроллерных плат в мире

— Publisher: Apress, 2011. — 272 pages
— Series: Technology in Action

ISBN-10: 1430237775
ISBN-13: 978-1430237778

Книга позволит быстро развить свои навыки программирования, поможет научиться основам разработки кода для Arduino, в том числе, написания функций и библиотек, необходимых для создания будущих проектов.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments
Introduction
Who This Book is For
Reading This Book
Downloading the Code

Getting Started
Arduino is for Makers
The Arduino Ecosystem
The Arduino Platform
Open-Source Hardware
Community
Arduinoland
Arduino is C… Mostly
What's Needed
Getting Up and Running
Installing the Software
Connecting the Arduino
Opening a Sketch
Selecting the Board and Serial Port
Uploading a Sketch
Summary

Sketching in Code
What is Sketching in Code?
Project 1: RGB Blink
Hooking It Up
Uploading the Source Code
Source Code Summary
The Structure of Arduino C
Using Comments
Basic Functions
Statements and Syntax
Verifying and Uploading
Verifying
Saving
Uploading
Common Errors
Summary

Working with Variables
Project 2: 7-Color Blink
Hooking It Up
Uploading the Source Code
Source Code Summary
What's a Variable?
Declaring Variables
Variable Names
Data Types
Variable Qualifiers
Predefined Constants
Variable Scope
Using Operators
Arithmetic Operators: +, -, *, /
Compound Operators: ++, --, +=, -=, *=, /=
Order of Operations

Making Decisions
Project 3: Tilt Blink
Hooking It Up
Uploading the Source Code
Source Code Summary
Comparative and Logical Operators
Control Statements
If
For
While
Do
Switch
Break
Continue
Summary

Digital Ins and Outs
Arduino I/O Demystified
Project 4: Noisy Cricket
Hooking It Up
Uploading the Source Code
Source Code Summary
Digital Functions
pinMode()
digitalWrite()
digitalRead()
State Changes
Toggle
Counting
Modality
Summary

Analog In, Analog Out
Analog Demystified
Project 5: Telematic Breath
Hooking It Up
Uploading the Source Code
Source Code Summary
Analog Functions
analogRead()
analogWrite()
analogReference()
Analog Serial Monitor
Reading Analog Values
Using the Serial Monitor
How It Works
Mapping Values
map()
constrain()
Summary

Advanced Functions
Timing Functions
delay()
delayMicroseconds()
millis()
micros()
Random Functions

random()
randomSeed()
Project 6: Ambient Temps
Hooking It Up
Uploading the Source Code
Source Code Summary
Writing Functions
Declaring Functions
Calling Functions
Function Returns
Function Parameters
Project 7: HSB Color Mixer
Hooking It Up
Uploading the Source Code
Source Code Summary
Hardware Interrupts
attachInterrupt()
detachInterrupt()
Summary

Arrays and Memory
Project 8: Decision Machine
Hooking It Up
Uploading the Source Code
Source Code Summary
Arrays
Declaring Arrays
Using Arrays
Character Arrays
Multidimensional Arrays
Arduino Memory
Checking Free RAM
Using Program Memory
Using EEPROM
Summary

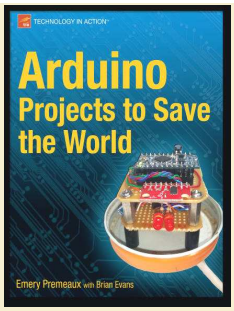
Hardware Libraries
Using Libraries
Creating an Instance
Initializing the Library
LiquidCrystal library
Example Code: Arduino Haiku
LiquidCrystal()
begin()
print()
clear()
setCursor()
Example Code: Symbols and Characters()
write()
createChar()
Example Code: Fish Tank Animation
scrollDisplayLeft() and scrollDisplayRight()
Servo Library
Example Code: Reminder Bell
Servo
attach()
write()
Stepper Library
Example Code: 60-Second Sweep
Stepper
setSpeed()
step()
SD Library
Example Code: SD Logger
File
SD.begin()
SD.open()
close()
write()
print()
Example Code: SD Flicker
available()
read()
Summary

Serial and I2C
Using Hardware Serial
Project 9: Serial to Servo
Hooking It Up
Uploading the Source Code
Source Code Summary
Serial Library
begin()
available()
read()
print()
println()
write()
Project 10: RFID Reader
Hooking It Up
Uploading the Source Code
Source Code Summary
SoftwareSerial Library
SoftwareSerial()
begin()
flush()
strncmp()
Project 11: Serial Time Clock
Hooking It Up
Uploading the Source Code
Source Code Summary
Wire Library
begin()
beginTransmission()
endTransmission()
write()
requestFrom()
read()
Summary

Continuing On
Build More Projects
Bonus Project 1: Make Something Tweet
Bonus Project 2: Make Something Move
Bonus Project 3: Mega-Size Something
Learn Another Language
Firmata
Processing
PureData
Contribute to the Community
Participate in Online Forums
Publish Your Project
Summary

Beginning Electronics
Basic Electronics
Circuits
Electricity
Common Components
Resistors
Capacitors
Diodes
Transistors
Switches
Motors
Reading Schematics
Prototyping
Breadboards
Soldering
Summary

Resources
Additional Resources
Forums
Tutorials
Other Stuff
Selected Suppliers
Parts Used in This Book



Brian Evans, Emery Premeaux

Arduino Projects to Save the World

Проекты Arduino для сохранения окружающей среды

— Publisher: Apress, 2011. — 256 pages

ISBN-10: 143023623X
ISBN-13: 978-1430236238

Эта книга покажет как используя плату Arduino, несколько проводов, датчиков и немного сообразительности можно создать устройства, которые понижают затраты электроэнергии, помогают выращивать собственную еду на грядке, следят за загрязнением воздуха и даже предупреждают о подземных толчках.

CONTENTS

About the Authors
About the Technical Reviewer
Acknowledgments
Preface

Chapter 1: Saving the World…
It's All About Sensors
Arduino's Analog to Digital Converter (ADC)
Conversion Process
Changing the Voltage Reference
Voltage Dividers
A Strategy for Prototyping Sensor Systems
Understand the Sensor
Figure Out the Equations
Write a Sample Serial Sketch
Put the Sensor Through its Paces
Integrate the Code into the Project by Building Sensor Functions
Consider Power Saving Whenever Possible
Supplies and Tools Needed
Building the BreadboardShield
Summary

Chapter 2: Spider Temps
The Hardware
Parts List
Optional
Building It
Mechanical Build
Determining Temperature Equations
Test Code
Basic SpiderTemps Code
Test It Out
SpiderTemps, Take Two: Calibration
Adding a Display
Battery Powered?
Boxing It Up
Making Mods
Conclusion

Chapter 3: Jungle Power
Diverse Power Sources
Solar Power
Wind Power
Water Power
Energy Harvesting
Three Sides to Every (Power) Story
Input (Batteries, Charge Controllers, and Free Energy)
Regulation Options
Selecting Your Sensor Node Arduino
Tips to Optimize Your System for Longer Battery Life
Lower the Operating Voltage
Using a Zener Diode to Drive Aref
Putting the Arduino to Sleep
Get Rid of LEDs Wherever Possible
The Build
Parts
Build Process
Measuring Current Draw
Determining Operational Duty Cycle (Arduino Takes a Nap)
More about Batteries
Choosing Solar Panels
Assembly
Prepare the Solar Panels
Wire the Power Supply Subsytems
Mount the Battery Case and Connect the Supplies
Install the Sensors
Build the Clock Circuit
Software
Libraries
Code
Testing It Out
Put It in a Case
Resources

Chapter 4: Telesensation
Getting the Lay of the Land
Planning the Message Flow and Hardware
A Simple Network
A Complex Network
A Look at Available Radio Options
Serial AM/FM Radio
Bluetooth UART/Serial Modems
Zigbee and Xbee
Freakduino
Antenna Considerations
Building the Two-Node Sensor Network
Freakduino-Chibi Version Hardware
Xbee version Hardware (Stalker version 2.0)
Processing umm… Processes!
Adding Logging Features
Reading Data into a Spreadsheet
Conclusion
Resources

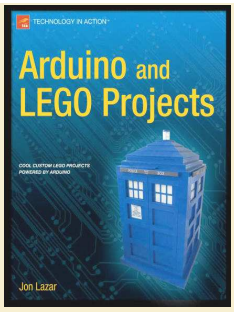
Chapter 5: Contributing to the Hive Mind
Introduction to Online Data Aggregation Services
Pachube.com
Google.com/powermeter
Sensorpedia.com
Open.Sen.se
Taking a Closer Look at Pachube
Terminology and Account Limitations
Pachube, Meet Arduino
Getting Started
Install Processing and Various Libraries.
The Build
Setting Up a Feed on Pachube
Arduino Sketch: Firmata
PC Side: Processing and EEML
Putting it All Together .
Going Further
Conclusion .
References

Chapter 6: The Mass Effect
The Data .
The Project .
Seismograph Technology .
How MEMS Accelerometers Work
The Build .
Building the Prototype Hardware
Software

Arduino Code
Processing
Building the Final Transducer System
Transducer Considerations
Cabling Considerations
Parts List
Getting Prepared
A System of Three Systems .
Testing and Assembly
Installation .
Going Further
References

Chapter 7: Staying Current
What Is Alternating Current?
Energy Monitor Hardware
Split-Core Current Transformer
AC/AC Adapter
Liquid Crystal Display
Three-Wire Extension Cord
Parts List
Building it
Current and Voltage Sensing
Connecting the Display
Building the Cable
Finishing it Up
Energy Monitor Code
Constants
setup() and loop()
calculatePower()
displayPower()
Calibrating
Current Settings
Voltage Settings
Entering the Constants
Wrapping it Up
Going Further
Energy Monitor SD Code
Code Summary
Further Still?
Conclusion
References

Index



Jon Lazar

Arduino and LEGO Projects
Проекты Arduino и LEGO

— Publisher: Apress, 2013. — 212 pages

ISBN-10: 1430249293
ISBN-13: 978-1430249290

Все мы знаем как удивителен LEGO. И все больше людей обнаруживает сколько удивительных вещей можно сделать с Arduino. В этой книге показывается как объединить Arduino и LEGO, чтобы собрать забавные гаджеты.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments

Chapter 1: LEGO, Arduino, and The Ultimate Machine
Introducing the Arduino
Your First Arduino Program
Programming the Ultimate Machine
Assembling the Arduino and Motor
Programming the Arduino
Building the Ultimate Machine
Selecting the Dimensions
Building the Brick Walls
Adding The Arduino
Adding LEGO Arms and a Switch
Raising the Walls
Building the Lid

Chapter 2: Using Sensors with the Android
The Ultrasound Sensor
Adding Additional Sensors
Building the Android
Start with the Foundation
Building a Harness for the Arduino
Adding a Level for the Power Plug
Building the Body
Adding Arms and Sensors
Separating the Body from the Head
Building the Head
Creating the Legs
Building the Arms
Building the Antenna

Chapter 3: Twitter PetI
Connecting the Arduino to the Internet
Building The Twitter Pet
Building the Base
Setting the Arduino in Place
Covering the Arduino
Adding Rings
Adding the Eyes and Nose
Adding the Head

Chapter 4: RFID and the Crystal Ball.i
Arduino and RFID
Generating Magic with Code
Building the Crystal Ball
Building the Base
Building the Lid
Building the Sphere
Building the Magic Wand

Chapter 5: Animating the TARDIS
Doctoring the TARDIS
Coding the Wave Shield
The Chameleon Circuit: Building the TARDIS
Building the Walls
Securing the Walls
Building the Windows
Installing the Windows
Adding the Arduino
Back to the Body
Creating Labels
Building the Roof

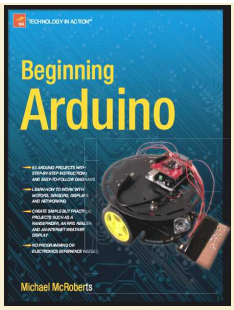
Chapter 6: Controlling LEGO Trains with Arduino
Arduino Train Controls
Programming the Train Controls
Building the Train Station
Building a LEGO Train
Adding Wheels
Adding the Battery Pack
Adding the IR Receiver

Chapter 7: Building a Light-Sensitive Box
The Box's Mechanics
Programming the Box
Building the Box
Adding the Arduino
Adding the Motor
Adjusting the Wall Height
Adding Hinges
Adding a Lid

Appendix A: Parts List

Chapter 1: LEGO, Arduino, and the Ultimate Machine
Chapter 2: Using Sensors with the Android
Chapter 3: Twitter Pet
Chapter 4: RFID and the Crystal Ball
Chapter 5: Animating the TARDIS
Chapter 6: Controlling LEGO Trains With Arduino
Electronics
LEGO for Train Station
LEGO for Train
Chapter 7: Building a Light Sensitive Box
Electronics
LEGO

Index



Michael McRoberts

Beginning Arduino

Arduino для начинающих

— Publisher: Apress, — Series: Technology in Action
— 1 edition 2010 — 472 p. ISBN-10: 1430232404 ISBN-13: 978-1430232407
— 2 edition 2013 — 424 p. ISBN-10: 143025016X ISBN-13: 978-1430250166

В книге на примере 50-и интересных проектов вы узнаете всё о популярной платформе Arduino.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments
Introduction
Introduction
How to Use This Book
What You Will Need
What Exactly is an Arduino?
Getting Started
Windows XP Installation
Windows 7 & Vista Installation
Mac OSX Installation
Board and Port Selection
Upload Your First Sketch
The Arduino IDE

Light 'Em Up
Project 1 – LED Flasher
Parts Required
Connecting Everything
Enter the Code
Project 1 – LED Flasher – Code Overview
Project 1 – LED Flasher – Hardware Overview
Project 2 – S.O.S. Morse Code Signaler
Project 2 – S.O.S. Morse Code Signaler – Code Overview
Project 3 – Traffic Lights
Parts Required
Connect It Up
Enter the Code
Project 4 – Interactive Traffic Lights
Parts Required
Connect It Up
Enter the Code
Project 4 – Code Overview
Project 4 – Interactive Traffic Lights Hardware Overview
Logic States
Pull-Down Resistors
Pull-Up Resistors
The Arduino's Internal Pull-Up Resistors
Summary

LED Effects
Project 5 – LED Chase Effect
Parts Required
Connect It Up
Enter the Code
Project 5 – LED Chase Effect – Code Overview
Project 6 – Interactive LED Chase Effect
Parts Required
Connect It Up
Enter The Code
Project 6 – Interactive LED Chase Effect – Code Overview
Project 6 – Interactive LED Chase Effect – Hardware Overview
Project 7 – Pulsating Lamp
Parts Required
Connect It Up
Enter the Code
Project 7 – Pulsating Lamp – Code Overview
Project 8 – RGB Mood Lamp
Parts Required
Connect It Up
Enter the Code
Project 8 – RGB Mood Lamp – Code Overview
Project 9 – LED Fire Effect
Parts Required
Connect It Up
Enter the Code
Project 9 – LED Fire Effect – Code Overview
Project 10 – Serial Controlled Mood Lamp
Enter the Code
Project 10 – Serial Controlled Mood Lamp – Code Overview
Summary

Simple Sounders and Sensors
Project 11 – Piezo Sounder Alarm
Parts Required
Connect It Up
Enter the Code
Project 11 – Piezo Sounder Alarm – Code Overview
Project 11 – Piezo Sounder Alarm – Hardware Overview
Project 12 – Piezo Sounder Melody Player
Enter the Code
Project 12 – Piezo Sounder Melody Player – Code Overview
Project 13 – Piezo Knock Sensor
Parts Required
Connect It Up
Enter the Code
Project 13 – Piezo Knock Sensor – Code Overview
Project 14 – Light Sensor
Parts Required
Connect It Up
Enter the Code
Project 14 – Light Sensor – Hardware Overview
Summary
Subjects and Concepts covered in Chapter 4:

Driving a DC Motor
Project 15 – Simple Motor Control
Parts Required
Connect It Up
Enter The Code
Project 15 – Simple Motor Control – Code Overview
Project 15 – Simple Motor Control – Hardware Overview
Transistors
Motors
Diodes
Project 16 – Using an L293D Motor Driver IC
Parts Required
Connect It Up
Enter the Code
Project 16 – Using an L293D Motor Driver IC – Code Overview
Project 16 – Using an L293D Motor Driver IC – Hardware Overview
Summary
Subjects and concepts covered in Chapter 5

Binary Counters
Project 17 – Shift Register 8-Bit Binary Counter
Parts Required
Connect It Up
Enter The Code
The Binary Number System
Project 17 – Shift Register 8-Bit Binary Counter Hardware Overview
Project 17 – Shift Register 8-Bit Binary Counter – Code Overview
Bitwise Operators
Bitwise AND (&)
Bitwise OR (|)
Bitwise XOR (^)
Bitwise NOT (~)
Bitshift Left (<<), Bitshift Right (>>)
Project 17 – Code Overview (continued)

Project 18 – Dual 8-Bit Binary Counters
Parts Required
Connect It Up
Enter the Code
Project 18 Code & Hardware Overview
Summary
Subjects and Concepts covered in Chapter 6

LED Displays
Project 19 – LED Dot Matrix Display – Basic Animation
Parts Required
Connect It Up
Enter the Code
Project 19 – LED Dot Matrix – Basic Animation – Hardware Overview
Multiplexing
Project 19 – LED Dot Matrix – Basic Animation – Code Overview
Project 20 – LED Dot Matrix Display – Scrolling Sprite
Enter the Code
Project 20 – LED Dot Matrix – Scrolling Sprite – Code Overview
Project 21 – LED Dot Matrix Display – Scrolling Message
Parts Required
Connect It Up
Enter the Code
Project 21 – LED Dot Matrix – Scrolling Message – Hardware Overview
Project 21 – LED Dot Matrix – Scrolling Message – Code Overview
Project 22 – LED Dot Matrix Display – Pong Game
Parts Required
Connect It Up
Upload the Code
Project 22 – LED Dot Matrix – Pong Game
Summary
Subjects and concepts covered in Chapter 7:

Liquid Crystal Displays
Project 23 – Basic LCD Control
Parts Required
Connect It Up
Enter The Code
Project 23 – Basic LCD Control – Code Overview
Project 23 – Basic LCD Control – Hardware Overview
Project 24 – LCD Temperature Display
Parts Required
Connect It Up
Enter The Code
Project 24 – LCD Temperature Display – Code Overview
Summary
Subjects and Concepts Covered in Chapter 8

Servos
Project 25 – Servo Control
Parts Required
Connect It Up
Enter The Code
Project 25 – Servo Control – Code Overview
Project 25 – Servo Control – Hardware Overview
Project 26 – Dual Servo Control
Parts Required
Connect It Up
Enter The Code
Project 26 – Dual Servo Control – Code Overview
Project 27 – Joystick Servo Control
Parts Required
Connect It Up
Enter The Code
Project 27 – Joystick Servo Control – Code Overview
Summary
Subjects and Concepts Covered in Chapter 9

Steppers and Robots
Project 28 – Basic Stepper Control
Parts Required
Connect It Up
Enter the Code
Project 28 – Basic Stepper Control – Code Overview
Project 28 – Basic Stepper Control – Hardware Overview
Project 29 – Using a Motor Shield
Parts Required
Connect It Up
Enter the Code
Project 29 – Using a Motor Shield – Code Overview
Project 29 – Using a Motor Shield – Hardware Overview
Project 30 – Line Following Robot
Parts Required
Connect It Up
Enter the Code
Project 30 – Line Following Robot – Code Overview
Summary
Subjects and Concepts covered in Chapter 10

Pressure Sensors
Project 31 – Digital Pressure Sensor
Parts Required
Connect It Up
Enter the Code
Project 31 – Digital Pressure Sensor – Code Overview
SPI – Serial Peripherals Interface
Project 31 – Digital Pressure Sensor – Code Overview (cont.)
Project 32 – Digital Barograph
Parts Required
Connect It Up
Enter the Code
Project 32 – Digital Barograph – Code Overview
Summary
Subjects and Concepts covered in Chapter 11

Touch Screens
Project 33 – Basic Touch Screen
Parts Required
Connect It Up
Enter the Code
Project 33 – Basic Touch Screen – Hardware Overview
Project 33 – Basic Touch Screen – Code Overview
Project 34 – Touch Screen Keypad
Parts Required
Connect It Up
Enter the Code
Project 34 –Touch Screen Keypad – Code Overview
Project 35 – Touch Screen Light Controller
Parts Required
Connect It Up
Enter the Code
Project 35 – Touch Screen Controller – Code Overview
Summary
Subjects and Concepts covered in Chapter 12

Temperature Sensors
Project 36 – Serial Temperature Sensor
Parts Required
Connect It Up
Enter the Code
Project 36 – Serial Temperature Sensor – Code Overview
Project 37 – 1-Wire Digital Temperature Sensor
Parts Required
Connect It Up
Enter the Code
Project 37 – 1-Wire Digital Temperature Sensor – Code Overview
Summary
Subjects and Concepts covered in Chapter 13

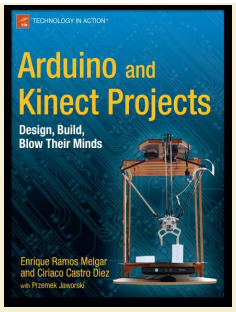
Ultrasonic Rangefinders
Project 38 – Simple Ultrasonic Rangefinder
Parts Required
Connect It Up
Enter the Code
Project 38 – Simple Ultrasonic Range Finder – Code Overview
Project 38 – Simple Ultrasonic Range Finder – Hardware Overview
Project 39 – Ultrasonic Distance Display
Parts Required
Connect It Up
Enter the Code
Project 39 – Ultrasonic Distance Display – Code Overview
Project 40 – Ultrasonic Alarm
Parts Required
Connect It Up
Enter the Code
Project 40 – Ultrasonic Alarm – Code Overview
Project 41 – Ultrasonic Theremin
Enter the Code
Project 41 – Ultrasonic Theremin – Code Overview
Summary
Subjects and Concepts covered in Chapter 14

Reading and Writing to an SD Card
Project 42 – Simple SD Card/Read Write
Parts Required
Connect It Up
Enter the Code
Project 42 – Simple SD Card Read/Write – Code Overview
Project 43 – Temperature SD Datalogger
Parts Required
Connect It Up

Enter the Code
Project 43 – Temperature SD Datalogger – Code Overview
Project 43 – Temperature SD Datalogger – Hardware Overview
Summary
Subjects and Concepts covered in Chapter 15

Making an RFID Reader
Project 44 – Simple RFID Reader
Parts Required
Connect It Up
Enter the Code
Project 44 – Simple RFID Reader – Hardware Overview
Project 45 – Access Control System
Parts Required
Connect It Up
Enter the Code
Project 45 – Access Control System – Code Overview
Summary
Subjects and Concepts covered in Chapter 16

Communicating over Ethernet
Project 46 – Ethernet Shield
Parts Required
Connect It Up
Enter the Code
Project 46 – Ethernet Shield – Code Overview
Project 47 – Internet Weather Display
Enter the Code
Project 47 – Internet Weather Display – Code Overview
Project 48 – Email Alert System
Enter the Code
Project 48 – Email Alert System – Code Overview
Project 49 – Twitterbot
Enter the Code
Project 49 – Twitterbot – Code Overview
Project 50 – RSS Weather Reader
Enter the Code
Project 50 – RSS Weather Reader – Code Overview
Summary
Subjects and Concepts covered in Chapter 17



Enrique Ramos Melgar, Ciriaco Castro Diez

Arduino and Kinect Projects: Design, Build, Blow Their Minds

Проекты Arduino и Kinect: замысел, сборка, программирование

— Publisher: Apress, 2012. — 416 pages
— Series: Technology in Action

ISBN-10: 1430241675
ISBN-13: 978-1430241676

В книге собрано десять увлекательных проектов для связки Arduino и Kinect.

CONTENTS

About the Authors
About the Technical Reviewer
Acknowledgments
Introduction
The Structure of This Book
The Content of the Chapters

Arduino Basics
What is Arduino?
A Brief History of Arduino
Installing Arduino
Installation on Mac OS X
Installation on Windows
Installation on Linux
Testing the Arduino
Arduino Hardware
Arduino Input and Output Pins
Pull-Up Resistors
Arduino Shields
Arduino IDE
Serial Monitor
Arduino Language
The setup() Function
The loop() Function
Variables
Your First Arduino Project
Breadboard
Building the Circuit
Programming the Arduino
Circuit Diagrams
Fritzing
Electronic Symbols
Electricity
AC/DC
Ohms Law
Joule's Law
Summary

Kinect Basics
A Brief History of the Kinect
Hacking the Kinect
Official Frameworks
The Kinect Sensor
Positioning your Kinect
Kinect Capabilities
RGB Image
IR Image
Depth Map
Hand and Skeleton Tracking
Kinect Drivers and Frameworks
OpenKinect: Libfreenect Drivers
PrimeSense: OpenNI and NITE
Microsoft Kinect for Windows
Kinect Theory
Structured-Light 3D Scanning
Converting the Light Coding Image to a Depth Map
Kinect Alternative: ASUS Xtion PRO
Summary

Processing
Processing Language
Installing Processing
Processing IDE
A First Processing Sketch
Processing Variables
Structure of a Processing Sketch
draw() Function
Processing Libraries
Simple-OpenNI
Installing Simple-OpenNI
Installation on Linux
Accessing the Depth Map and RGB Image
The Third Dimension
Processing in 3D
Camera Control Libraries
KinectOrbit Example
Kinect Space
Linear and Two-Dimensional Arrays
Coloring the Point Cloud
NITE Functions
Hand Tracking
Skeleton Tracking
Summary

Arduino and Kinect: "Hello World"
Parts List for Hello World
Serial Communication
Serial-Controlled LED
Writing Your Own Communication Protocol
Serial-Controlled LEDs
Kinect-Controlled LEDs
Feedback from Arduino
Summary

Kinect Remote Control
Parts List for Kinect Remote Control
Hacking a Remote Control
Connecting the Remote to Arduino
Assembling the Prototype Shield
Testing the Circuit
Kinect Hand Tracking and Gesture Recognition
Libraries and Setup
NITE Callbacks
Draw Loop and Other Functions
Connecting the Processing Sketch to Arduino
Summary

Kinect Networked Puppet
The Puppet
Servos
Building the Stage
Building the Puppet
Building the Circuit
Testing the Servos
Setting the Servos to the Starting Position
Skeleton Tracking On-Screen
Simple-OpenNI Events
Angle Calculation
Network Communication
Communicating Within a Local Network
Communicating over the Internet
Server Applet: Sending the Angles over a Network
Client Applet: Controlling the Puppet
Final Arduino Code
Summary

Mood Lamps
RGB Color Space

Arduino Nano
Building the Circuit
Resistors
Understanding Resistor Color Codes
Testing the Circuit
XBee Wireless Module
Arduino Programming
The Lamp Class
Object-Oriented Programming
User Control Sketch
Variable Declaration
Setup() Function
Draw() Function
User Control
Data Storage and Retrieval
Serial Communication
Display Functions
Simple-OpenNI Callbacks
Summary

Kinect-Driven Drawing Robot
Building the Robot
Part 1
Part 2
Part 3
The Base
Building the Circuit
Testing the Circuit
Firmata and the Arduino Library for Processing
Servo Test
Robot Simulation
Angle Measuring Program
Robot Simulation Program
Driving the Physical Robot
Kinect: Tangible Table Interface
Calibrating the Point Cloud
Virtual Robot Model
Polishing the Input
The Drawing Robot in Action
Summary

Kinect Remote-Controlled Vehicles
Electrical Motors and the H-Bridge
Hacking a Car
Building the Circuit
Testing the Circuit
Arduino Testing Sketch
Proximity Sensors
XBee and Wireless
Kinect RC Interface
Setup Function
Draw Function
Other Functions
Summary

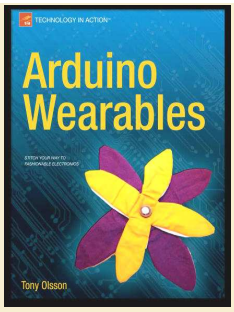
Biometric Station
Hacking a Bathroom Scale
Seven-Segment LCD
Hacking the LCD
Acquiring the LCD Signal
Reading the LCD Signal
Sending the Signal to Processing
Decoding the LCD Signal
Using the Weight Data
Implementing the Biometric Recognition
Imports and Variable Declaration
Setup Function
User Interface
Draw Function
Additional Functions
Summary

3D Modeling Interface
The Interface
Arduino LilyPad
Flex Sensors
Sewing the Circuit
Testing the Circuit
Arduino Serial Sender
Processing Glove Data Receiver
Going Wireless
LilyPad XBee
SoftwareSerial Library
Wireless Arduino Code
Implementing the Modeling Interface
The GloveInterface Class
Geometric Classes
Modeling Interface Main Program
Summary

Turntable Scanner
The Theory
Building a Turntable
Connecting the Gears
Building the Circuit
Arduino Code
Processing Code
Variable Declaration
Setup and Draw Functions
Additional Functions
Exporting the Point Cloud
The exportPly Function
Surface Reconstruction in Meshlab
Summary

Kinect-Controlled Delta Robot
About This Project
The Delta Robot
Building a Delta Robot
The Legs
The Base
The Effector
The Gripper
Hanging the Robot
Building the Circuit
Delta Robot Simulation Software
Inverse Kinematics
DeltaRobot Class
deltaLeg Class
Driving the Delta Robot Simulation with the Mouse
Kinect-Controlled Delta Robot
Gripper Control
Sending the Data to Arduino
Arduino Code

Summary



Tony Olsson

Arduino Wearables
«Носимый» Arduino

— Publisher: Apress, 2012. — 336 pages
— Series: Technology in Action

ISBN-10: 1430243597
ISBN-13: 978-1430243595

Вы, наверное, видели футболки и шляпы со световым LED оформлением, и возможно другие электронные гаджеты, встроенные в одежду, но с Arduino вы можете научиться делать свои собственные носимые электронные творения. Каждая глава покажет весь путь от идеи до готового проекта. Даже если у вас нет опыта работы с Arduino, эта книга поможет вам настроить всё программное обеспечение и оборудование, которое требуется. Вы освоите простые проекты, а затем приступите к более сложным.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments
Preface

Chapter 1: Introduction
Wearables.
World, Say Hello to Arduino
Sharing is Caring
Talking the Talk
Wearables
Wearable Computing
Inflatables
Moveables
Haptics
Embedded Technology
E-textile
Conductive Materials
Hacking
Prototyping
Techno Fashion
Interactivity
DIY
High and Low Tech
Critical Design
Physical Computing
Work Process
The Idea
Researching
Design
Building and Testing
Where to Buy Stuff
SparkFun Electronics
Adafruit Industries
RS Components
Farnell
Robot Italy
PlugHouse
Seeed Studio
Squarebit
electro:kit
Arduino Store
LessEMF
Further Reading

Chapter 2: Software
Installing the IDE
Installing the IDE on Windows
Installing the IDE on Mac OS X
Installing the FTDI Driver on OS X
Running the IDE
Examining the File Menu
Examining the Edit Menu.
Examining the Tools Menu
Examining the IDE Buttons
What Is Software?
How We Write Code
Variables
The Basic Structure of a Program
Basic Commands
Setting pinMode
Writing a Pin's State
Adding a Delay
Summary

Chapter 3: Hardware
The Arduino Hardware
A Closer Look
On the Inside
Board Layout
USB Connector
Digital Pins
Analog Pins
Power Pins
External Power
Electricity
Voltage, Current, and Resistance
Batteries
Battery Connectors
Testing the Hardware
Summary

Chapter 4: Smart Materials and Tools
Materials
Thermochromic Ink
Kanthal Wire
Conductive Foam.
Conductive Thread
Conductive Textile
Conductive Velcro
Conductive Paint
Smart Wire
Nitinol Wire
Tools
Alligator Clips
Breadboard
Multimeter
Camera
Helping Hands
Needle-nose Pliers
Soldering Iron
Wire Cutters
Summary

Chapter 5: LED Bracelets
Materials and Tools Needed
Soft Bracelet
Hard Bracelet
Using Multimeters
Conductivity
Measuring Resistance
Measuring Voltage
Using LEDs
Serial vs. Parallel

Basic Schematic for Parallel LED Bracelet Circuit
Soft Bracelet
Step 1-5
Hard Bracelet
Step 1-3
Finishing Up

Chapter 6: Solar-Powered Glow-in-the-Dark Bag
Materials and Tools Needed
Solar Panels
Power Diode
Resistors
Making a Battery Charger
Making the Bag
LDR Sensor Switch
Adding the Solar Panel
Adding the EL Panels
Putting Everything Together
The Final Bag
Wrapping Up

Chapter 7: Piano Tie.
Tools and Materials Needed
Making Sounds with Piezos
Cutting the Tie
Adding the Buttons
Stitching the Schematic
Checking the Layout
Writing the Code
Testing the Buttons
Making Sound
Piecing the Code Together
Piano Code
Synthesizing
Tone and noTone
Piano Tie Code Using tone and noTone
Finishing the Tie
Wrapping Up
Notes and Frequencies

Chapter 8: Bag Alarm
Tool and Materials Needed
Cutting the Shape
Analog Sensors
Making an Analog Zipper Sensor
Theory Behind the Zipper
Testing the Zipper
Serial Communication
Test Code
Making the Circuit
Stitching the Bag
Making the Button Hole
Sounding the Alarm
Bag Alarm Code
Using &&
Breaking into the Bag
Bag of Fun
Using Arrays
Storing the Notes and Making Melodies
Wrapping Up

Chapter 9: Beatbox Hoodie
Tools and Materials Needed
Making a Hoodie
Stitching Things
Skin Buttons
Button Arm and Hoodie Lining
Completing the Circuit
Connecting the Speakers
The Microphone
Battery Pack
Arm Buttons
Coding the Sequencer
The Code
Time Is the Key
millis()
Stereo vs. Mono
Final Stitching
Wrapping Up

Chapter 10: Sunshine Umbrella
Materials and Tools Needed
Getting Started
Multiplexing
Sensing Water
Constructing the Umbrella
Stitching the Water Sensor
Soldering the Matrix
Hooking Everything Up
Writing the Code
LED Test Code
Random Rain
Pin Point Control
Wrapping Up

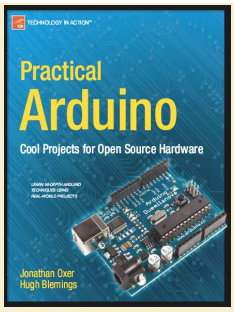
Chapter 11: Beat Dress
Tool and Materials Needed
Getting Started
The Little Dress Design
Cutting the Dress
Reenter the Matrix
Dress Stitching and LED Poking
Hooking Up the Arduino and Microphone
Writing the Code
Lightning Beat
Dress Texting
Wrapping Up

Chapter 12: Shape Memory Flower
Materials and Tools Needed
Shape Memory Alloy
Generating Heat
Testing the Wire
Creating the Flower
Connecting the Flower to a LilyPad
Programming the Flower
Using Booleans.
Adding the Light Sensor
Reprogramming the Wire
Wrapping Up

Chapter 13: EL Wire Dress
Materials and Tools Needed
EL Wire
The Dress Design
Making the Electronic Belt
EL Components
Adding Power to the Sequencer
Programming the Dress
Random Blinking
Adding Randomness to the Blinking
Fading the Wires with PWM
Finalizing the Dress
Wrapping Up

Chapter 14: Making Things Tiny
Materials and Tools Needed
The ATtiny
Libraries and Programming the ATtiny
Adding a Library
Tiny Speed
Programming the ATtiny
The Tiny Chip That Could
Three Tiny Projects
Skin Sound Bracelet
The Angry Rabbit
Tiny Boards
Wrapping Up

Index



Jonathan Oxer, Hugh Blemings

Practical Arduino. Cool Projects for Open Source Hardware
Arduino на практике.
Отличные проекты для оборудования с открытым исходным кодом

— Publisher: Apress, 2009. — 456 pages
— Series: Technology in Action

ISBN-10: 1430224770
ISBN-13: 978-1430224778

Создайте свои собственные конструкции на Arduino. Научитесь языку программирования, создавая практические проекты на базе Arduino, которые можно реализовать в домашних условиях.

CONTENTS

About the Author
About the Technical Reviewers
Acknowledgments

Introduction
Fundamentals
Sharing Your Work
Practical Electronics for Software Developers
Current, Voltage, and Power
Mains Is Nasty
Reading Schematics
Resistance and Resistors
Ohm's Law and Current Limiting
Choosing Wire
Diodes
Power Supplies
USB Power
Batteries
Wall Warts/Plugpacks
Capacitance and Capacitors
Fundamentals
Capacitor Types
Power Supply Bypass
ESD Precautions
Tools
Parts
Further Reading

Appliance Remote Control
Parts Required
Instructions
Test and Investigate Appliance Remote
Assemble Reed Relay Shield
Connect Reed Relay Shield to Remote Control
Create Reed Relay Control Program
ApplianceRemoteControl
ApplianceRemoteControlCompact
Test Reed Relay Shield and Sketch
Variations
Wireless Link
Automatic Trigger
Socket Connections

Time-Lapse Camera Controller
Parts Required
Instructions
Assemble Prototyping Shield
Connect Camera Shutter Release
Remote Shutter Release Connector
Infrared Remote Control
Modify Camera
Configure Camera
Calculate Photo Interval
Configure and Load Sketch
Check Memory Capacity
Set Up Your Shoot
Process the Images
Linux
Windows
Macintosh
Variations

Virtual USB Keyboard
Parts Required
Instructions
Populate Prototyping Shield
Prepare the UsbKeyboard Library
Compile and Upload Sketch
Variations
Chording Keyboard

PS/2 Keyboard or Mouse Input
Parts Required
Instructions
PS/2 Connections
Recycled 6-Pin Mini-DIN Sockets
6-Pin Mini-DIN Panel Sockets
PS/2 Extension Cable
Keyboard Software
Mouse Software
Variations Barcode Reader for a Stock Control System
Resources

Security/Automation Sensors
Parts Required
Instructions
Security Sensor Basics
Assemble Four-Channel Alarm Sensor Shield
Install End-of-Line Resistors on Sensor
Load Test Program
PIR Placement
Variations
Visual Display Written in Processing
Home Security System
Multiple Buttons on One Input

Online Thermometer
Parts Required
Instructions
Mount PCB Plugs on Shield
Connect Data Lines
Assemble Sensors
Load Program
Variations

Touch Control Panel
Parts Required
Instructions
How Resistive Touch Screens Work
Basic Touch Screen Connection Test
Arduino TouchScreen Library
Controlling a "Processing" Program
Construct Hardware for Home Automation Control Panel
Calibrate Hot Zones
Mount Arduino

Speech Synthesizer
Parts Required
Instructions
Speech Output Signal
Beginning Assembly
SpeakJet PWM "Audio" Output

Quick Test
Fit Status Indicators
Making a Line-Level Output Cable
On-Board Amplifier
Speech Synthesizer Software
Create Your Own Messages
Variations
Read SpeakJet Status
Resources

Water Flow Gauge
Parts Required
Instructions
Replace Sensor Connector
Prepare Prototyping Shield
Prepare LCD Module
Fit LCD to Case
Fit Arduino in Case
Determine Scaling Factor
Configure, Compile, and Test Sketch
Hardware Interrupts
Volatile Variables
Flow Gauge Sketch
Install Flow Sensor
Variations
Online Logging
Multiple Sensors
Resources

Oscilloscope/Logic Analyzer
Parts Required
Instructions
Assemble the Case
Fit the Prototyping Shield
Make Test Probes
How Successive Approximation ADC Works
Connection to Circuit under Test
Install Software in Arduino
Analog Read Version
Digital Read Version
Digital Read Sketch with Optimized Communications Format
Install Processing and Run Visualization Program
Variations
Input Voltage Prescaler
Resources

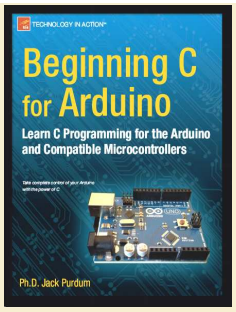
Water Tank Depth Sensor
Parts Required
Instructions
Assemble the Shield
Determine Sensor Installation Method
Assemble the Sensor Housing
Assemble the Arduino Housing
Install the WiShield Library and Driver
Load the Tank-Level Sketch
Prettier Web Interface
Calibrating the "Empty Tank" Level
Install the Sensor and Arduino
Calibrating the "Full Tank" Level
Variations
Upload Data to Pachube
Control Pumps or Irrigation
Local Level Display
Reading Multiple Tanks

Weather Station Receiver
Parts Required
Instructions
Test and Install the Weather Station
Understanding the Circuit
Assemble the Receiver Shield
Weather Station Receiver Program
Twitter Weather Updates
Variations
Private Online Weather Station

RFID Access Control System
Parts Required
Instructions
Assemble the Prototyping Shield
Power Supply
RFID Serial Connection
Relay Output
Indicator LEDs
Manual Release Button
Assemble the ID-12 Reader Module
Assemble the RDM630 Reader
Simple Stand-Alone Sketch
Install the Strike Plate
Install the Reader
Install the Arduino
Variations
Extensible Read Head
Speech Synthesizer Feedback
Intelligent Desk Pad
Resources

Vehicle Telemetry Platform
Parts Required
Instructions
Check the Vehicle Interface
Obtain a USB/OBD-II or RS-232 Adapter
Test the USB/OBD-II Adapter
Understanding OBD-II Modes and Parameters
Prepare the USB/OBD-II Adapter
Assemble the OBD-II Cable
Prepare the GPS Module
Assemble the Power Supply on the Shield
Fit the Serial Connections on the Shield
Prepare the VDIP1 Module
The LCD Module
Logging Control Button and Status LEDs
Mount in Sub-Assemblies in the Case
OBDuino Mega Sketch
OBDuinoMega.pde
LCD.pde
GPS.pde
VDIP.pde
Host.pde
PowerFail.pde
Using the OBDuinoMega Sketch Menu Buttons
Running Logging
Generate Google Earth Track
Generate Charts of Parameters
Variations
Mobile Internet Connection
Vehicle Control
Speech Synthesizer Output
3D Accelerometer
Digital Compass
"Knight Rider"–Style Alarm Status
Battery Condition Check
Resources

Resources
Simple Voltage Regulators
Power Dissipation
Dropout Voltage
Efficiency
Driving LEDs
Multiplexing
Charlieplexing
Output Circuitry and Isolation
Semiconductor and Mechanical Switches
Reed Relays
Relays
Bipolar Transistors
Transistor Arrays
FETs
Optocouplers
Solid State Relays
Digital Input/Output Expansion
Shift Registers As Outputs
Shift Registers As Inputs
Input Conditioning
Voltage Dividers and Nonisolated Input Conditioning
Isolated Input Conditioning
Inexpensive DIY Prototyping Shields
Writing an Arduino Library
Develop Functionality As a Sketch
Create the Library Header File
Create the Library Class File
Create the Example Sketch
Create Supporting Files
Platform-Specific Variations
Summary



Jack Purdum

Beginning C for Arduino: Learn C Programming for the Arduino

Знакомство с Си для Arduino:
обучение программированию на Си для Arduino

— Publisher: Apress, 2012. — 280 pages
— Series: Technology in Action
ISBN-10: 1430247762
ISBN-13: 978-1430247760

Книга написана для тех, кто не имеет опыта работы с микроконтроллерами и их программированием, но хотел бы учиться и экспериментировать. Она знакомит вас с языком программирования Си, на простых примерах программирования плат Arduino. Автор использует привлекательный стиль обучения, чтобы привить хорошие приёмы программирования, которые были отточены за время его 25-летнего преподавания в университете.

CONTENTS

About the Author
About the Technical Reviewer
Acknowledgments
Introduction
Assumptions About You
Resources

Chapter 1: Introduction
Assumptions About You
What You Need
An Atmel-Based Microcontroller Card
Breadboard
Miscellaneous Parts
Verifying the Software
Verifying the Hardware
Attaching the USB Cable
Selecting Your
c Board in the Integrated Development Environment
Port Selection
Loading and Running Your First Program
The Blink Program
Running the Blink Program Code
Compiling the Blink Program Code
Uploading the Blink Program
Summary

Chapter 2: Arduino C
The Building Blocks of All Programming Languages
Expressions
Statements
Statement Blocks
Function Blocks
The Five Program Steps
1. Initialization Step
2. Input Step
3. Process Step
4. Output Step
5. Termination Step
The Purpose of the Five Program Steps
A Revisit to the Blink Program
Program Comments
Data Definition
The setup() Function
The loop() Function
Summary
Exercises

Chapter 3: Arduino C Data Types
Unknown
Keywords in C
Variable Names in C
The boolean Data Type
The char Data Type
Binary Data
The char Data Type and Character Sets
Generating a Table of ASCII Characters
The byte Data Type
The int Data Type
The word Data Type
The long Data type
The float and double Data Types
Floating Point Precision
The string Data Type
String Data Type
The void Data Type
The array Data Type
Defining versus Declaring Variables
Symbol Tables
lvalues and rvalues
The Bucket Analogy
Using the cast Operator
Summary
Exercises

Chapter 4: Decision Making in C
Relational Operators
The if Statement
A Modified Blink Program
The Circuit
The Program Code
The if-else Statement
Cascading if Statements
The Increment and Decrement Operators
Two Types of Increment Operator (++)
Two Flavors of the Decrement Operator (--)
Precedence of Operators
The switch Statement
The goto Statement
Getting Rid of Magic Numbers
The C Preprocessor
Heads or Tails
Initialization Step
Input Step
Process Step
Output Step
Termination Step
Something to Think About
Summary
Exercises

Chapter 5: Program Loops in C
The Characteristics of Well-Behaved Loops
Condition 1: Variable Initialization
Condition 2: Loop Control Test
Condition 3: Changing the Loop Control Variable's State
Using a for Loop
The while Loop
The do-while Loop
The break and continue Keywords
The break Statement
The continue Statement
A Complete Code Example
Step 1. Initialization
Step 2. Input

Step 3. Process
Step 4. Output
Step 5. Termination
Listing 5-1 is Sorta Dumb Code
Loops and Coding Style
Summary
Exercises

Chapter 6: Functions in C
The Anatomy of a Function
Function Type Specifier
Function Name
Function Arguments
Function Body
Function Signature
What Makes a "Good" Function
Functions Use Task-Oriented Names
The Function Should Be Cohesive
Functions Should Avoid Coupling
Writing Your Own Functions
Function Design Considerations
Argument List
Function Body
Logical Operators
Logical AND Operator (&&)
Logical OR (||)
Logical NOT (!)
Writing Your Function
The IsLeapYear() Function and Coding Style
Arguments versus Parameters
Why Use a Specific Function Style?
Leap Year Calculation Program
Passing Data Into and Back From a Function
Pass by Value
Summary
Exercises

Chapter 7: Storage Classes and Scope
Hiding Your Program Data
Statement Block Scope
Local Scope
Name Collisions and Scope
Global Scope
Global Scope and Name Conflicts
Scope and Storage Classes
The auto Storage Class
The register Storage Class
The static Storage Class
The extern Storage Class
The volatile keyword
Summary
Exercises

Chapter 8: Introduction to Pointers
Defining a Pointer
Pointer Name
Asterisk
Pointer Type Specifiers and Pointer Scalars
Using a Pointer
The Indirection Operator (*)
Using Indirection
Summary of Pointer Rules
Why Are Pointers Useful?
Pointers and Arrays
The Importance of Scalars
Summary
Exercises

Chapter 9: Using Pointers Effectively
Relational Operations and Test for Equality Using Pointers
Pointer Comparisons Must be Between Pointers to the Same Data
Pointer Arithmetic
Constant lvalues
Two-Dimensional Arrays
A Small Improvement
How Many Dimensions?
Two-Dimensional Arrays and Pointers
Treating the Two-Dimensional Array of chars as a String
Pointers to Functions
Arrays of Pointers to Functions
enum Data Type
The Right-Left Rule
Summary
Exercises

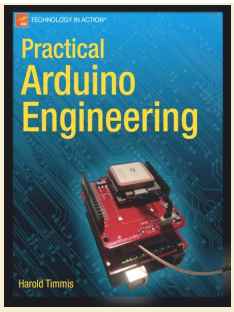
Chapter 10: Structures, Unions, and Data Storage
Structures
Declaring a Structure
Defining a Structure
Accessing Structure Members
Returning a Structure from a Function Call
Using Structure Pointers
Initializing a Structure
Arrays of Structures
Unions
EEPROM Memory
Using EEPROM
Data Logging
Other Storage Alternatives
Shields
Other Uses for Secure Digital Storage
Summary
Exercises

Chapter 11: The C Preprocessor and Bitwise Operations
Preprocessor Directives
#undef
#line
#if, Conditional Directives
#include
Parameterized Macros
Bitwise Operators
Bitwise Shift Operators
Using Different Bases for Integer Constants
Parameterized Macros...continued
Summary
Exercises

Chapter 12: Arduino Libraries
Libraries
Arduino Libraries
Other Libraries
Writing Your Own Library
The Library Header File
The Library Code File (Dates.cpp)
Setting the Arduino IDE to Use Your Library
A Sample Program Using the Dates Library
Adding the Easter Program to the IDE
The keywords.txt File
Keyword Coloring (theme.txt)
Summary
Exercises

Appendix A: Suppliers and Sources
Suppliers
Seeedino Studio
Diligent Inc
OSEPP
Tinyos Electronics
Cooking Hacks
Sources
Software

Appendix B: Electronic Components for Experiments
Microcontroller Board
Solderless Breadboard
Electronic Components
Online Component Purchases
Experiment!



Harold Timmis

Practical Arduino Engineering
Практическое конструирование с Arduino

— Publisher: Apress, 2011. — 328 pages
— Series: Technology in Action

ISBN-10: 1430238852
ISBN-13: 978-1430238850

Платы Arduino произвели впечатление на хакеров и профессиональных инженеров. Если ли Вы человек, увлеченным своим хобби, или профессионал, то Arduino позволит вам продвинуться дальше туманной идеи и макетной платы. Это важно для правильного проектирования приборов и, действительно, полного тестирования ваших проектов перед изготовлением.

Книга объясняет в общих чертах процесс разработки: от основных требований и предварительного проекта - к анализу прототипа и тестированию. Каждая глава иллюстрирует этот процесс и демонстрирует, как вы можете получить прибыль от реализации технических идей — независимо от того, играете ли вы просто в своем подвале или хотите предать гласности и продать ваши разработки.

CONTENTS

About the Author
About the Technical Reviewers
Acknowledgments
Preface

The Process of Arduino Engineering
Gathering Your Hardware
Gathering Your Tools
Understanding the Engineering Process
Requirements Gathering
Creating the Requirements Document
Gathering the Hardware
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

Understanding the Arduino Software
Getting Started with setup() and loop()
Initializing Variables
Writing Conditional Statements
Working with Loops
Communicating Digitally
Communicating with Analog Components
Serial Communication
Using Arduino Libraries
NewSoftwareSerial
TinyGPS
ColorLCDShield Library
Putting Together the Arduino Language Basics
Summary

Robot Engineering Requirements: Controlling Motion
Hardware Explained: The H-bridge
Gathering the Hardware for this Chapter
Understanding the Basics of Motor Control
Project 3-1: Turning on a Motor with a Switch
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 3-2: Controlling the Speed of a Motor with a Potentiometer
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 3-3: Controlling Multiple Motors with the Arduino
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 3-4: Controlling Speed and Direction
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 3-5: Controlling Motors with Serial Commands
Requirements Gathering
Gathering the Hardware
Outlining the Software Requirements
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

Adding Complexity to the Robot: Working with LCDs
Configuring a Color LCD Shield
Introducing the Monochrome and Color LCD Shields
Working with the LiquidCrystal and ColorLCDShield (Epson or Phillips) Libraries
Using the LiquidCrystal Library
ColorLCDShield Library
Exploring the Basics of LCD Control
Project 4-1: Displaying Multiple Sensor Values
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 4-2: Creating a Menu on the Monochrome LCD
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 4-3: Creating a Slot Machine with the Color LCD Shield
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 4-4: Using a Keypad to Communicate with the Color LCD
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 4-5: Creating the Customer's Robot
Requirements Gathering and Creating the Requirements Document
Gathering the Hardware
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

Robot Integration Engineering a GPS Module with the Arduino
Hardware Explained: microSD Shield
Understanding NMEA Protocol
Libraries Explained: TinyGPS and SdFat Libraries

TinyGPS
SdFat Library
The Basics of GPS Communication with the Arduino
Project 5-1: Writing Raw GPS Data to the Serial Monitor
Hardware for This Project
Configuring the Hardware
Writing the Software
Project 5-2: Writing GPS Data to a Monochrome LCD
Hardware for This Project
Configuring the Hardware
Writing the Software
Project 5-3: Creating a Car Finder
Hardware for This Project
Configuring the Hardware
Writing the Software
Project 5-4: GPS Data Logger
Hardware for This Project
Configuring the Hardware
Writing the Software
Requirements Gathering and Creating the Requirements Document
Hardware
Software
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

Interlude: Home Engineering from Requirements to Implementation
Understanding the Voltage Divider
Hardware Explained: Sensors
Photoresistor
Tilt Sensor
Flex Sensor
Force Sensitive Resistor (FSR)
Digital Temperature and Humidity Sensor
Digital Temperature Sensor (I2C)
Libraries Explained: Wire Library and DHT22 Library
Wire Library
DHT22 Library
Understanding the Basics of Sensors
Project 6-1: Flower Pot Analyzer
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 6-2: Using a FSR Sensor
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 6-3: Using a Flex Sensor
Hardware for this Project
Configuring the Hardware
Writing the Software
Project 6-4: Digital Level
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 6-5: Using a DHT22 Sensor with a Monochrome LCD
Gathering the Hardware
Configuring the Hardware
Writing the Software
Project 6-6: Wireless Temperature Monitor
Requirements Gathering and Creating the Requirements Document
Gathering the Hardware
Outlining the Software Requirements
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Final Prototype
Summary

Robot Perception: Object Detection with the Arduino
Hardware Explained: Ultrasonic Sensor, Servo, and Buzzer
Ultrasonic Sensor
Servo
Buzzer
Libraries Explained: The Servo Library
Basics of the Ultrasonic Sensor and the Servo
Project 7-1: Digital Ruler
Hardware for this Project
Configuring the Hardware
Writing the Software
Project 7-2: Object Alarm
Hardware for This Project
Configuring the Hardware
Writing the Software
Project 7-3: Solar Controller
Hardware for This Project
Configuring the Hardware
Writing the Software
Requirements Gathering and Creating the Requirements Document
Hardware
Software
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

Mature Arduino Engineering: Making an Alarm System Using the Arduino
Hardware Explained: PIR
Basic Security System
Project 8-1: Door Alarm
Hardware for This Project
Configuring the Hardware
Writing the Software
Requirements Gathering and Creating the Requirements Document
Hardware
Software
Configuring the Hardware
Writing the Arduino Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

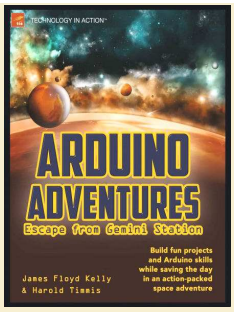
Error Messages and Commands: Using GSM Technology with Your Arduino
Hardware Explained: Cellular Shield
Understanding the AT Command Set
The Basics of GSM Communication
Project 9-1: Sending a Text Message
Hardware for This Project
Configuring the Hardware
Writing the Software
Project 9-2: Door Alarm with SMS Messaging
Hardware for This Project
Configuring the Hardware
Writing the Software
Requirements Gathering and Creating the Requirements Document
Hardware
Software
Configuring the Hardware
Writing the Software
Debugging the Arduino Software
Troubleshooting the Hardware
Finished Prototype
Summary

Control and Instrumentation: The Xbox Controller and the LabVIEW Process
Introduction to the LabVIEW Environment
The Front Panel
The Controls Palette
The Block Diagram
The Functions Palette
The Tools Palette
LabVIEW Functions Explained
The While Loop
The Case Structure
The Sequence Structure
Numerical Functions
String Functions
Comparison Functions
Serial Functions
Input Device Control Functions
Gathering Requirements and Creating the Requirements Document
Hardware
Software
Configuring the Hardware
Writing the Software
Getting Started
Designing the GUI
Programming the Application
Adding Serial Functions

Completing the While Loops Condition
Adding a Merge Errors Function
Adding a SubVI
Error Handling
Uploading the Code to the Arduino
Operation
Debugging the LabVIEW Software
Troubleshooting the Hardware
Finished Prototype
Summary

Controlling Your Project: Bluetooth Arduino
Gathering Requirements and Creating the Requirements Document
Gathering Hardware
Gathering Software
Configuring the Hardware
Soldering the Headers
Pins and Beyond
Configuring the Chassis and Arduino
Writing the Software
Reviewing the Arduino Software
Uploading the Software and Attaching the Bluetooth Shield
Operating the Robot
Debugging the Software
Troubleshooting the Hardware
Finished Prototype
Summary

Hardware and Tools



James Floyd Kelly, Harold Timmis

Arduino Adventures: Escape from Gemini Station
Приключения с Arduino: побег со станции Близнецов

— Publisher: Apress, 2013. — 332 pages

ISBN-10: 1430246057
ISBN-13: 978-1430246053

«Приключения с Arduino: побег со станции Близнецов» – это забавное знакомство с микроконтроллером Arduino. Вы попадёте в действие научно-фантастического приключенческого романа, окажетесь свидетелем того как Кейд и Элли исследуют станцию Близнецов — орбитальный музей, созданный для сохранения и совместного использования технологии на протяжении веков.

Шаг за шагом на простых примерах книга знакомит читателя с платформой Arduino. От простого мигающего светодиода, до более сложных проектов по управлению роботом. После изучения материала книги вы сможете реализовать любой вид электронно-управляемого устройства, которое сможете вообразить, используя потрясающе популярную плату Arduino.

CONTENTS

About the Authors
About the Technical Reviewers
Acknowledgments
Introduction

Chapter 1: Trouble at Gemini Station
Trouble Begins
On the Level, or Not?
Andrew 5.0
Boom!
Escape, or Not
A Plan

Chapter 2: Challenge 1: Fun Stuff to Know
What Is an Arduino?
Giving an Arduino a Job to Do
Installing the Software
Things to Watch for on Windows
The Development Environment
Ready to Build Something?

Chapter 3: Challenge 1: Examining the Hardware
Locating the Parts You'll Need
Potentiometer
Solderless Breadboard
The Arduino Uno
Wire
Let's Build Gizmo #1
What's Next?

Chapter 4: Challenge 1: Examining the Software
The Arduino IDE
The Challenge #1 Sketch
Beginning the Sketch
Con?guring the Serial Port
Listening on the Serial Port
Translating the Input into Digits
Displaying the Result
Solving Challenge #1

Chapter 5: Damage Assessment
The Face of Andrew
An Embarrassed Cade
The Unlocking

Chapter 6: Challenge 2: Fun Stuff to Know
Let's Look at a Battery
And Now a Circuit
Current Flow
Ready to Build Something?

Chapter 7: Challenge 2: Examining the Hardware
The Push Button
The Light Emitting Diode
The Resistor
Let's Build Gizmo #2
What's Next?

Chapter 8: Challenge 2: Examining the Software
Functions Explained
The Challenge 2 Sketch
Solve Challenge #2

Chapter 9: Feeling The Heat
On a Pedestal
Chutes and Ladders
Green-eyed Hatches

Chapter 10: Challenge 3: Fun Stuff to Know
Looking at the Temperature Sensor
Ready to Build Something?

Chapter 11: Challenge 3: Examining the Hardware
What Is a Sensor?
Let's Build Gizmo 3!

Chapter 12: Challenge 3: Examining the Software
The Conditional If-Else Statement
The Challenge #3 Sketch
Solve Challenge #3

Chapter 13: Uninvited Guest
Upward
Spooky?
Urgency!
Danger!
Bucket

Chapter 14: Challenge 4: Fun Stuff to Know
Looking at the Bucket Mover
Understanding the ICs
Ready to Build Something?

Chapter 15: Challenge 4: Examining the Hardware
New Hardware
Let's Build Gadget #4

Chapter 16: Challenge 4: Examining the Software
The Challenge 4 Sketch
Breaking It Down
Solve Challenge 4

Chapter 17: Hide and Seek
The Crossing
Five Minutes!
Run!
Walk

Chapter 18: Challenge 5: Fun Stuff to Know
Let's Look at the Challenge 5 Gizmo
Ready to Build Something?

Chapter 19: Challenge 5: Examining the Hardware
A Closer Look at the PIR Sensor
Let's Build The Challenge 5 Gizmo

Chapter 20: Challenge 5: Examining the Software
Thinking Through the Solution
Understanding the Tone Function
The Challenge #5 Sketch
Solve Challenge #5

Chapter 21: Carousel Ride
Close Call
Nothing To See Here
An Engineering Problem

Chapter 22: Challenge 6: Fun Stuff to Know
Let's Look at the Challenge 6 Gizmo
Ready to Build Something?

Chapter 23: Challenge 6: Examining the Hardware
A Closer Look at a Servo Motor
Let's Build the Challenge 6 Gizmo

Chapter 24: Challenge 6: Examining the Software
Servo Library Explained
The Challenge #6 Sketch
Solve Challenge #6

Chapter 25: Push the Button
Backup Plan
Control Center
Crazy Plan
The Flashlight

Chapter 26: Challenge 7: Fun Stuff to Know
Let's Look at the Challenge 7 Gizmo
Ready to Build Something?

Chapter 27: Challenge 7: Examining the Hardware
A Closer Look at a Photoresistor
Let's Build the Challenge 7 Gizmo
Chapter 28: Challenge 7: Examining Software
The Challenge #7 Sketch
Solve Challenge #7

Chapter 29: Off the Station
The Hulk
Launch Problem
The Final Fix

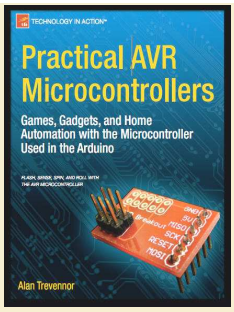
Chapter 30: Challenge 8: Fun Stuff to Know
The Basic Components
Let's Look at the Challenge 8 Chassis
Ready to Build Something?
Chapter 31: Challenge 8: Examining the Hardware
New Hardware
Let's Build Gizmo #8

Chapter 32: Challenge 8: Examining Software
Functions Explained
The Challenge #8 Sketch
Solve Challenge #8
You're Not Done!

Chapter 33: Epilogue
Three Weeks Later

Appendix A: Parts List
Challenge 1: Potentiometer
Challenge 2: LED Flashlight
Challenge 3: Temperature Detector
Challenge 4: Motor Control
Challenge 5: Motion Detector
Challenge 6: Servo Motor Control
Challenge 7: Light-Sensing Motor Control
Challenge 8: Build Your Own Robot

Tools



Alan Trevennor

Practical AVR Microcontrollers: Games, Gadgets, and Home Automation with the Microcontroller Used in the Arduino

Микроконтроллеры AVR на практике: игры, гаджеты и домашняя автоматизация с микроконтроллером, используемым в Arduino

— Publisher: Apress, 2012. — 416 pages
— Series: Technology in Action

ISBN-10: 1430244461
ISBN-13: 978-1430244462

Вы изучите, как использовать микроконтроллер AVR, чтобы сделать собственные изящные проекты и гаджеты. В первой части вы начнёте с установки среды разработки, узнаете чем "голый" AVR отличается от Arduino, и далее, вы приобретёте опыт, переходя от простых проектов к более сложным.

CONTENTS

Foreword
About the Author
About the Technical Reviewer
Acknowledgments
Introduction

Part 1: The Basics

Chapter 1: A Brief History of Microcontrollers
A Microcontroller Timeline
Why Microcontrollers?
Why Should You Learn About Microcontrollers?
What Can You Do with a Microcontroller?
Why AVR?
Summary
Coming Up Next

Chapter 2: Building Our AVR Test Bed
Test Bed Details
Test Bed Ingredients
A Desktop Computer
Summary List of Other Parts
Toolkit: Required Items
Toolkit: Optional Items
Piece Parts
AVR ISP Programmer
Other Programmers
AVR ISP Programming Adaptor and Pin Headers
Assembling the Programming Adaptor
Test Bed Software—Installation and Setup
Choosing the Software: Why Arduino?
Test Bed Testing
TTL Level Serial Port
Summary
Coming Up Next

Chapter 3: Arduino and the Naked AVR
Comparing Approaches
AVR Pulls Ahead
Go AVR!
Putting All the Pieces in Place
The AVR Type
Arduino's Configuration Files:
AVR Out of the Box
Using an External Crystal
The AVR and Its Fuses
AVRDude
The Trouble with AVRDude's Terminal Mode
AVRDude: Getting Started
Arduino and the AVR ID Problem
Pin Name Translations
AVR: Speak to Me!
Terminal Emulators
Summary
Coming Up Next
Summary

Chapter 4: Moving On!
Making Things That Move
The Servo Motor
Giving a Servo Motor a Testing Time
The Gem Light
Drive On!
MOS-What?
Relays
Solenoids
Motors (Non-Servo)
Sensing Movement
Sensor Switches
Active Sensors
Summary
Coming Up Next

Chapter 5: Smarten Up!
Intelligent Devices
Increased Functionality
Adaptive Behavior
Increased Efficiency
Improved Problem Management
Capacity for Repurposing
Greener Products
Support from the Mother Ship
Embedded Intelligence Benefits
Downsides of Intelligent Devices
The Anatomy of a Smart Device
Summary
Coming Up Next

Chapter 6: Digitally Speaking
When Intelligent Devices Speak
Communications Channel Types
Error Checking
Microcontroller KISS Communications
The RS-232 Connection
RS-232 or Direct Logic Level Connection?
Designing the Time-Getter Protocol
Talking Tokens
Object Models
Implementing a Tokenized Object Model on AVR
Summary of Object Models and Tokenization
Summary
Coming up

Part 2: The Projects

Chapter 7: Introduction to the Projects Section
Project Bases
Project Chapter Formats
Project Scope and Difficulty

Chapter 8: Project 1: Good Evening, Mr. Bond: Your Secret Panel
A Life Ruined by Movies and TV
Making It Real

The Fireline Fiasco
Thinking Again
Racking Up a Success
Hiding the Button
Position Sensors
The Secret Compartment
The Electronics
Sliding Panel Electronics Troubleshooting
Software Commentary
All Together Now!
Summary
Coming up Next

Chapter 9: Project 2: Crazy Beams—Exercise Your Pet!
The Project
Sourcing the Lasers
Project Build
Assembly
The Electronics
Crazy Beams—The Software
Summary
Coming Up Next

Chapter 10: Project 3: WordDune
How Much Do You Really See?
The Hardware
WordDune: The Build
WordDune Gameplay
WordDune: Game Software Setup
WordDune Commands
General Software Structure
WordDune Sketch Code Walk
WordDune: Declarations Section
setup()
loop()
format_EEPROM()
addToDictionary()
dumpEEPROMToSerialChannel()
dumpDictToSerialChannel()
Summary
Coming Up Next

Chapter 11: Project 4: The Lighting Waterfall
The Passageway Light
Proving Circuit
Waterfall Lights Software
Moving to Making It
Waterfall Lights Mark II
Mark II Electronics
Take-Two Circuit Diagram
Summary
Coming Up Next

Chapter 12: Moving to Mesmerize
Duck Shooter Game
The Duck Shooter Circuit
Making Duck Shooter into a Keeper
Building Duck Shooter
Going Further
Duck Shooter Software
MCU Shadow Puppets
Building the Shadow Puppets
The Shadow Puppet Circuit
Shadow Puppets Software
The Moire Wheel
Waterslide Decals
Building the Moire Wheel Project
Moire Wheel Software
Summary
Coming Up Next

Chapter 13: Smart Home Enablers
Is Your Home Smart?
Socket Cracking
On the Radio: The Un-wired Home
Radial or Wiring Needs
Back to Basics
Low-Cost Wireless Communication
Smart Home Using a Mixed Network
A Simple HomeHelp Protocol
Using a Low-Cost Smart Home Wireless Node
Door Sensor Circuit Diagram
Door Sensor Sender Software
Door Sensor Receiver Software
Door Sensor Summary.
Remotely Commandable Light Stand
Light Stand Electronics
Light Stand: RGB Values Sender
Light Stand Software: Receiver
The Light Stand in Action
Summary
Coming Up Next

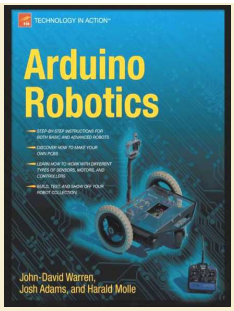
Appendix A: Common Components
Resistors
Resistor Specifications
Resistor Types and Packagings
Capacitors
Measuring Capacitance
Time for a Capacitance
Capacitor Shapes and Sizes
Light-Emitting Diodes

Appendix B: A Digital Electronics Primer
The Highs and the Lows
I Count—In Denary?
Deciding, Logically
General Gates
AND Gate
NAND Gate
OR Gate
NOR Gate
XOR Gate
Understanding the Specifications of Gates
Real Live Gates

Appendix C: Breadboards

Appendix D: Serial Communications
Data Transfer Basics
Binary Me!
The Byter Bit
The Trouble with Parallel
Tear It Down, Ship It Out!
Asynchronous Data Transmission.
Stop Bits
Word Length
Parity Checking and Error Detection
Protocol Overhead
Okay, So Tell Me About RS-232 and RS-485
Are There Other Types of Serial Data Interface?

Index



John-David Warren, Josh Adams, Harald Molle

Arduino Robotics
Роботы Arduino

— Publisher: Apress, 2011. — 628 pages
— Series: Technology in Action

ISBN-10: 1430231831
ISBN-13: 978-1430231837

Эта книга покажет Вам, создать различных роботов, в основе управления которых находится плата Arduino.

CONTENTS

About the Authors
About the Technical Reviewers
Introduction

CHAPTER 1 The Basics
Electricity
Electrical Analogy
Electrical Basics
Circuits
Measuring Electricity
Multi-Meters
Measuring Voltage
Measuring Amperage
Measuring Capacitance
Measuring Resistance
Calculating Resistor Power Using Ohm's Law
Oscilloscope
Loads
Inductive Loads
Resistive Loads
Electrical Connections
Series Connections
Parallel Connections
Series and Parallel Connection
Electronics
Semi-Conductors
Datasheets
Integrated Circuits
Packages
Through-Hole Components
IC Sockets
Surface-Mount Components (SMT or SMD)
Arduino Primer
Arduino Variants
Standard Arduino
Arduino Mega
Clones
Arduino IDE
The Sketch
Variable Declaration
The Setup Function
The Loop Function
Signals
Digital Signals
Digital Inputs
Digital Outputs
Special Case: External Interrupts
Analog Signals
Analog Inputs
Analog Outputs (PWM)
Duty-Cycle
Frequency
Homemade PWM Example
Building Circuits
Circuit Design
Schematics
Prototyping
Breadboard
Perforated Prototyping Board (Perf-Board)
Printed Circuit Boards
Soldering
Soldering Shortcuts
Building a Robot
Hardware
Basic Building Tools
Materials
Work Area

CHAPTER 2 Arduino for Robotics
Interfacing Arduino
Relays
Types of Relays
Relay Configurations
Uses
Calculating Current Draw
Back-EMF Considerations
Solid-State switches
Transistors
Bipolar Junction Transistor (BJT)
Mosfets
Logic-Level vs. Standard
Mosfet Capacitance
On-State Resistance—Rds(On)
Calculating heat using Rds(On) and amperage of DC motor
Parallel Mosfets
Photo-Transistors
Interfacing a Motor-Controller
Motor-controller ICs
Electronic Speed Controllers (ESCs)
User Control
Tethered (Wired) Control
Infrared Control (IR)
Radio Control Systems
Amplitude Modulation (AM)
Frequency Modulation (FM)
Spread Spectrum 2.4gHz
Xbee
Sensor Navigation
Contact Sensing
Bump Switch
Distance and Reflection Sensing
IR Sensor
Ultrasonic Range Finder
Laser Range Finder
Orientation (Positioning)
Accelerometer
Gyroscope
Inertial Measurement Unit (IMU)
Global Positioning Satellite (GPS)
Real Time Kinetic (RTK) GPS
Non-Autonomous Sensors
Camera
Current Sensor

CHAPTER 3 Let's Get Moving
Electric Motors
Brushed DC Motor (Permanent Magnet Type)
Brushless Motors
Stepper Motors
Gear Motors
Servo Motors
Continuous Rotation
Linear Actuators
Calculating Power

Driving
Finding the Right Motor
The H-Bridge
Generating a Brake
Implementation
Method 1: Simple Switches
Method 2: DPDT Relay with Simple
Method 3: P-Channel and N-Channel Mosfets
Method 4: N-Channel H-Bridge
H-Bridge ICs
Changing PWM Frequencies
Back EMF
Current Sensing
Commercial H-Bridges (Motor-Controllers)
Small (Up to 3amps)
Medium (Up to 10amps)
Large (Over 10amps)
The Open Source Motor Controller (OSMC)
Batteries
Nickel Cadmium (NiCad)
Nickel Metal Hydride (NiMH)
Lithium Polymer (LiPo)
Lead-Acid
Charging
Materials
Wood, Metals
Nuts and Bolts
Plastics
Chain and Sprockets
Wheels

CHAPTER 4 Linus the Line-Bot
Parts List for Linus
How Linus Works
The Track
Building the IR Sensor Board
Modifying a Servo for Continuous Rotation
Method 1: Direct DC Drive with External Speed Controller
Method 2: Servo Pulse Drive with Internal Motor Driving Circuitry
Fitting the Drive wheels
Making a Frame
Making Connections
Installing Batteries
Install Power Switch
Loading the Code
Making the Track
Testing
Add-ons
LED Lights
Painting
Adding Speed Regulator (Potentiometer)

CHAPTER 5 Wally the Wall-Bot
How Wally Works
Parts List for Wally
The Motor-Controller
The High-Side Switches
The Low-Side Switches
Building the Circuit
Building the Frame
Installing the Sensors
Installing the Battery and Power Switch
Installing a Power Switch
The Code
Code Objectives

CHAPTER 6 Making PCBs
PCB Basics
What You Need to Get Started
Circuit Design
Searching for Open-Source Designs
Making Your Own Designs
Schematics
Board Layouts
Working with the Schematic Editor
Working with the Board Editor
Transferring the Design
Let's Make an Arduino Clone: the Jduino
Making the Transfer
Etching
Measuring the Solution
Etching: Method 1
Etching: Method 2
Removing the Toner
Drilling, Soldering
Building the Arduino Clone
Building the BJT H-Bridge
Testing

CHAPTER 7 The Bug-Bot
Reading a Switch with Arduino
How the Bug-bot Works
Antennae Sensors
Bumper Sensors
Parts List for the Bug-bot
The Motors
Modifying the Servo Motors
Controlling the Servo Motors
Converting the Pulse Value to Degrees
Mounting the Wheels to the Servos
Building the Frame
Marking the PlexiGlass
Cutting the PlexiGlass
Mounting the Motors
Mounting the Caster Wheels
Mounting the Arduino
Installing the Battery
Making the Sensors
The Front Antennae Sensors
The Rear Bump Sensors
Making Wire Connections
Loading the Code
Creating a Delay
Variables
The Code
Making a Top Hat

CHAPTER 8 Explorer-Bot
How the Explorer-Bot Works
R/C Control
Powerful Motors
Current Sensing
Video-Enabled
Xbee-Enabled
Parts List for the Explorer-Bot
Building the Frame
Specs
Adding Battery Bracket
Cut Top Frame Brackets
Cut Top Frame Braces
Cut and Bend Main Frame Piece
Add Crossbar and Mount Caster Wheel
Plexiglass Deck (Optional)
Building the Motor Controller
Current Sensing and Limiting
H-bridge Design
Setting Up the Arduino
Connecting the H-Bridges
Setting Up Xbee
Testing the Xbees
Adding a Camera
Pan and Tilt
Make First Bracket
Make Second Bracket
Loading the Code

CHAPTER 9 RoboBoat
Some Words in Advance
Parts List for the RoboBoat
Polystyrene Foam
Epoxy Resin
Gloves
Fiberglass Fabric
Glue
Styrofoam Cutter and Razor Knife
Miscellaneous
The RoboBoat Design
Assembling the Boat
The Templates
Gluing the Templates on the EPS/XPS Board
Cutting Out the Segments
Gluing the Segments Together
Inserting the Foam Anchors
The Coating
Applying the Finish
The Fins
Painting
The Deck
Completing the Assembly
The Propulsion Assembly
The Baseplate
The Pivot
The Tube
The Rudder Horns
The Motor

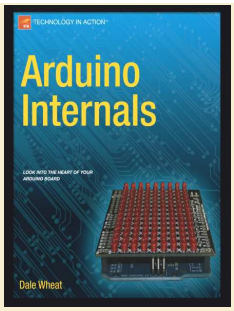
The Servo
The Pushrods
Electronics
The Heart of the System: The ArduPilot PCB
The GPS Module
The Electronic Speed Controller (ESC)
The Motor
The Rudder Servo
The Battery Pack
Assembling the Electronics
The Programming Adapter
Software and Mission Planning
GPS Receivers
The Software
AP_RoboBoat Module
Debug Module
Init Module
Navigation Module
PID_control Module
Servo_control Module
Header Files
Installing the Software
Installing the Arduino IDE
Configuring the Arduino IDE
Compiling and Uploading the Code
Customizing the Code
Customizing the Rudder Control
Customizing the PID Constants
Customizing the Motor Speed
Customizing the Waypoint Timeout
Customizing the Waypoint Radius
Customizing the Rudder Extents
Mission Planning
Employing Google Earth for the Coordinates
Putting It All Together
Integrating the System
Ships Ahoy!
Troubleshooting
The Thrust of the Motor/Propeller Is Not Sufficient
The Motor Does Not Start

CHAPTER 10 Lawn-Bot 400
How the Lawn-bot 400 Works
Lawn Mower Deck
High-Capacity Batteries
Steel Frame
Dump-Bucket
Pneumatic Tires
Headlights
Failsafe
Tools and Parts List
Lawnmower
Parts Lists
The Wheels
Front Caster Wheels
Rear Drive Wheels
Installing Sprockets
The Frame
The Drive-Train
Motor Mount Brackets
Installing the Chain
The Motor Controller
Buying a Motor Controller
Sabertooth 2x25
Cooling Fans
Motor Controller feedback
Current sensor IC
The Arduino
Securing Connections for a Bumpy Ride
Building a Breakout Board
The Failsafe
R/C Toggle Switch
Power Relays
Avoiding an R/C Failsafe
Making Connections
The Code
Adding Cosmetics and Accessories
Painting
Headlights
Dump Bucket
Lawnmower Kill-Switch

CHAPTER 11 The Seg-Bot
How the Seg-Bot Works
Inertial Measurement Unit
Steering and Gain
Engage Switch
Parts List for the Seg-Bot
Selecting the Right Sensors
3.3v Power
Accelerometer
Gyroscope
Cycle Time
Gyro Starting Point
Gyroscope Drift
Gyroscope Versus Accelerometer Summary
Filtering the Angle
Weighted Average
Making the IMU Adapter Board
Selecting the Motors
Electric Brake Removal
Motor Mounting Position
Selecting the Motor Controller
SoftwareSerial Library
Sabertooth Simplified Serial
The Batteries
Sealed Lead-Acid
Charging
12v Supply
The Frame
Frame Design
Building the Frame
Inputs
Steering
Gain
Engage Switch
Level-Start
Mounting the Inputs to the Frame
Installing the Electronics
Soldering the Inputs
Wiring the Connections
Reviewing the Code
Checking the Angle Readings
The read_pots() Function
The auto_level() Function
The update_motor_speed() Function
The time_stamp() Function
The serial_print_stuff() Function
The Full Code
Testing
Summary
References

CHAPTER 12 The Battle-Bot
Robotic Combat Is Born
Battle-Bot Rules and Regulations
No Price Limit!
Is This Going to Be Expensive?
Parts List for the Battle-Bot
Input Control
Fly Sky CT-6: The $32, 5-ch*, 2.4GHz Radio Alternative
Attention Hackers
The Electronics
The Arduino
The Motor Controllers
The Frame
To Buy or Build?
Modifying the Wheels
Building the Frame
The Drive Train
Gearing
Calculating Gear Ratio
Modifications
Chain Tensioning Nut
Threading Sequence
Measuring the Chain
Adding Spacers
Batteries
Securing Electronics
Protecting Your Brains
Making Connections
The Code
Armor
Weapons
Additional Information

CHAPTER 13 Alternate Control
Using Processing to Decode the Signals
Parts List for Alternate Control
Selecting the Input
Processing Prerequisites
Following Protocol
Examining the Processing Sketch
Explanation
Testing Processing
Examining the Arduino Sketch



Dale Wheat

Arduino Internals
Arduino изнутри

— Publisher: Apress, 2011. — 392 pages
— Series: Technology in Action

ISBN-10: 1430238828
ISBN-13: 978-1430238829

Автор делится своими глубокими знаниями плат Arduino. Книга проведет вас в самое сердце плат Arduino.

CONTENTS

About the Author
About the Technical Reviewers
Acknowledgments
Preface
Intended Audience
What This Book Isn't
Chapter Overview
Summary

Hardware
What Is an Arduino?
The Arduino Uno
Processor
Serial Port
Power Supply
Expansion Connectors
Shields
The Arduino Mega 2560
Previous Hardware
Arduino Serial
Arduino USB
Arduino Extreme
Arduino Nuova Generazione (New Generation)
Arduino Diecimila
Arduino Duemilanove
Arduino Mega
Who Makes Arduinos?
Officially Licensed Products
Everybody Else
Build Your Own
Arduino Printed Circuit Boards
Breadboard Arduinos
Summary

Software
Hosts and Targets
Step by Step
Step 1: Write Some Code
Step 2: Compile the Code
Step 3: Program the Device
Step 4: Test and Debug
Step 5. Repeat
Semiautomatic
Blinking in C
Going Further
Summary

Atmel AVR
Origins
AVR Device Families
When in Doubt: Product Datasheets
Device Packaging
Through-Hole DIPs
Surface-Mount Devices (SMDs)
Extra Pins
Pin Descriptions
Power Pins
AVR Core
Clock Sources
Address Spaces
Instruction Set
Internal Peripherals
General Purpose Input/Output (I/O)
External Interrupts
Timer/Counters
USART
Two-Wire Serial Interface (TWI), a.k.a. I2C
Analog Inputs
Summary

Supporting Hardware
Schematic Diagrams
Component Types
Reference Designators
Component Values
Component Value Tolerances
Other Component Parameters
The Connections
Getting Power to the Board
The Barrel Connector
Input Power Conditioning
Voltage Regulator
Power Circuit Evolution
Serial Interface
RS-232 Interface
The Processor
Power Consumption
I/O Drive Capability
The -RESET Signal
The Time Base
Decoupling Capacitors
Blinky Lights
Room for Expansion
The Mechanical Form Factor
Universal Serial Bus (USB): Signals Plus Power
Summary

Arduino Software
Open Source Software
Multiplatform Support
The Arduino Heritage
Installing the Software
The Process, or "How to Arduino"
A Tour of the User Interface
The File Menu
The Edit Menu and the Edit Context Menu
The Sketch Menu
The Tools Menu
The Help Menu
Summary

Optimizations
How Will You Know It Worked?
Shrink Blink
How Blink Works
Measuring Space-Saving Optimizations
Code Analysis
Life Without pinMode()
Abbr. & Shrtcts
Binary Notation
Further Analysis
Easy Toggling
Further Reduction
Wasting Time More Efficiently

Using Lower-Level Code
Saving Space with Simple Serial Communication
What "Hello, world!" Does
Writing to Configuration Registers
Transmitting Data
A String of Characters
Printing Numbers
Saving SRAM
Measuring SRAM Requirements
The Bare Minimum
Memory Sections
Where Variables Live
Using the Appropriate Data Type
Strings of Characters, Revisited
Low Power or High Speed?
Electronic Measurements
The Arduino as Test Equipment
As Fast As Possible
Slowing It Down
Further Power Reductions
Summary

Hardware Plus Software
Available Peripherals
Serial Port(s)
General-Purpose Digital Inputs and Outputs
Timers and Counters
Pulse-Width-Modulation (PWM) Outputs
Analog Inputs
External Interrupts
Interrupt Reference
Summary

Example Projects
Beyond the Blinking LED: Starting Simply
Slow Enough
Mostly Optimized Six-Channel Dimmer
It's Dim, Alright
Other Uses for a Blinking LED
Infrared Remote Control
TV-B-Gone
A Lot of Blinking LEDs
A Direct-Drive Example
Direct-Drive with LED Drivers
Multiplexing Techniques
A Digital Clock
Accuracy
User Interface
Additional Features
Summary

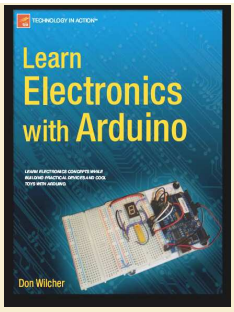
Project Management
Documentation
Source-Code Comments
Whitespace
Code What You Mean, Mean What You Code
Automated Documentation
Writing For Your Audience
Hardware Documentation
Going Further
Teamwork and Collaborative Development
Blogs
Forums
Wikis
Revision Control Systems
A Note About Revisionor Version-Numbering
Project-Hosting Web Sites
Licensing Your Work
Patents and Trademarks
Copyright
Open Source
The Public Domain
Summary

Hardware Design
Learning About Hardware
Things You Must Have
Things You Want
Infrared Proximity Sensor
A Modest Prototype
Some Modest Improvements
Printed Circuit Boards
PCB Layout Techniques
A First Attempt
A More Compact Version
Making the Connection
Your Own Custom Arduino
Compatibility with Existing Arduinos and Shields
Power Supply Options
Processor Selection
Anything Else?
Design Software
CadSoft EAGLE
EAGLE Tips
Summary

Software Design
Advanced Topics Within Arduino
Writing Arduino Libraries
Alternate Cores
And Without Arduino
The Bare Metal, Revisited
Other Development Environments
Summary

Networking
Point-to-Point Networking
Talking Over the Serial Port
Arduino to Arduino
MIDI: Musical Instrument Digital Interface
The Internet
Summary

More Example Projects
An Autonomous Robot
Power Supply
Motion Control
Electric Motors and Actuators
Sensors
Light Sensors
Touch Sensors
Noncontact Sensors
Audio Sensors
Indicators, Controls, and Other Forms of Communication
Control Systems
Open-Loop Systems
Closed-Loop Systems
Example Robot Projects
A Practice Robot
The Next Robot
Your Ultimate Robot



Don Wilcher

Learn Electronics with Arduino

Учитесь электронике с Arduino

— Publisher: Apress, 2012. — 280 pages
— Series: Technology in Action

ISBN-10: 1430242663
ISBN-13: 978-1430242666

С этой книгой вы научитесь электронике при построении забавных и практических устройств и гаджетов на основе Arduino.

CONTENTS

Chapter 1: Electronic Singing Bird
Parts List
What Is Physical Computing ?
How It Works
Pulse Width Modulation Basics
Transistor Basics
Transformer Action
The Voltage Divider
Light Detection Circuits with a Photocell
Testing the Light Detection Circuit with a Voltmeter and an Oscilloscope
Using a Voltmeter
Using an Oscilloscope
Assembly of the Electronic Singing Bird Circuit on a Breadboard
Creating the Interactive Control Software
What Is a Sketch ?
Final Testing of the Electronic Singing Bird
Further Discovery Methods

Chapter 2: Mini Digital Roulette Games
Parts List
How It Works
Forward Biasing a LED
LED Circuit Analysis
The LED Bar Display
Mini Roulette Game, Version 1
Adding the Game Software
The Seven-Segment LED Display Basics
Testing the Seven-Segment LED Display
Build an Arduino-based Seven Segment LED Display Flasher-Tester
The 7447 BCD-to-Decoder IC Basics
Build a BCD-to-Decimal Circuit with Seven Segment LED Display
Assembly of the Final Circuit on the Breadboard
Adding the Mini Digital Roulette Game Software
Final Testing of the Mini Digital Roulette Game
Further Discovery Method Suggestions

Chapter 3: An Interactive Light Sequencer Device
Parts List
Remix Revisited
How It Works
The Potentiometer
Measurement Setup Procedure
How to Drive Multiple LEDs with a Microcontroller
Building the Remixed Interactive LED Sequencer Device
Creating the Sequential-Switching Software
Final Testing of the Interactive Light Sequencer Device
Further Discovery Methods

Chapter 4: Physical Computing and DC Motor Control
Parts Lists
Remixing Revisited
How It Works
A Base Biasing Transistor Driver Circuit
D1: Flyback Diode
Experimenting with a Transistor Relay Driver DC Motor Control Circuit
Electromechanical Relay Preparation
The Basics of Physical Computing with Electric Motors
Achieving Motor Speed Control with Physical Computing
Potentiometer Input Control
The 2N2222 Transistor Pinout
The Motor Speed Control Software
Light Detection Input Control
Final Testing of the Devices
Further Discovery Methods

Chapter 5: Motion Control with an Arduino: Servo and Stepper Motor Controls
Parts List
Remixing Motion Controls
How It Works
Experimenting with a Servo Motor
Fritzing Software
Try It!
Physical Computing: A Servo Motor with a Potentiometer
Physical Computing: A Servo Motor with a Joystick
Physical Computing: A Servo Motor with a FlexiForce Sensor
Motion Control Basics
The Darlington Transistor
The Unipolar Stepper Motor
A Multisim Digital Controller Model for a Unipolar Stepper Motor
Build an Arduino Unipolar Stepper Motor Controller
Adding a Speed Control Function
Final Testing of the Servo and Stepper Motor Controllers
Further Discovery Method Suggestions

Chapter 6: The Music Box
Parts List
Remixing Physical-Computing and Driver Interface Circuits
How It Works
Experimenting with PWM
Building and Testing a Basic Music Box Controller
Try It!
Driving a Speaker
Physical Computing and the Music Box Controller
What Is a PMOSFET ?
A PMOSFET Multisim Circuit Model
Sketch for the Interactive Music Box Controller
Building and Testing a Basic Music Box Controller with a Keypad
Final Testing of the Music Box Controllers
Further Discovery Methods

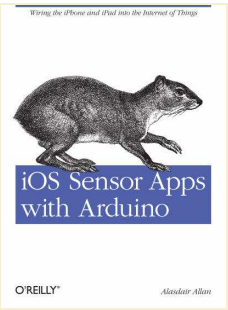
Chapter 7: Fun with Haptics
Parts List
Remixing Physical Computing and Driver Interface Circuits
How It Works
Experimenting with a Vibration Motor
Physical Computing: A Vibration Motor
Try It Out
Keypad Haptics
Mechatronics and Haptics
FlexiForce Sensor Haptics
A Robot End Effector Test Stand
Final Testing of Haptics Controllers
Further Discovery Method Suggestions

Chapter 8: LCDs and the Arduino
Parts List
Remixing Physical-Computing Input Interface Circuits
How It Works: The LCD Test Jig
The Real “Hello World”: Arduino and the LCD
Try It Out!

The Vanishing Message
Building an Evaluation Board
Further Discovery Methods

Chapter 9: A Logic Checker
Parts List
Input Interface Circuits
How It Works
Testing a NAND Gate
The Seven-Segment LED Display and the Arduino
Building a Smart Logic Probe
Building an Improved Smart Logic Probe
Further Discovery Methods

Chapter 10: Man, It's Hot: Temperature Measurement and Control
Parts List
What Is a Precision Centigrade Temperature Sensor ?
How It Works
Building an Electronic Thermometer
A Computer Thermometer
Final Completion of Computer Thermometer
Try It Out!
An LCD Electronic Thermometer
A Temperature Controller
Further Discovery Method
Final Thoughts and Suggestions



Alasdair Allan

iOS Sensor Apps with Arduino: Wiring the iPhone and iPad into the Internet of Things

Подключение датчиков к устройствам на iOS при помощи Arduino

— Publisher: O'Reilly Media, 2011. — 128 pages

ISBN-10: 1449308481
ISBN-13: 978-1449308483

Превратите свой iPhone или iPad в центр распределенной сети датчиков с помощью микроконтроллера Ардуино. С этим кратким руководством Вы изучите, как соединить датчики с устройством на iOS, и сделайте так, чтобы они обменивались друг с другом через Ардуино. Вы также создадите приложение для iOS, которое проанализирует значения, получаемые с датчиков, и графически изобразит результаты измерений в режиме реального времени.

CONTENTS

Preface

1. Introduction to the Arduino
The Arduino
Powering the Board
Input and Output
Communicating with the Board
Installing the Software
Connecting to the Board
Blinking an LED
Uploading the Sketch
Making a Serial Connection
Summary

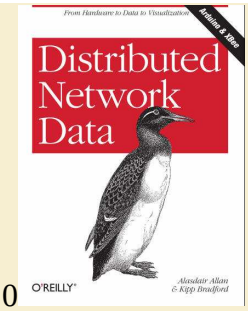
2. Connecting the iPhone to the Arduino
The Apple MFi Program
The Redpark Serial Cable
Testing the Cable
Connecting to the Arduino
Connecting to an iOS Device
A Simple Serial Application
Adding the Redpark Serial Library
Connecting the Arduino
Sending Data Back to the Arduino
Log Messages
Summary

3. Controlling the Arduino from the iPad
An Arduino on Your iPad
Adding the Serial Library
Building the User Interface
Integrating the Serial Library
Listening for Messages on the Arduino
Putting It All Together
Going Further
Adding a Log Window
Summary

4. Using External Sensors from the iPhone
The LV-MaxSonar-EZ1
Analog Output
Pulse Width Output
RS-232 Serial Output
MaxSonar Range Finder for iPhone
Adding the Serial Library
The CorePlot Library
Building the User Interface
Building the Backend
Writing the Arduino Sketch
Putting It All Together
Turning Things On and Off
Connecting Directly to the Cable

5. Connecting to an XBee Network
XBee Modules
Series 1 or Series 2?
Regular Versus Pro?
802.15.4 or ZigBee?
Which Aerial?
How to Configure an XBee Series 1 Radio
Connecting the XBee to Your Mac
XBee Addressing
Configuring Two XBee Radios
Connecting an XBee to an Arduino
Connecting an XBee to an iOS Device
XBee to RS-232 Serial
Going Further

6. Other Ways to Connect
Using the Network
Using Ethernet
Using WiFi
Using a Soft Modem
Switch Science Board
HiJack Board
Using the MIDI Protocol
The HIDDUINO
Summary



Alasdair Allan, Kipp Bradford

Distributed Network Data
Распределённая сеть передачи данных

— Publisher: O'Reilly Media, 2013. — 170 pages
— Series: Technology in Action

ISBN-10: 1449360262
ISBN-13: 978-1449360269

Постройте свою собственную распределенную сеть датчиков, чтобы собрать, проанализировать, и отобразить данные в реальном времени об окружающем нас мире – включая уровень шума, температуру, и т.п. С этой книгой Вы сможете превратить Вашу идею в рабочее устройство, используя легко изучаемую плату Arduino и стандартные датчики.

CONTENTS

Preface

1. Introduction to Arduino
Saying Hello in the Old World
The Arduino
The Board
Powering the Board
Input and Output
Communicating with the Board
Installing the Software
Connecting to the Board
Blinking an LED
Uploading the Sketch
Making a Serial Connection

2. Getting Started
The Breadboard
The Sensor
The DHT-22
Wiring the Breadboard
Writing the Software
The DHT Library
Arduino Sketch
Running the Software

3. Adding Another Sensor
The Sensor
PIR Sensor
Wiring the Breadboard
Modifying the Software
Running the Software

4. Finishing the Breadboard
The Sensor
Electret Microphone
Wiring the Breadboard
Modifying the Software
Running the Software
Adding Some LEDs
Modifying the Software
Running the Software
Making the Output Machine-Readable
Communicating with Python

5. Moving from Breadboard to Prototype
The Prototype
Fritzing
Installing the Software
Building a Fritzing Circuit
Cleaning up the Fritzing Diagram
Dropping Breadboard Crumbs
Making the Board Permanent
Solder
Soldering Irons
The Protoboard
Octopus Arms or Helping Hands
Let the Solder Flow

6. Simplifying the Design
Arduino Proto Shields
Fritzing Revisited
Power and Ground
Cleaning up the Fritzing Diagram
Rules for Pin Power and Ground
Saving Power

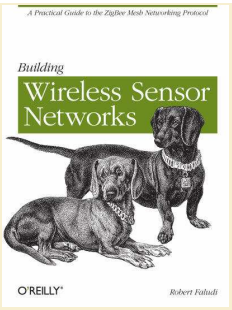
7. Building Point-to-Point XBee Networks
XBee Modules
Series 1 or Series 2?
Regular vs Pro?
802.15.4 or Zigbee?
Which Antenna?
How to Configure an XBee Series 1 Radio
Connecting the XBee to your Mac
XBee Addressing
Configuring Two XBee Radios
Connecting an XBee to an Arduino
Going Wireless with XBees

8. Building Many-to-Point XBee Networks
Addressing for Multiple XBees
Addressing the Arduino
Individual Call and Response
Switching to Mesh Networks

9. Visualizing with Processing
Processing
Installing the Software
Reading Data From a File
Reading Data Directly From the Serial Port
Plotting Temperature in Real Time

10. Visualizing with LabVIEW
LabVIEW
Installing the Software
Simple LabVIEW with Arduino
Graphing the Data

11. Going Further
Arduino
XBee Networking
Fritzing
EAGLE
Processing
ProcessingJS
LabVIEW
Data Visualization



Robert Faludi

Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing
Создание беспроводных сенсорных сетей:
с ZigBee, XBee, Ардуино и их обработкой

— Publisher: O'Reilly Media, 2011. — 322 pages

ISBN-10: 0596807732
ISBN-13: 978-0596807733

Книга научит создавать системы распределенных датчиков и интеллектуальные интерактивные устройства, используя протокол беспроводных сетей ZigBee и XBee. Изучив практическое руководство, вы создадите ряд полезных проектов, включая полную беспроводную сеть ZigBee, которая передаёт данные от удалённых датчиков.

CONTENTS

Preface

1. Getting Ready
Buying an XBee Radio
Hardware
Antennas
Vendors
Buying an Adapter
Digi Evaluation Board
USB Adapters
Breakout Boards
Arduino Board Adapter Hack
What Are All Those Pins?
Choosing a Terminal Program
Firmware Update Software
Terminal Software for Configuring Settings
Summary

2. Up and Running
Radio Basics
Electromagnetic Spectrum
Inverse Square Law
Introduction to ZigBee
Network Topology
Addressing Basics
PAN Addresses
Channels
All Together Now
XBee Firmware Updates
Reading Current Firmware and Configuration
Configuring XBee
Settings
Connecting from Windows
Connecting from Macintosh
Command Mode and Transparent Mode
AT Commands (Are Your Friend)
Using AT Commands
Basic ZigBee Chat
Parts
Addresses
Coordinator
Router
Two Computers
One Computer
Chat
Troubleshooting
Success!

3. Build a Better Doorbell
ZigBee and Arduino
About Arduino
Arduino Basics
Connecting to Arduino
Doorbell Projects
Parts
Prepare Your Radios
Connect Power from Arduino to Breadboard
XBee Breakout Board
XBee Connections
Doorbell Introduction
Switch Input...
...and Buzzer Output
Configure Your XBees
Program the Arduino Doorbell
Feedback Doorbell
Feedback Light
Program the Arduino Feedback Doorbell
Extra: Nap Doorbells and More

4. Ins and Outs
The Story of Data
Direct, Indirect, Subtext
I/O Concepts
Why XBee Direct?
XBee Direct Limitations
XBee I/O Features
AT Configuration I/O Commands
Advanced I/O Commands
Romantic Lighting Sensor
Basic Romantic Lighting Sensor
Prepare the Sensor Board
Prepare the Base Station
Romantic Lighting Sensor with Feedback
API Ahead

5. API and a Sensor Network
What's an API?
Protocols
Humans
Computers
XBee API Protocol
Start Delimiter
Length Bytes
Frame Data Bytes
Checksum
API Frame Types
AT Commands
AT Responses
ZigBee Transmit Request
ZigBee Transmit Status
ZigBee Receive Packet
I/O Data Sample Rx Indicator
Remote AT Command Request
Remote Command Response
Using What You Need
Libraries
Simple Sensor Network
Parts
Prepare Your Coordinator Radio
Prepare Your Router Radios
Prepare the Sensor Boards
Prepare the Base Station
Program the Base Station
Yay!

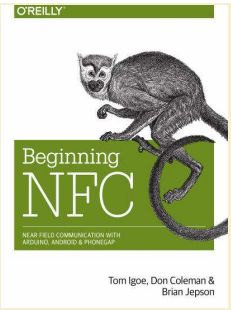
6. Sleeping, Then Changing the World
Sleep Mode
End Devices
Configuring Sleep
Easy Sleeping
Simple Sensor with Sleep Project
Parts
Prepare Your End Device Radios
Configure Your End Device XBees
Direct Actuation
Direct Actuation Example
Parts
Prepare Your Coordinator Radio
Prepare Your Router Radios
Prepare the Actuator Boards
Configure Your Router XBees
Prepare the Base Station
Simple Actuator Node Code in Processing
Summary

7. Over the Borders
Gateways
XBee as Embedded Gateway
Other Embedded Gateways
Internet Gateways
Internet Media
Computers Versus Dedicated Devices
ConnectPorts
Selecting a ConnectPort
Setting Up a ConnectPort
Configuring a ConnectPort
Remote Management
iDigi Connectivity Server
iDigi Features
Adding a ConnectPort
Viewing Configurations
Firmware Updates and Remote Reboot
Viewing an XBee Network
XBee Internet Gateway (XIG)
Installing and Configuring XIG
Testing XIG
XIG Example
Twitter Reader
Parts
Prepare Your ConnectPort with XBee Internet Gateway
Prepare Your Router Radio
Configure Your Router Radio
Prepare the Twitter Reader Board
Program the Arduino
Moving Forward

8. More to Love
Advanced ZigBee
ZigBee Stack Layers
Application Support Layer
Routing
Security
ZigBee Protocol References
Serial Flow Control
RTS and CTS
Sharing Data
Pachube
Simple Sensor Network with Pachube
API Key
Build the Simple Sensor Network in Chapter 5
Program the Base Station
Simple Sensor Network Pachube Code in Processing
The Future of ZigBee
Next Steps for You
Making Stuff
Sharing Your Work

Appendix: Resource Guide

Index



Tom Igoe, Don Coleman

Beginning NFC: Near Field Communication with Arduino, Android, and PhoneGap
Введение в NFC:
Near Field Communication для Arduino, Android и PhoneGap

— Publisher: O'Reilly Media, 2014. — 246 pages

ISBN-10: 1449372066
ISBN-13: 978-1449372064

Войдите в мир NFC (беспроводная высокочастотная связь малого радиуса действия) – быстро-развивающаяся технология, которая позволяет устройствам в непосредственной близости обмениваться данными, используя радиосигналы. С большим количеством примеров (примеры кода, упражнений и проектов) это практическое руководство покажет вам, как создать приложения NFC для Android, Arduino и устройств на Linux.

CONTENTS

1. Introduction
Who This Book Is For
Recommended Reading
What's Covered in This Book
What You'll Need
Hardware
Software
Other Useful NFC Apps
Conventions Used in This Book
Using Code Examples
Safari® Books Online
How to Contact Us
Acknowledgments

2. NFC and RFID
What's RFID?
What's NFC?
How RFID Operates
RFID Standards
How NFC Operates
NFC Data Exchange Format (NDEF)
The Architecture of NFC
NFC Tag Types
Where to Get Tags
Device-to-Tag Type Matching
What You Can Do with NFC
Conclusion

3. Getting Started with PhoneGap and the PhoneGap-NFC Library
Why Android?
Hello, World! Your First PhoneGap App
Setting Up the Development Environment
Creating a PhoneGap Project
The Important Files
A Simple Locator App
PhoneGap Meets NFC: NFC Reader
Installing the NFC Plug-In
Writing the NFC Reader App
Troubleshooting
Conclusion

4. Introducing NDEF
NDEF Structure
Type Name Format
Payload Type
Payload Identifier
Payload
Record Layout
Record Header
How Big Can an NDEF Message Be?
Record Chunking
Additional Info
NDEF in Practice
A Tag Writer Application: Foursquare Check-In
Writing an NDEF Record to a Tag
Writing Different Record Types
PhoneGap-NFC NDEF Helper Functions, Summarized
Conclusion

5. Listening for NDEF Messages
PhoneGap-NFC's Event Listeners
An NDEF Reader Application
Listening for Multiple Events
Reading the NDEF Messages
Filtering Tags Using Record Types
Filtering Using MIME-Types
Android's Tag Dispatch System
Types of Intents
Android Application Records
Conclusion

6. An NFC Application in Practice
The User Interaction
Getting to Know Hue, Getting to Know All About Hue
The Hue Data Format
The Hue's REST API
The Android Shell
The PhoneGap Media API
The User Interface
The Application Code
Housekeeping Functions
Global Event Handlers
Hub Communication Functions
Lighting User Interface Event Handlers
Music User Interface Event Handlers
NFC Event Handlers
Enabling Background Dispatch
Conclusion

7. Introduction to Arduino and NFC
Digital Meets Physical: Arduino
The Hardware Heart of NFC
The Arduino Development Environment
Serial Communication
Installing Arduino Libraries
The Arduino NDEF Library
Reading NDEF in Arduino
Writing NDEF in Arduino
A Microcontroller NFC Application: Hotel Key Cards
The Interaction and Data Format
The Arduino NDEF Writer Device
The Arduino NDEF Reader and Door Lock Device
A Browser Interface for the Arduino NDEF Writer Device
Node.js Application Specification: package.json
The Client-Side Code
The Server-Side Code
Conclusion

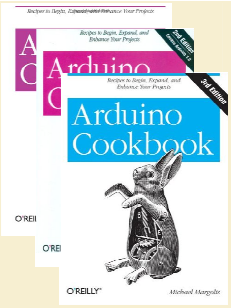
8. Peer-to-Peer Exchange
Sending Peer-to-Peer Messages in PhoneGap
Receiving Peer-to-Peer Messages in PhoneGap
Handover
Static Handover

Sending Handover Messages in PhoneGap
Peer-to-Peer Using Arduino
Card Emulation
Conclusion

9. NFC on Embedded Systems
Introduction to Embedded Linux Devices and Package Managers
Network, USB, and NFC
NFC on Embedded Linux: The Overview
Housekeeping Details
Get Comfortable with Your Editing Workflow
Know Your Package Manager
Set the Date and Time
Make a Downloads Directory
BeagleBone Peculiarities
Raspberry Pi Peculiarities
Installing the Tools for NFC
Installing the Libusb Compatibility Library
Installing libnfc
Testing the Installation on Raspberry Pi or BeagleBone
Installing libfreefare
Libnfc and Libfreefare Command Line Tools
NDEF Reading and Writing in Node.js
Web Interface for Tag Writer
Tags Controlling Physical Output
BeagleBone Version
Raspberry Pi Version
Conclusion

A. NFC Specification Codes

Index



Michael Margolis

Arduino Cookbook

Справочник по Arduino

— Publisher: O'Reilly Media
— 1 edition (03 31, 2011) — 662 p. ISBN-10: 0596802471, 978-0596802479
— 2 edition (12 30, 2011) — 724 p. ISBN-10: 1449313876, 978-1449313876
— 3 edition (07 22, 2014) — 800 p. ISBN-10: 1449371965, 978-1449371968

Хотите создавать устройства, которые взаимодействуют с физическим миром? Эта книга идеально подходит для тех, кто хочет экспериментировать с популярной платформой Arduino. Вы найдете более 200 советов и приемов для построения различных устройств, таких как игрушки, детекторы, роботы и др.

CONTENTS

Who This Book Is For
How This Book Is Organized
What Was Left Out
Code Style (About the Code)
Arduino Platform Release Notes
Conventions Used in This Book
Using Code Examples
Safari® Books Online
How to Contact Us
Acknowledgments
Notes on the Second Edition

Chapter 1. Getting Started
1.0 Introduction
Arduino Software
Arduino Hardware
See Also
1.1 Installing the Integrated Development Environment (IDE)
Problem, Solution, Discussion, See Also
1.2 Setting Up the Arduino Board
Problem, Solution, Discussion, See Also
1.3 Using the Integrated Development Environment (IDE) to Prepare an Arduino Sketch
Problem, Solution, Discussion, See Also
1.4 Uploading and Running the Blink Sketch
Problem, Solution, Discussion, See Also
1.5 Creating and Saving a Sketch
Problem, Solution, Discussion, See Also
1.6 Using Arduino
Problem, Solution, Discussion, See Also

Chapter 2. Making the Sketch Do Your Bidding
2.0 Introduction
2.1 Structuring an Arduino Program
Problem, Solution, Discussion, See Also
2.2 Using Simple Primitive Types (Variables)
Problem, Solution, Discussion, See Also
2.3 Using Floating-Point Numbers
Problem, Solution, Discussion, See Also
2.4 Working with Groups of Values
Problem, Solution, Discussion, See Also
2.5 Using Arduino String Functionality
Problem, Solution, Discussion
Choosing between Arduino Strings and C character arrays
See Also
2.6 Using C Character Strings
Problem, Solution, Discussion, See Also
2.7 Splitting Comma-Separated Text into Groups
Problem, Solution, Discussion, See Also
2.8 Converting a Number to a String
Problem, Solution, Discussion
2.9 Converting a String to a Number
Problem, Solution, Discussion, See Also
2.10 Structuring Your Code into Functional Blocks
Problem, Solution, Discussion, See Also
2.11 Returning More Than One Value from a Function
Problem, Solution, Discussion
2.12 Taking Actions Based on Conditions
Problem, Solution, Discussion, See Also
2.13 Repeating a Sequence of Statements
Problem, Solution, Discussion, See Also
2.14 Repeating Statements with a Counter
Problem, Solution, Discussion, See Also
2.15 Breaking Out of Loops
Problem, Solution, Discussion, See Also
2.16 Taking a Variety of Actions Based on a Single Variable
Problem, Solution, Discussion, See Also
2.17 Comparing Character and Numeric Values
Problem, Solution, Discussion, See Also
2.18 Comparing Strings
Problem, Solution, Discussion, See Also
2.19 Performing Logical Comparisons
Problem, Solution, Discussion
2.20 Performing Bitwise Operations
Problem, Solution, Discussion, See Also
2.21 Combining Operations and Assignment
Problem, Solution, Discussion, See Also

Chapter 3. Using Mathematical Operators
3.0 Introduction
3.1 Adding, Subtracting, Multiplying, and Dividing
Problem, Solution, Discussion, See Also
3.2 Incrementing and Decrementing Values
Problem, Solution, Discussion, See Also
3.3 Finding the Remainder After Dividing Two Values
Problem, Solution, Discussion, See Also
3.4 Determining the Absolute Value
Problem, Solution, Discussion, See Also
3.5 Constraining a Number to a Range of Values
Problem, Solution, Discussion, See Also
3.6 Finding the Minimum or Maximum of Some Values
Problem, Solution, Discussion, See Also
3.7 Raising a Number to a Power
Problem, Solution, Discussion
3.8 Taking the Square Root
Problem, Solution, Discussion
3.9 Rounding Floating-Point Numbers Up and Down
Problem, Solution, Discussion
3.10 Using Trigonometric Functions
Problem, Solution, Discussion, See Also
3.11 Generating Random Numbers
Problem, Solution, Discussion, See Also
3.12 Setting and Reading Bits
Problem, Solution, Discussion, See Also
3.13 Shifting Bits
Problem, Solution, Discussion, See Also
3.14 Extracting High and Low Bytes in an int or long
Problem, Solution, Discussion, See Also
3.15 Forming an int or long from High and Low Bytes
Problem, Solution, Discussion, See Also

Chapter 4. Serial Communications
4.0 Introduction
Serial Hardware
Software Serial
Serial Message Protocol
New in Arduino 1.0
See Also
4.1 Sending Debug Information from Arduino to Your Computer
Problem, Solution, Discussion, See Also
4.2 Sending Formatted Text and Numeric Data from Arduino
Problem, Solution, Discussion, See Also

4.3 Receiving Serial Data in Arduino
Problem, Solution, Discussion, See Also
4.4 Sending Multiple Text Fields from Arduino in a Single Message
Problem, Solution, Discussion, See Also
4.5 Receiving Multiple Text Fields in a Single Message in Arduino
Problem, Solution, Discussion, See Also
4.6 Sending Binary Data from Arduino
Problem, Solution, Discussion, See Also
4.7 Receiving Binary Data from Arduino on a Computer
Problem, Solution, Discussion, See Also
4.8 Sending Binary Values from Processing to Arduino
Problem, Solution, Discussion
4.9 Sending the Value of Multiple Arduino Pins
Problem, Solution, Discussion, See Also
4.10 How to Move the Mouse Cursor on a PC or Mac
Problem, Solution, Discussion, See Also
4.11 Controlling Google Earth Using Arduino
Problem, Solution, Discussion, See Also
4.12 Logging Arduino Data to a File on Your Computer
Problem, Solution, Discussion, See Also
4.13 Sending Data to Two Serial Devices at the Same Time
Problem, Solution, Discussion, See Also
4.14 Receiving Serial Data from Two Devices at the Same Time
Problem, Solution, Discussion
Receiving data from multiple SoftwareSerial ports
4.15 Setting Up Processing on Your Computer to Send and Receive Serial Data
Problem
Solution

Chapter 5. Simple Digital and Analog Input
5.0 Introduction
5.1 Using a Switch
Problem, Solution, Discussion, See Also
5.2 Using a Switch Without External Resistors
Problem, Solution, Discussion
5.3 Reliably Detecting the Closing of a Switch
Problem, Solution, Discussion, See Also
5.4 Determining How Long a Switch Is Pressed
Problem, Solution, Discussion
5.5 Reading a Keypad
Problem, Solution, Discussion, See Also
5.6 Reading Analog Values
Problem, Solution, Discussion, See Also
5.7 Changing the Range of Values
Problem, Solution, Discussion, See Also
5.8 Reading More Than Six Analog Inputs
Problem, Solution, Discussion, See Also
5.9 Displaying Voltages Up to 5V
Problem, Solution, Discussion
5.10 Responding to Changes in Voltage
Problem, Solution, Discussion
5.11 Measuring Voltages More Than 5V (Voltage Dividers)
Problem, Solution, Discussion

Chapter 6. Getting Input from Sensors
6.0 Introduction
See Also
6.1 Detecting Movement
Problem, Solution, Discussion, See Also
6.2 Detecting Light
Problem, Solution, Discussion, See Also
6.3 Detecting Motion (Integrating Passive Infrared Detectors)
Problem, Solution, Discussion
6.4 Measuring Distance
Problem, Solution, Discussion, See Also
6.5 Measuring Distance Accurately
Problem, Solution, Discussion, See Also
6.6 Detecting Vibration
Problem, Solution, Discussion
6.7 Detecting Sound
Problem, Solution, Discussion
6.8 Measuring Temperature
Problem, Solution, Discussion, See Also
6.9 Reading RFID Tags
Problem, Solution, Discussion
6.10 Tracking Rotary Movement
Problem, Solution, Discussion
6.11 Tracking the Movement of More Than One Rotary Encoder
Problem, Solution, Discussion
6.12 Tracking Rotary Movement in a Busy Sketch
Problem, Solution, Discussion
6.13 Using a Mouse
Problem, Solution, Discussion, See Also
6.14 Getting Location from a GPS
Problem, Solution, Discussion, See Also
6.15 Detecting Rotation Using a Gyroscope
Problem, Solution, Discussion
Using the older LISY300AL gyro
Measuring rotation in three dimensions using the ITG-3200 sensor
See Also
6.16 Detecting Direction
Problem, Solution, Discussion
6.17 Getting Input from a Game Control Pad (PlayStation)
Problem, Solution, Discussion, See Also
6.18 Reading Acceleration
Problem, Solution, Discussion, See Also

Chapter 7. Visual Output
7.0 Introduction
Digital Output
Analog Output
Controlling Light
LED specifications
Multiplexing
Maximum pin current
7.1 Connecting and Using LEDs
Problem, Solution, Discussion, See Also
7.2 Adjusting the Brightness of an LED
Problem, Solution, Discussion, See Also
7.3 Driving High-Power LEDs
Problem, Solution, Discussion
How to Exceed 40 mA per Pin
See Also
7.4 Adjusting the Color of an LED
Problem, Solution, Discussion, See Also
7.5 Sequencing Multiple LEDs: Creating a Bar Graph
Problem, Solution, Discussion, See Also
7.6 Sequencing Multiple LEDs: Making a Chase Sequence (Knight Rider)
Problem, Solution, Discussion
7.7 Controlling an LED Matrix Using Multiplexing
Problem, Solution, Discussion
7.8 Displaying Images on an LED Matrix
Problem, Solution, Discussion, See Also
7.9 Controlling a Matrix of LEDs: Charlieplexing
Problem, Solution, Discussion, See Also
7.10 Driving a 7-Segment LED Display
Problem, Solution, Discussion
7.11 Driving Multidigit, 7-Segment LED Displays: Multiplexing
Problem, Solution, Discussion
7.12 Driving Multidigit, 7-Segment LED Displays Using MAX7221 Shift Registers
Problem
Solution
Solution
7.13 Controlling an Array of LEDs by Using MAX72xx Shift Registers
Problem, Solution, Discussion, See Also
7.14 Increasing the Number of Analog Outputs Using PWM Extender Chips (TLC5940)
Problem, Solution, Discussion, See Also
7.15 Using an Analog Panel Meter as a Display
Problem, Solution, Discussion, See Also

Chapter 8. Physical Output
8.0 Introduction
Motion Control Using Servos
Solenoids and Relays
Brushed and Brushless Motors
Stepper Motors
Troubleshooting Motors
8.1 Controlling the Position of a Servo
Problem, Solution, Discussion
8.2 Controlling One or Two Servos with a Potentiometer or Sensor
Problem, Solution, Discussion
8.3 Controlling the Speed of Continuous Rotation Servos
Problem, Solution, Discussion
8.4 Controlling Servos Using Computer Commands
Problem, Solution, Discussion, See Also
8.5 Driving a Brushless Motor (Using a Hobby Speed Controller)
Problem, Solution, Discussion
8.6 Controlling Solenoids and Relays
Problem, Solution, Discussion
8.7 Making an Object Vibrate
Problem, Solution, Discussion
8.8 Driving a Brushed Motor Using a Transistor
Problem, Solution, Discussion
8.9 Controlling the Direction of a Brushed Motor with an H-Bridge
Problem, Solution, Discussion

8.10 Controlling the Direction and Speed of a Brushed Motor with an H-Bridge
Problem, Solution, Discussion
8.11 Using Sensors to Control the Direction and Speed of Brushed Motors (L293 H-Bridge)
Problem, Solution, Discussion, See Also
8.12 Driving a Bipolar Stepper Motor
Problem, Solution, Discussion, See Also
8.13 Driving a Bipolar Stepper Motor (Using the EasyDriver Board)
Problem, Solution, Discussion
8.14 Driving a Unipolar Stepper Motor (ULN2003A)
Problem, Solution, Discussion

Chapter 9. Audio Output
9.0 Introduction
9.1 Playing Tones
Problem
Solution
See Also
9.2 Playing a Simple Melody
Problem
Solution
9.3 Generating More Than One Simultaneous Tone
Problem, Solution, Discussion
9.4 Generating Audio Tones and Fading an LED
Problem, Solution, Discussion, See Also
9.5 Playing a WAV File
Problem, Solution, Discussion, See Also
9.6 Controlling MIDI
Problem, Solution, Discussion, See Also
9.7 Making an Audio Synthesizer
Problem, Solution, Discussion, See Also

Chapter 10. Remotely Controlling External Devices
10.0 Introduction
10.1 Responding to an Infrared Remote Control
Problem, Solution, Discussion
10.2 Decoding Infrared Remote Control Signals
Problem, Solution, Discussion, See Also
10.3 Imitating Remote Control Signals
Problem, Solution, Discussion, See Also
10.4 Controlling a Digital Camera
Problem, Solution, Discussion, See Also
10.5 Controlling AC Devices by Hacking a Remote-Controlled Switch
Problem, Solution, Discussion, See Also

Chapter 11. Using Displays
11.0 Introduction
11.1 Connecting and Using a Text LCD Display
Problem, Solution, Discussion, See Also
11.2 Formatting Text
Problem, Solution, Discussion, See Also
11.3 Turning the Cursor and Display On or Off
Problem, Solution, Discussion
11.4 Scrolling Text
Problem, Solution, Discussion
11.5 Displaying Special Symbols
Problem, Solution, Discussion, See Also
11.6 Creating Custom Characters
Problem, Solution, Discussion
11.7 Displaying Symbols Larger Than a Single Character
Problem, Solution, Discussion, See Also
11.8 Displaying Pixels Smaller Than a Single Character
Problem, Solution, Discussion
11.9 Connecting and Using a Graphical LCD Display
Problem, Solution, Discussion
11.10 Creating Bitmaps for Use with a Graphical Display
Problem
Solution
See Also
11.11 Displaying Text on a TV
Problem, Solution, Discussion, See Also

Chapter 12. Using Time and Dates
12.0 Introduction
12.1 Creating Delays
Problem, Solution, Discussion, See Also
12.2 Using millis to Determine Duration
Problem, Solution, Discussion, See Also
12.3 More Precisely Measuring the Duration of a Pulse
Problem, Solution, Discussion, See Also
12.4 Using Arduino as a Clock
Problem, Solution, Discussion, See Also
12.5 Creating an Alarm to Periodically Call a Function
Problem, Solution, Discussion
12.6 Using a Real-Time Clock
Problem, Solution, Discussion, See Also

Chapter 13. Communicating Using I2C and SPI
13.0 Introduction
I2C
Migrating Wire code to Arduino 1.0
Using 3.3 Volt Devices with 5 Volt Boards
SPI
See Also
13.1 Controlling an RGB LED Using the BlinkM Module
Problem, Solution, Discussion, See Also
13.2 Using the Wii Nunchuck Accelerometer
Problem, Solution, Discussion, See Also
13.3 Interfacing to an External Real-Time Clock
Problem
Solution
See Also
13.4 Adding External EEPROM Memory
Problem, Solution, Discussion, See Also
13.5 Reading Temperature with a Digital Thermometer
Problem, Solution, Discussion, See Also
13.6 Driving Four 7-Segment LEDs Using Only Two Wires
Problem, Solution, Discussion, See Also
13.7 Integrating an I2C Port Expander
Problem, Solution, Discussion, See Also
13.8 Driving Multidigit, 7-Segment Displays Using SPI
Problem, Solution, Discussion
13.9 Communicating Between Two or More Arduino Boards
Problem, Solution, Discussion, See Also

Chapter 14. Wireless Communication
14.0 Introduction
14.1 Sending Messages Using Low-Cost Wireless Modules
Problem, Solution, Discussion, See Also
14.2 Connecting Arduino to a ZigBee or 802.15.4 Network
Problem, Solution, Discussion
Series 2 configuration
Series 1 configuration
Talking to the Arduino
See Also
14.3 Sending a Message to a Particular XBee
Problem, Solution, Discussion, See Also
14.4 Sending Sensor Data Between XBees
Problem, Solution, Discussion
Series 2 XBees
Series 1 XBees
See Also
14.5 Activating an Actuator Connected to an XBee
Problem, Solution, Discussion
Series 2 XBees
Series 1 XBees
See Also
14.6 Sending Messages Using Low-Cost Transceivers
Problem, Solution, Discussion, See Also
14.7 Communicating with Bluetooth Devices
Problem, Solution, Discussion, See Also

Chapter 15. Ethernet and Networking
15.0 Introduction
Arduino 1.0 Enhancements
Alternative Hardware for Low Cost Networking
15.1 Setting Up the Ethernet Shield
Problem, Solution, Discussion, See Also
15.2 Obtaining Your IP Address Automatically
Problem, Solution, Discussion
15.3 Resolving Hostnames to IP Addresses (DNS)
Problem, Solution, Discussion
15.4 Requesting Data from a Web Server
Problem, Solution, Discussion
15.5 Requesting Data from a Web Server Using XML
Problem
Solution
15.6 Setting Up an Arduino to Be a Web Server
Problem, Solution, Discussion
15.7 Handling Incoming Web Requests
Problem, Solution, Discussion
15.8 Handling Incoming Requests for Specific Pages
Problem, Solution, Discussion
15.9 Using HTML to Format Web Server Responses
Problem, Solution, Discussion, See Also
15.10 Serving Web Pages Using Forms (POST)
Problem, Solution, Discussion
15.11 Serving Web Pages Containing Large Amounts of Data
Problem, Solution, Discussion, See Also
15.12 Sending Twitter Messages
Problem, Solution, Discussion, See Also
15.13 Sending and Receiving Simple Messages (UDP)
Problem, Solution, Discussion
15.14 Getting the Time from an Internet Time Server
Problem, Solution, Discussion, See Also

15.15 Monitoring Pachube Feeds
Problem, Solution, Discussion, See Also
15.16 Sending Information to Pachube
Problem, Solution, Discussion

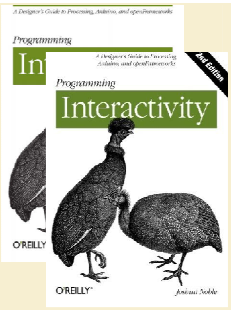
Chapter 16. Using, Modifying, and Creating Libraries
16.0 Introduction
16.1 Using the Built-in Libraries
Problem, Solution, Discussion, See Also
16.2 Installing Third-Party Libraries
Problem, Solution, Discussion
16.3 Modifying a Library
Problem, Solution, Discussion, See Also
16.4 Creating Your Own Library
Problem, Solution, Discussion, See Also
16.5 Creating a Library That Uses Other Libraries
Problem, Solution, Discussion
16.6 Updating Third-Party Libraries for Arduino 1.0
Problem, Solution, Discussion, See Also

Chapter 17. Advanced Coding and Memory Handling
17.0 Introduction
Preprocessor
See Also
17.1 Understanding the Arduino Build Process
Problem, Solution, Discussion, See Also
17.2 Determining the Amount of Free and Used RAM
Problem, Solution, Discussion, See Also
17.3 Storing and Retrieving Numeric Values in Program Memory
Problem, Solution, Discussion, See Also
17.4 Storing and Retrieving Strings in Program Memory
Problem, Solution, Discussion, See Also
17.5 Using #define and const Instead of Integers
Problem, Solution, Discussion, See Also
17.6 Using Conditional Compilations
Problem, Solution, Discussion, See Also

Chapter 18. Using the Controller Chip Hardware
18.0 Introduction
Registers
Interrupts
Timers
Analog and Digital Pins
See Also
18.1 Storing Data in Permanent EEPROM Memory
Problem, Solution, Discussion, See Also
18.2 Using Hardware Interrupts
Problem, Solution, Discussion, See Also
18.3 Setting Timer Duration
Problem, Solution, Discussion, See Also
18.4 Setting Timer Pulse Width and Duration
Problem, Solution, Discussion, See Also
18.5 Creating a Pulse Generator
Problem, Solution, Discussion, See Also
18.6 Changing a Timer's PWM Frequency
Problem, Solution, Discussion, See Also
18.7 Counting Pulses
Problem, Solution, Discussion, See Also
18.8 Measuring Pulses More Accurately
Problem, Solution, Discussion, See Also
18.9 Measuring Analog Values Quickly
Problem, Solution, Discussion, See Also
18.10 Reducing Battery Drain
Problem, Solution, Discussion, See Also
18.11 Setting Digital Pins Quickly
Problem, Solution, Discussion
18.12 Uploading Sketches Using a Programmer
Problem, Solution, Discussion, See Also
18.13 Replacing the Arduino Bootloader
Problem, Solution, Discussion, See Also
18.14 Reprogram the Uno to Emulate a Native USB device
Problem, Solution, Discussion, See Also

Appendix A. Electronic Components
Capacitor
Diode
Integrated Circuit
Keypad
LED
Motor (DC)
Optocoupler
Photocell (Photoresistor)
Piezo
Pot (Potentiometer)
Relay
Resistor
Solenoid
Speaker
Stepper Motor
Switch
Transistor
See Also

Appendix B. Using Schematic Diagrams and Data Sheets
How to Read a Data Sheet
Choosing and Using Transistors for Switching
Appendix C. Building and Connecting the Circuit
Using a Breadboard
Connecting and Using External Power Supplies and Batteries
Using Capacitors for Decoupling
Using Snubber Diodes with Inductive Loads
Working with AC Line Voltages
Appendix D. Tips on Troubleshooting Software Problems
Code That Won't Compile
Code That Compiles but Does Not Work as Expected
Appendix E. Tips on Troubleshooting Hardware Problems
Still Stuck?
Appendix F. Digital and Analog Pins
Appendix G. ASCII and Extended Character Sets
Appendix H. Migrating to Arduino 1.0
Migrating Print Statements
Migrating Wire (I2C) Statements
Migrating Ethernet Statements
Migrating Libraries
New Stream Parsing Functions



Joshua Noble

Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks

Интерактивное программирование: руководство разработчика по проектированию Arduino и openFrameworks

— Publisher: O'Reilly Media,
— 1 edition 2009. — 736 pages ISBN-10: 0596154143, ISBN-13: 978-0596154141
— 2 edition 2012. — 728 pages ISBN-10: 144931144X, ISBN-13: 978-1449311445

Всё готово для проведения насыщенных интерактивных экспериментов с вашими разработками, проектами или прототипами? Это идеальное место для начала. С этим руководством вы будете исследовать несколько тем в интерактивном искусстве и дизайне, в том числе: 3D-графику, звук, физическое взаимодействие, компьютерное зрение и геолокацию; узнаете основы программирования и электроники, необходимые для их реализации.

CONTENTS

Preface

PART I. INTRODUCTIONS

1. Introducing Interaction Design
What This Book Is for
Programming for Interactivity
The Nature of Interaction
Messages and Interaction
Interfaces and Interaction
Languages of Interaction
Design and Interaction
Art and Interaction
Data Exchange and Exploration
Working Process

2. Programming Basics
Why You'll Read This Chapter More Than Once
The Nature of Code
Variables
Simple Types
Arrays
Casting
Operators
Control Statements
if/then
for Loop
while Loop
continue
break
Functions
Defining a Function
Passing Parameters to a Method
Some Suggestions on Writing Functions
Overloading Functions
Objects and Properties
Scope
Review

3. Processing
Downloading and Installing Processing
Exploring the Processing IDE
The Basics of a Processing Application
The setup() Method
The draw() Method
The Basics of Drawing with Processing
The rect(), ellipse(), and line() Methods
RGB Versus Hexadecimal
The fill() Method
The background() Method
The line() Method
The stroke() and strokeWeight() Methods
The curve() Method
The vertex() and curveVertex() Methods
Capturing Simple User Interaction
The mouseX and mouseY Variables
The mousePressed() Method
The mouseReleased() and mouseDragged() Methods
The keyPressed and key Variables
Importing Libraries
Downloading Libraries
Loading Things into Processing
Loading and Displaying Images
Displaying Videos in the Processing Environment
Using the Movie Class
Reading and Writing Files
Running and Debugging Applications
Exporting Processing Applications
Conclusion
Review

4. Arduino
Starting with Arduino
Installing the IDE
Configuring the IDE
Touring Two Arduino Boards
The Controller
Duemilanove Versus Mini
Touring the Arduino IDE
The Basics of an Arduino Application
The setup Statement
The loop Method
Features of the Arduino Language
Constants
Methods
Arrays
Strings
How to Connect Things to Your Board
Hello World
Debugging Your Application
Importing Libraries
Running Your Code
Running Your Board Without a USB Connection
Review

5. Programming Revisited
Object-Oriented Programming
Classes
The Basics of a Class
Class Rules
Public and Private Properties
Inheritance
Processing: Classes and Files
C++: Classes and Files
.cpp and .h
A Simple C++ Application
Pointers and References
Reference
Pointer
When to Use Pointers
Large Data Objects
Pointers and Arrays
When Are You Going to Use This?
Review

6. openFrameworks
Your IDE and Computer
Windows
Mac OS X
Linux
Taking Another Quick Tour of C++
Basic Variable Types
Arrays
Methods
Classes and Objects in C++
Getting Started with oF
Touring an oF Application
Methods
Variables
Creating "Hello, World"
Drawing in 2D
Setting Drawing Modes
Drawing Polygons
Displaying Video Files and Images
Images
Video
Importing Libraries
ofxOpenCv
ofxVectorGraphics
ofxVectorMath
ofxNetwork
ofxOsc
Compiling an oF Program
Compiling in Xcode
Compiling in Code::Blocks
Debugging an oF Application
Using the printf Statement
Using the GNU Debugger
Using the Debugger in Xcode
Using the Debugger in Code::Blocks
Review

PART II. THEMES

7. Sound and Audio
Sound As Feedback
Sound and Interaction
How Sound Works on a Computer
Audio in Processing
Instantiating the Minim Library
Generating Sounds with Minim
Filtering Sounds with Minim
Sound in openFrameworks
openFrameworks and the FMOD Ex Library
The Sound Object Library
The Magic of the Fast Fourier Transform
Physical Manipulation of Sound with Arduino
A Quick Note on PWM
Creating Interactions with Sound
Further Resources
Review

8. Physical Input
Interacting with Physical Controls
Thinking About Kinetics
Getting Gear for This Chapter
Controlling Controls
The Button As an Electrical Object
The Button As an Interactive Object
The Button As a Value in Code
Turning Knobs
The Dial As an Interactive Object
Potentiometers
Using Lights
Wiring an LED
Detecting Touch and Vibration
Reading a Piezo Sensor
Getting Piezo Sensors
Communicating with Other Applications
Sending Messages from the Arduino
openFrameworks
Detecting Motion
PIR Motion Sensor
Reading Distance
Reading Input from an Infrared Sensor
Understanding Binary Numbers
Binary Numbers
Bits and Bit Operations
Why Do You Need to Know Any of This?
Detecting Forces and Tilt
Introducing I2C
What Is a Physical Interface?
What's Next
Review

9. Programming Graphics
The Screen and Graphics
Seeing Is Thinking, Looking Is Reading
Math, Graphics, and Coordinate Systems
Drawing Strategies
Use Loops to Draw
Use Arrays to Draw
Draw Only What You Need
Use Sprites
Processing and Transformation Matrices
Creating Motion
Shaping the Gaze
Setting the Mood
Creating Tweens
Using Vectors
Using Graphical Controls
ControlP5 Library
Event Handling
Importing and Exporting Graphics
Using PostScript in Processing
Using PostScript Files in oF
What's Next
Review

10. Bitmaps and Pixels
Using Pixels As Data
Using Pixels and Bitmaps As Input
Providing Feedback with Bitmaps
Looping Through Pixels
Manipulating Bitmaps
Manipulating Color Bytes
Using Convolution in Full Color
Analyzing Bitmaps in oF
Analyzing Color
Analyzing Brightness
Detecting Motion
Using Edge Detection
Using Pixel Data
Using Textures
Textures in oF
Textures in Processing
Saving a Bitmap
What's Next
Review

11. Physical Feedback
Using Motors
DC Motors
Stepper Motors
Other Options
Using Servos
Connecting a Servo
Communicating with the Servo
Wiring a Servo
Using Household Currents
Working with Appliances
Introducing the LilyPad Board
Using Vibration
Using an LED Matrix
Using the Matrix Library
Using the LedControl Library
Using the SPI Protocol
Using LCDs
Serial LCD
Using Solenoids for Movement
What's Next
Review

12. Protocols and Communication
Communicating Over Networks
Using XML
Understanding Networks and the Internet
Network Organization
Network Identification
Network Data Flow
Handling Network Communication in Processing
Client Class
Server Class
Sharing Data Across Applications
Understanding Protocols in Networking
Using ofxNetwork
Creating Networks with the Arduino
Initializing the Ethernet Library
Creating a Client Connection

Creating a Server Connection
Using Carnivore to Communicate
Installing the Carnivore Library
Creating a Carnivore Client
Communicating with Bluetooth
Using Bluetooth in Processing
Using the bluetoothDesktop Library
Using the Arduino Bluetooth
Communicating Using MIDI
Review

PART III. EXPLORATIONS

13. Graphics and OpenGL
What Does 3D Have to Do with Interaction?
Understanding 3D
Working with 3D in Processing
Lighting in Processing
Controlling the Viewer's Perspective
Making Custom Shapes in Processing
Using Coordinates and Transforms in Processing
Working with 3D in OpenGL
So, What Is OpenGL?
Transformations
OpenGL in Processing
OpenGL in openFrameworks
Using Matrices and Transformations in OpenGL
Using Vertices in OpenGL
Drawing with Textures in oF
Lighting in OpenGL
Blending Modes in OpenGL
Using Textures and Shading in Processing
Applying Material Properties
Using Another Way of Shading
What Does GLSL Look Like?
Vertex Shaders
Fragment Shader
Variables Inside Shaders
Using an ofShader Addon
What to Do Next
Review

14. Detection and Gestures
Computer Vision
Interfaces Without Controls
Example CV Projects
OpenCV
Using Blobs and Tracking
Starting with ofxOpenCV
Tracking Blobs with ofxOpenCV
Using OpenCV in Processing
Exploring Further in OpenCV
Detecting Gestures
Using ezGestures in Processing
Using Gestures in oF
Implementing Face Recognition
Exploring Touch Devices with oF
TouchKit
Tuio
Touchlib
reacTIVision
What's Next
Review

15. Movement and Location
Using Movement As and in Interaction
Using Software-Based Serial Ports
Understanding and Using GPS
Storing Data
Logging GPS Data to an Arduino
Using the Breadcrumbs Library
Implementing Hardware-Based Logging
Sending GPS Data
Determining Location by IP Address
What to Do Next
Review

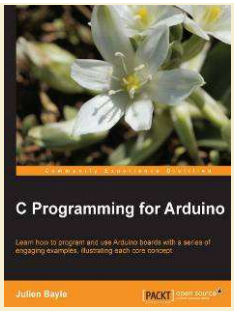
16. Interfaces and Controls
Examining Tools, Affordances, and Aesthetics
Reexamining Tilt
Exploring InputShield
Understanding Touch
Exploring Open Source Touch Hardware
Nort_/D
Liquidware TouchShield
Drawing to the TouchShield Screen
Controlling Servos Through the TouchShield
Setting Up Communication Between Arduino and TouchShield
Communicating Using OSC
Using the Wiimote
Using the Wii Nunchuck in Arduino
Tracking Wii Remote Positioning in Processing
What's Next
Review

17. Spaces and Environments
Using Architecture and Space
Sensing Environmental Data
Using an XBee with Arduino
Creating a Simple Test
Configuring the XBee Module
Addressing in the XBee
XBee Library for Processing
Placing Objects in 2D
Using the X10 Protocol
Setting Up an RFID Sensor
Reading Heat and Humidity
What's Next
Review

18. Further Resources
What's Next?
Software Tools
Construction Processes
Artificial Intelligence
Physics
Hardware Platforms
Bibliography
Interaction Design
Programming
Hardware
Art
Conclusion

Appendix
Circuit Diagram Symbols
Programming Glossary

Index



Julien Bayle

C Programming for Arduino
Программирование на C для Arduino

— Publisher: Packt Publishing, 2013. — 512 pages

ISBN-10: 1849517584
ISBN-13: 978-1849517584

Создание Ваших собственных электронных устройств является захватывающей забавой? Эта книга поможет Вам. После знакомства с Arduino вы сможете создать свой собственный электронный проект. Вы изучите язык C и научитесь, как подключать кнопки, светодиоды, LCD, сетевые модули и многое другое.

CONTENTS

Preface

Chapter 1: Let's Plug Things
What is a microcontroller?
Presenting the big Arduino family
About hardware prototyping
Understanding Arduino software architecture
Installing the Arduino development environment (IDE)
Installing the IDE
How to launch the environment?
What does the IDE look like?
Installing Arduino drivers
Installing drivers for Arduino Uno R3
Installing drivers for Arduino Duemilanove, Nano, or Diecimilla
What is electricity?
Voltage
Current and power
What are resistors, capacitors, and so on?
Wiring things and Fritzing
What is Fritzing?
Power supply fundamentals
Hello LED!
What do we want to do exactly?
How can I do that using C code?
Let's upload the code, at last!

Chapter 2: First Contact with C
An introduction to programming
Different programming paradigms
Programming style
C and C++?
C is used everywhere
Arduino is programmed with C and C++
The Arduino native library and other libraries
Discovering the Arduino native library
Other libraries included and not directly provided
Some very useful included libraries
Some external libraries
Checking all basic development steps
Using the serial monitor
Baud rate
Serial communication with Arduino
Serial monitoring
Making Arduino talk to us
Adding serial communication to Blink250ms
Serial functions in more detail
Serial.begin()
Serial.print() and Serial.println()
Digging a bit…
Talking to the board from the computer

Chapter 3: C Basics – Making You Stronger
Approaching variables and types of data
What is a variable?
What is a type?
The roll over/wrap concept
Declaring and defining variables
Declaring variables
Defining variables
String
String definition is a construction
Using indexes and search inside String
charAt()
indexOf() and lastIndexOf()
startsWith() and endsWith()
Concatenation, extraction, and replacement
Concatenation
Extract and replace
Other string functions
toCharArray()
toLowerCase() and toUpperCase()
trim()
length()
Testing variables on the board
Some explanations
The scope concept
static, volatile, and const qualifiers
static
volatile
const
Operators, operator structures, and precedence
Arithmetic operators and types
Character types
Numerical types
Condensed notations and precedence
Increment and decrement operators
Type manipulations
Choosing the right type
Implicit and explicit type conversions
Implicit type conversion
Explicit type conversion
Comparing values and Boolean operators
Comparison expressions
Combining comparisons with Boolean operators
Combining negation and comparisons
Adding conditions in the code
if and else conditional structure
switch…case…break conditional structure
Ternary operator
Making smart loops for repetitive tasks
for loop structure
Playing with increment
Using imbricated for loops or two indexes
while loop structure
do…while loop structure
Breaking the loops
Infinite loops are not your friends

Chapter 4: Improve Programming with Functions, Math, and Timing
Introducing functions
Structure of a function
Creating function prototypes using the Arduino IDE
Header and name of functions
Body and statements of functions
Benefits of using functions
Easier coding and debugging
Better modularity helps reusability
Better readability
C standard mathematical functions and Arduino
Trigonometric C functions in the Arduino core
Some prerequisites

Trigonometry functions
Exponential functions and some others
Approaching calculation optimization
The power of the bit shift operation
What are bit operations?
Binary numeral system
AND, OR, XOR, and NOT operators
Bit shift operations
It is all about performance
The switch case labels optimization techniques
Optimizing the range of cases
Optimizing cases according to their frequency
The smaller the scope, the better the board
The Tao of returns
The direct returns concept
Use void if you don't need return
Secrets of lookup tables
Table initialization
Replacing pure calculation with array index operations
The Taylor series expansion trick
The Arduino core even provides pointers
Time measure
Does the Arduino board own a watch?
The millis() function
The micros() function
Delay concept and the program flow
What does the program do during the delay?
The polling concept – a special interrupt case
The interrupt handler concept
What is a thread?
A real-life polling library example

Chapter 5: Sensing with Digital Inputs
Sensing the world
Sensors provide new capacities
Some types of sensors
Quantity is converted to data
Data has to be perceived
What does digital mean?
Digital and analog concepts
Inputs and outputs of Arduino
Introducing a new friend – Processing
Is Processing a language?
Let's install and launch it
A very familiar IDE
Alternative IDEs and versioning
Checking an example
Processing and Arduino
Pushing the button
What is a button, a switch?
Different types of switches
A basic circuit
Wires
The circuit in the real world
The pull-up and pull-down concept
The pseudocode
The code
Making Arduino and Processing talk
The communication protocol
The Processing code
The new Arduino firmware talk-ready
Playing with multiple buttons
The circuit
The Arduino code
The Processing code
Understanding the debounce concept
What? Who is bouncing?
How to debounce

Chapter 6: Sensing the World – Feeling with Analog Inputs
Sensing analog inputs and continuous values
How many values can we distinguish?
Reading analog inputs
The real purpose of the potentiometer
Changing the blinking delay of an LED with a potentiometer
How to turn the Arduino into a low voltage voltmeter?
Introducing Max 6, the graphical programming framework
A brief history of Max/MSP
Global concepts
What is a graphical programming framework?
Max, for the playground
MSP, for sound
Jitter, for visuals
Gen, for a new approach to code generation
Summarizing everything in one table
Installing Max 6
The very first patch
Playing sounds with the patch
Controlling software using hardware
Improving the sequencer and connecting Arduino
Let's connect Arduino to Max 6
The serial object in Max 6
Tracing and debugging easily in Max 6
Understanding Arduino messages in Max 6
What is really sent on the wire?
Extracting only the payload?
ASCII conversions and symbols
Playing with sensors
Measuring distances
Reading a datasheet?
Let's wire things
Coding the firmware
Reading the distance in Max 6
Measuring flexion
Resistance calculations
Sensing almost everything
Multiplexing with a CD4051 multiplexer/demultiplexer
Multiplexing concepts
Multiple multiplexing/demultiplexing techniques
Space-division multiplexing
Frequency-division multiplexing
Time-division multiplexing
The CD4051B analog multiplexer
What is an integrated circuit?
Wiring the CD4051B IC?
Supplying the IC
Analog I/O series and the common O/I
Selecting the digital pin

Chapter 7: Talking over Serial
Serial communication
Serial and parallel communication
Types and characteristics of serial communications
Synchronous or asynchronous
Duplex mode
Peering and bus
Data encoding
Multiple serial interfaces
The powerful Morse code telegraphy ancestor
The famous RS-232
The elegant I2C
The synchronous SPI
The omnipresent USB

Chapter 8: Designing Visual Output Feedback
Using LEDs
Different types of LEDs
Monochromatic LEDS
Polychromatic LEDs
Remembering the Hello LED example
Multiple monochromatic LEDs
Two buttons and two LEDs
Control and feedback coupling in interaction design
The coupling firmware
More LEDs?
Multiplexing LEDs
Connecting 75HC595 to Arduino and LEDs
Firmware for shift register handling
Global shift register programming pattern
Playing with chance and random seeds
Daisy chaining multiple 74HC595 shift registers
Linking multiple shift registers
Firmware handling two shift registers and 16 LEDs
Current short considerations
Using RGB LEDs
Some control concepts
Different types of RGB LEDs
Lighting an RGB LED
Red, Green, and Blue light components and colors
Multiple imbricated for() loops
Building LED arrays
A new friend named transistor
The Darlington transistors array, ULN2003
The LED matrix
Cycling and POV
The circuit
The 3 x 3 LED matrix code
Simulating analog outputs with PWM
The pulse-width modulation concept
Dimming an LED
A higher resolution PWM driver component
Quick introduction to LCD
HD44780-compatible LCD display circuit
Displaying some random messages

Chapter 9: Making Things Move and Creating Sounds
Making things vibrate
The piezoelectric sensor
Wiring a vibration motor
Firmware generating vibrations
Higher current driving and transistors
Controlling a servo
When do we need servos?
How to control servos with Arduino
Wiring one servo
Firmware controlling one servo using the Servo library
Multiple servos with an external power supply
Three servos and an external power supply
Driving three servos with firmware
Controlling stepper motors
Wiring a unipolar stepper to Arduino
Firmware controlling the stepper motor
Air movement and sounds
What actually is sound?
How to describe sound
Microphones and speakers
Digital and analog domains
How to digitalize sound
How to play digital bits as sounds
How Arduino helps produce sounds
Playing basic sound bits
Wiring the cheapest sound circuit
Playing random tones
Improving the sound engine with Mozzi
Setting up a circuit and Mozzi library
An example sine wave
Oscillators
Wavetables
Frequency modulation of a sine wave
Adding a pot
Upgrading the firmware for input handling
Controlling the sound using envelopes and MIDI
An overview of MIDI
MIDI and OSC libraries for Arduino
Generating envelopes
Implementing envelopes and MIDI
Wiring a MIDI connector to Arduino
Playing audio files with the PCM library
The PCM library
WAV2C – converting your own sample
Wiring the circuit
Other reader libraries

Chapter 10: Some Advanced Techniques
Data storage with EEPROMs
Three native pools of memory on the Arduino boards
Writing and reading with the EEPROM core library
External EEPROM wiring
Reading and writing to the EEPROM
Using GPS modules
Wiring the Parallax GPS receiver module
Parsing GPS location data
Arduino, battery, and autonomy
Classic cases of USB power supplying
Supplying external power
Supplying with batteries
Power adapter for Arduino supply
How to calculate current consumption
Drawing on gLCDs
Wiring the device
Demoing the library
Some useful methods' families
Global GLCD methods
Drawing methods
Text methods
Using VGA with the Gameduino Shield

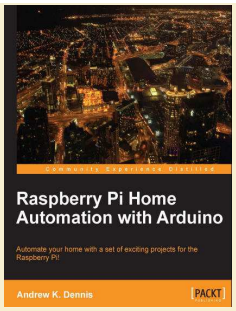
Chapter 11: Networking
An overview of networks
Overview of the OSI model
Protocols and communications
Data encapsulation and decapsulation
The roles of each layer
Physical layer
Data link layer
Network layer
Transport layer
Application/Host layers
Some aspects of IP addresses and ports
The IP address
The subnet
The communication port
Wiring Arduino to wired Ethernet
Making Processing and Arduino communicate over Ethernet
Basic wiring
Coding network connectivity implementation in Arduino
Coding a Processing Applet communicating on Ethernet
Some words about TCP
Bluetooth communications
Wiring the Bluetooth module
Coding the firmware and the Processing applet
Playing with Wi-Fi
What is Wi-Fi?
Infrastructure mode
Ad hoc mode
Other modes
The Arduino Wi-Fi shield
Basic Wi-Fi connection without encryption
Arduino Wi-Fi connection using WEP or WPA2
Using WEP with the Wi-Fi library
Using WPA2 with the Wi-Fi library
Arduino has a (light) web server
Tweeting by pushing a switch
An overview of APIs
Twitter's API
Using the Twitter library with OAuth support
Grabbing credentials from Twitter
Coding a firmware connecting to Twitter

Chapter 12: Playing with the Max 6 Framework
Communicating easily with Max 6 – the [serial] object
The [serial] object
Selecting the right serial port
The polling system
Parsing and selecting data coming from Arduino
The readAll firmware
The ReadAll Max 6 patch
Requesting data from Arduino
Parsing the received data
Distributing received data and other tricks
Creating a sound-level meter with LEDs
The circuit
The Max 6 patch for calculating sound levels
The firmware for reading bytes
The pitch shift effect controlled by hand
The circuit with the sensor and the firmware
The patch for altering the sound and parsing Arduino messages

Chapter 13: Improving your C Programming and
Creating Libraries
Programming libraries
The header file
The source file
Creating your own LED-array library
Wiring six LEDs to the board
Creating some nice light patterns
Designing a small LED-pattern library
Writing the LEDpatterns.h header
Writing the LEDpatterns.cpp source
Writing the keyword.txt file
Using the LEDpatterns library
Memory management
Mastering bit shifting
Multiplying/dividing by multiples of 2
Packing multiple data items into bytes
Turning on/off individual bits in a control and port register
Reprogramming the Arduino board

Conclusion
About Packt Publishing
About Packt Open Source
Writing for Packt

Index



Andrew K. Dennis

Raspberry Pi Home Automation with Arduino
Raspberry Pi и домашняя автоматизация с Arduino

— Publisher: Packt Publishing, 2013. — 176 pages

ISBN-10: 1849695865
ISBN-13: 978-1849695862

Если у Вас будет желание, то эта книга покажет Вам как объединить Raspberry Pi с платой Arduino, чтобы автоматизировать весь Ваш дом.

CONTENTS

Preface

Chapter 1: An Introduction to the Raspberry Pi, Arduino, and Home Automation
What we will explore in this book
History and background of the Raspberry Pi
Raspberry Pi hardware specifications
Dimensions
3.5mm analog audio jack
Composite RCA port
Two USB 2.0 ports plus one micro USB
HDMI port
SD card port
256 MB/512 MB SDRAM shared with GPU
CPU
GPU
Ethernet port
GPIO pins
History and background of Arduino
Raspberry Pi to Arduino shield connection bridge
Shield specifications
XBee socket
Power source selector
UART
Digital GPIO pins
Serial Peripheral Interface (SPI) pins
In Circuit Serial Programmer (ICSP) connector
Power pins
Analog inputs
Raspberry Pi GPIO connector
Soldering
Writing software for the Arduino
What home automation is
A history of home automation
X10 – a standard is born
The dot.com boom and open source – a new set of technologies
Commercial products
Arrival of the Raspberry Pi
Summary

Chapter 2: Getting Started Part 1 – Setting up Your Raspberry Pi
The SD card – our Raspberry Pi's storage device
Pre-installed SD card versus a blank one
Setting up the SD card
Formatting our card
Formatting instructions for Windows 7
Formatting instructions for Mac OS X
Formatting instructions for Linux
BerryBoot – our tool for installing an operating system
Downloading the BerryBoot zip
Windows
Mac
Linux
Hooking up the Raspberry Pi
Downloading the right operating system
Installing Raspbian
Installation complete
Windows users
Mac and Linux users
Summary

Chapter 3: Getting Started Part 2 – Setting up Your Raspberry Pi to Arduino Bridge Shield
Raspberry Pi to Arduino bridge shield
Checking which version of the Raspberry Pi we have
Setting up the Raspberry Pi to Arduino shield and LED
Installing the software
The Arduino IDE
A quick look at the language
arduPi – a library for our Raspberry Pi and Arduino shield
Installing arduPi
Leafpad – a text editor
Blinking LED application
A guide to the code
Compiling and running our application
Summary

Chapter 4: Our First Project – A Basic Thermometer
Building a thermometer
Setting up our hardware
An introduction to resistors
Thermistor
10K Ohm resistor
Wires
Breadboard
Connecting our components
Software for our thermometer
Geany IDE
Installing the IDE
An introduction to Makefiles
Thermometer code
Writing our application
Compiling and testing
What if it doesn't work
Up and running
Summary

Chapter 5: From Thermometer to Thermostat – Building upon Our First Project
Safety first
Introducing the thermostat
Setting up our hardware
Relays
Connecting the relay
Setting up our software
A program to test the relay
Installing screen
cURL
Thermostat code
Testing our thermostat and fan
Attaching the fan
Starting your thermostat application
Debugging problems
Summary

Chapter 6: Temperature Storage – Setting up a Database to Store Your Results
SQLite
Installing SQLite Version 3.x
Creating a database
A table to record our temperature
A table to record our rooms
Writing some SQL
Apache web server

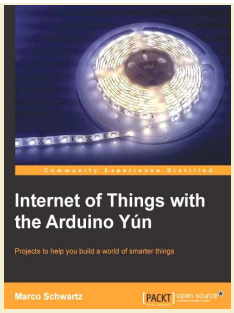
Setting up a basic web server
WSGI
Setting up WSGI
Creating a Python application to write to our database
Conclusion
HTSQL
Download HTSQL
Configuring HTSQL
Testing our Arduino shield with our database
Summary

Chapter 7: Curtain Automation – Open and Close the Curtains Based on the Ambient Light
Photoresistors
Motor shield and motors
Setting up the photoresistor
Wiring up the components
Testing the photoresistor with software
Debug
Setting up the motor shield
Wiring up the components
Curtain control application
Pulse Width Modulation
Threads
Writing our code
Debugging problems
Connecting to your blinds/curtains
Setting the timing
Attaching the hardware
Debugging problems
Summary

Chapter 8: Wrapping up
A brief review of what we have learned
Next steps
Prototyping Pi Plate
The wiringPi library
The Gertboard
Introduction to the Gertboard components
GPIO PCB expansion board
GPIO Pins
Motor controller
Open collector driver
Buffered I/O
Atmel ATmeg chip microcontroller
Convertors – analog to digital and digital to analog
Writing software for the Gertboard
Ideas for next step projects
Expanding the curtain automation tool to include temperature sensing
Changing the motor on the curtain automation project to a stepper motor
Switching lights on with a photoresistor
Holiday lights from LEDs
The future of home automation
3D printing
RFID chips
EEG headsets
Summary

Appendix: References
Raspberry Pi
Raspberry Pi to Arduino bridge shield
Linux
Python
C/C++
Arduino
SQL
HTSQL
Apache
Electronics
Packt Publishing titles
Home automation technology
3D printing
EEG headsets
Miscellaneous resources

Index



Marco Schwartz

Internet of Things with the Arduino Yun
Интернет-сервисы и Arduino Yun

— Publisher: Packt Publishing, 2014. — 112 pages

ISBN-10: 1783288000
ISBN-13: 978-1783288007

Из книги вы узнаете, как взаимодействуют различные датчики и исполнительные устройства с Arduino. Исследуете возможности, предлагаемые Интернетом: как с помощью Arduino загрузить измерения на Google Docs, картинки в Dropbox и отправить потоковое видео на YouTube.

Изучите, как использовать Arduino в качестве мозга робота, которым можно полностью управлять через Wi-Fi.

CONTENTS

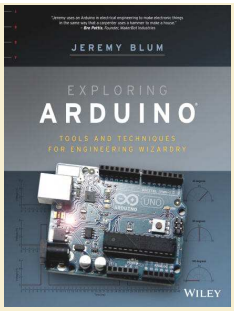
Preface

Chapter 1:
Building a Weather Station Connected to the Cloud
The required hardware and software components
Connecting the sensors to the Arduino Yún board
Creating a Temboo account
Sending data to Google Docs and displaying it
Creating automated e-mail alerts
Making your Arduino Yún board tweet sensor data
Summary

Chapter 2:
Creating a Remote Energy Monitoring and Control Device
The required hardware and software components
Connecting the components to the Yún board
Testing your hardware connections
Sending data to Google Docs
Building an interface to switch the lights on/off remotely
Summary

Chapter 3:
Making Your Own Cloud-connected Camera
Getting started
The required hardware and software components
Making hardware connections
Testing your hardware connections
Recording pictures when motion is detected
Sending pictures to Dropbox at regular intervals
Live video streaming via Wi-Fi
Summary

Chapter 4:
Wi-Fi-controlled Mobile Robot
Building the mobile robot
The required hardware and software components
Robot assembly
Connecting the Arduino Yún and Uno boards
Testing the robot's hardware connections
Building the Arduino sketch
Building the computer interface
Summary



Jeremy Blum

Exploring Arduino: Tools and Techniques for Engineering Wizardry

Изучение Arduino: инструменты и методы инженерного искусства

— Publisher: Wiley, 2013. — 384 pages

ISBN-10: 1118549368
ISBN-13: 978-1118549360

Научитесь легко создавать гаджеты, роботов и многое другое с помощью Arduino.

Эта уникальная книга написана экспертом Arduino Джереми Блумом, использует популярную платформу микроконтроллера Arduino в качестве инструмента, чтобы научить вас электротехнике, программированию и человеко-машинному взаимодействию. Являетесь ли вы подающим надежды человеком, увлечённым своим хобби, или инженером: вы извлечёте пользу из этих уроков.

CONTENTS

Introduction

Part I. Arduino Engineering Basics

Chapter 1. Getting Up and Blinking with the Arduino
Exploring the Arduino Ecosystem
Arduino Functionality
Atmel Microcontroller
Programming Interfaces
General I/O and ADCs
Power Supplies
Arduino Boards
Creating Your First Program
Downloading and Installing the Arduino IDE
Running the IDE and Connecting to the Arduino
Breaking Down Your First Program

Chapter 2. Digital Inputs, Outputs, and Pulse-Width Modulation
Digital Outputs
Wiring Up an LED and Using Breadboards
Working with Breadboards
Wiring LEDs
Programming Digital Outputs
Using For Loops
Pulse-Width Modulation with analogWrite()
Reading Digital Inputs
Reading Digital Inputs with Pulldown Resistors
Working with "Bouncy" Buttons
Building a Controllable RGB LED Nightlight

Chapter 3. Reading Analog Sensors
Understanding Analog and Digital Signals
Comparing Analog and Digital Signals
Converting an Analog Signal to a Digital One
Reading Analog Sensors with the Arduino: analogRead()
Reading a Potentiometer
Using Analog Sensors
Working with Analog Sensors to Sense Temperature
Using Variable Resistors to Make Your Own Analog Sensors
Using Resistive Voltage Dividers
Using Analog Inputs to Control Analog Outputs

Part II. Controlling Your Environment

Chapter 4. Using Transistors and Driving Motors
Driving DC Motors
Handling High-Current Inductive Loads
Using Transistors as Switches
Using Protection Diodes
Using a Secondary Power Source
Wiring the Motor
Controlling Motor Speed with PWM
Using an H-Bridge to Control DC Motor Direction
Building an H-bridge Circuit
Operating an H-bridge Circuit
Driving Servo Motors
Understanding the Difference Between Continuous Rotation and Standard Servos
Understanding Servo Control
Controlling a Servo
Building a Sweeping Distance Sensor

Chapter 5. Making Sounds
Understanding How Speakers Work
The Properties of Sound
How a Speaker Produces Sound
Using tone() to Make Sounds
Including a Definition File
Wiring the Speaker
Making Sound Sequences
Using Arrays
Making Note and Duration Arrays
Completing the Program
Understanding the Limitations of the tone() Function
Building a Micro Piano

Chapter 6. USB and Serial Communication
Understanding the Arduino's Serial Communication Capabilities
Arduino Boards with an Internal or External FTDI USB-to-Serial Converter
Arduino Boards with a Secondary USB-Capable ATMega MCU
Emulating a Serial Converter
Arduino Boards with a Single USB-Capable MCU
Arduino Boards with USB-Host Capabilities
Listening to the Arduino
Using print Statements
Using Special Characters
Changing Data Type Representations
Talking to the Arduino
Reading Information from a Computer or Other Serial Device
Telling the Arduino to Echo Incoming Data
Understanding the Differences Between Chars and Ints
Sending Single Characters to Control an LED
Sending Lists of Values to Control an RGB LED
Talking to a Desktop App
Talking to Processing
Installing Processing
Controlling a Processing Sketch from Your Arduino
Sending Data from Processing to Your Arduino
Learning Special Tricks with the Arduino Leonardo (and Other 32U4-Based Arduinos)
Emulating a Keyboard
Typing Data into the Computer
Commanding Your Computer to Do Your Bidding
Emulating a Mouse

Chapter 7. Shift Registers
Understanding Shift Registers
Sending Parallel and Serial Data
Working with the 74HC595 Shift Register
Understanding the Shift Register Pin Functions
Understanding How the Shift Register Works
Shifting Serial Data from the Arduino

Converting Between Binary and Decimal Formats
Controlling Light Animations with a Shift Register
Building a "Light Rider"
Responding to Inputs with an LED Bar Graph

Part III. Communication Interfaces

Chapter 8. The I2C Bus
History of the I2C Bus
I2C Hardware Design
Communication Scheme and ID Numbers
Hardware Requirements and Pull-Up Resistors
Communicating with an I2C Temperature Probe
Setting Up the Hardware
Referencing the Datasheet
Writing the Software
Combining Shift Registers, Serial Communication, and I2C Communications
Building the Hardware for a Temperature Monitoring System
Modifying the Embedded Program
Writing the Processing Sketch

Chapter 9. The SPI Bus
Overview of the SPI Bus
SPI Hardware and Communication Design
Hardware Configuration
Communication Scheme
Comparing SPI to I2C
Communicating with an SPI Digital Potentiometer
Gathering Information from the Datasheet
Setting Up the Hardware
Writing the Software
Creating an Audiovisual Display Using SPI Digital Potentiometers
Setting Up the Hardware
Modifying the Software

Chapter 10. Interfacing with Liquid Crystal Displays
Setting Up the LCD
Using the LiquidCrystal Library to Write to the LCD
Adding Text to the Display
Creating Special Characters and Animations
Building a Personal Thermostat
Setting Up the Hardware
Displaying Data on the LCD
Adjusting the Set Point with a Button
Adding an Audible Warning and a Fan
Bringing It All Together: The Complete Program
Taking This Project to the Next Level

Chapter 11. Wireless Communication with XBee Radios
Understanding XBee Wireless Communication
XBee Radios
The XBee Radio Shield and Serial Connections
3.3V Regulator
Logic Level Shifting
Associate LED and RSSI LED
UART Selection Jumper or Switch
Hardware vs. Software Serial UART Connection Option
Configuring Your XBees
Configuring via a Shield or a USB Adapter
Programming Option 1: Using the Uno as a Programmer (Not Recommended)
Programming Option 2: Using the SparkFun USB Explorer (Recommended)
Choosing Your XBee Settings and Connecting Your XBee to Your Host Computer
Configuring Your XBee with X-CTU
Configuring Your XBee with a Serial Terminal
Talking with Your Computer Wirelessly
Powering Your Remote Arduino
USB with a Computer or a 5V Wall Adapter
Batteries
Wall Power Adapters
Revisiting the Serial Examples: Controlling Processing with a Potentiometer
Revisiting the Serial Examples: Controlling an RGB LED
Talking with Another Arduino: Building a Wireless Doorbell
System Design
Transmitter Hardware
Receiver Hardware
Transmitter Software
Receiver Software

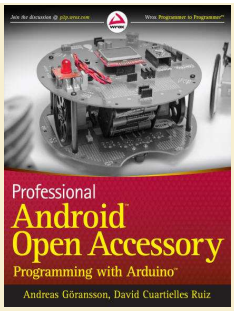
Part IV. Advanced Topics and Projects

Chapter 12. Hardware and Timer Interrupts
Using Hardware Interrupts
Knowing the Tradeoffs Between Polling and Interrupting
Ease of Implementation (Software)
Ease of Implementation (Hardware)
Multitasking
Acquisition Accuracy
Understanding the Arduino's Hardware Interrupt Capabilities
Building and Testing a Hardware-Debounced Button Interrupt Circuit
Creating a Hardware-Debouncing Circuit
Assembling the Complete Test Circuit
Writing the Software
Using Timer Interrupts
Understanding Timer Interrupts
Getting the Library
Executing Two Tasks Simultaneously(ish)
Building an Interrupt-Driven Sound Machine
Sound Machine Hardware
Sound Machine Software

Chapter 13. Data Logging with SD Cards
Getting Ready for Data Logging
Formatting Data with CSV Files
Preparing an SD Card for Data Logging
Interfacing the Arduino with an SD Card
SD Card Shields
SD Card SPI Interface
Writing to an SD Card
Reading from an SD Card
Using a Real-Time Clock
Understanding Real-Time Clocks
Using the DS1307 Real-Time Clock
Using the RTC Arduino Third-Party Library
Using the Real-Time Clock
Installing the RTC and SD Card Modules
Updating the Software
Building an Entrance Logger
Logger Hardware
Logger Software
Data Analysis

Chapter 14. Connecting Your Arduino to the Internet
The Web, the Arduino, and You
Networking Lingo
IP Address
Network Address Translation
MAC Address
HTML
HTTP
GET/POST
DHCP
DNS
Clients and Servers
Networking Your Arduino
Controlling Your Arduino from the Web
Setting Up the I/O Control Hardware
Designing a Simple Web Page
Writing an Arduino Server Sketch
Connecting to the Network and Retrieving an IP via DHCP
Replying to a Client Response
Putting It Together: Web Server Sketch
Controlling Your Arduino via the Network
Controlling Your Arduino over the Local Network
Using Port Forwarding to Control your Arduino from Anywhere
Sending Live Data to a Graphing Service
Building a Live Data Feed on Xively
Creating a Xively Account
Creating a Data Feed
Installing the Xively and HttpClient Libraries
Wiring Up Your Arduino
Configuring the Xively Sketch and Running the Code
Displaying Data on the Web
Adding Feed Components
Adding an Analog Temperature Sensor
Adding Additional Sensor Readings to the Datastream

Appendix. Deciphering the ATMega Datasheet and Arduino Schematics
Reading Datasheets
Breaking Down a Datasheet
Understanding Component Pin-outs
Understanding the Arduino Schematic



Andreas Goransson, David Cuartielles Ruiz

Professional Android Open Accessory Programming with Arduino

Профессиональное программирование на Android Open Accessory для Arduino

— Publisher: Wrox, 2013. — 408 pages

ISBN-10: 1118454766
ISBN-13: 978-1118454763

Это руководство показывает программистам Android, как использовать AOA с платформой Ардуино, чтобы управлять такими системами как освещение, кондиционирование воздуха и др.

CONTENTS

INTRODUCTION

PART I: WELCOME TO THE WONDERFUL WORLD OF ACCESSORIES

CHAPTER 1: INTRODUCTION TO ANDROID OPEN ACCESSORY
I, Android
The Three Laws of Android
The Android Philosophy
Other Popular Systems
Preinstalled Applications
What Is Android Open Accessory?
Android USB in Short
Developing Android Accessories
What Is Arduino?
How Does AOA Work with Arduino?
What Can You Do with AOA?
What Can't You Do with AOA?
Why it Matters that Google Chose Arduino
Summary

CHAPTER 2: SETTING UP THE (ARDUINO) HARDWARE
Choosing Microcontroller Boards for Your Project
One Platform, Many Architectures
Shields
Choosing Sensors and Actuators for Your Project
Sensors
Actuators
Powering up Your Project
Ways to Power up Your Project
Arduino Feeding Your Phone
Summary

CHAPTER 3: UNDERSTANDING DATA COMMUNICATION
Data Communication Basics
Protocols
Terminology
Hardware Layer for the Communication Protocol
ADB
Accessory Mode
Host Mode
TCP/IP
Audio Port
Bluetooth Options
Introducing MQTT
Heads Up!
MQTT Messages
P2PMQTT: A Modi? ed MQTT
Establishing a Connection
Subscribing to a Topic
Publishing a Message
Disconnecting
Summary

CHAPTER 4: SETTING UP DEVELOPMENT ENVIRONMENTS
Setting up Android Development
Android Development Environment
Hello, Android!
Setting up Arduino Development
Arduino Development Environment
Hello, Arduino!
Hello Open Accessory App
The Temperature Sensor
The Arduino Sketch
The Android Project
Ready to Go
Summary

CHAPTER 5: CREATING THE ACCESSORY LIBRARY
Getting Started with Android Libraries
Building the P2PMQTT Library
Preparing the Library Project
Sketching the API
Implementing MQTT
Decoding MQTT
Managing Open Accessory Connections
Creating the Connection Class
USB Connection
Bluetooth Connection
Creating the Connection
Summary

CHAPTER 6: USING YOUR ACCESSORY LIBRARY
Using Custom Android Libraries
The WroxAccessories Library
Building the Mini Projects
The LSMSD
The Parking Assistant
The Basic Robot
The Sampler
Summary

CHAPTER 7: DIGITAL ARDUINO
Digital Actuators
The Blinking LEDs
Controlling a Desk Lamp — The Relay
Digital Project 1: Large SMS Display
Writing the Arduino Program
Digital Sensors
Buttons and Switches
Tilt Sensor
Digital Project 2: Small Sampler
Summary

CHAPTER 8: ANALOG ARDUINO
Analog Actuators
The Piezo Element
Motors
Analog Project 1: The Basic Robot
Analog Sensors
Potentiometers
Ultrasound Sensors
Analog Project 2: The Parking Assistant
Summary

PART II: PROJECTS

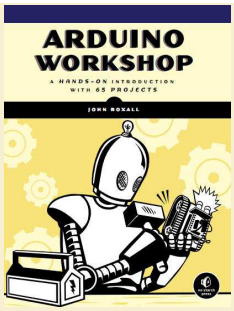
CHAPTER 9: BIKE RIDE RECORDER
The Concept Behind Bike Computers

The Design Brief
Working with the Arduino Side
Creating the Hardware and Mechanics
Programming the Bike Computer
Building the Android App
Creating the Bike Ride Recorder Project
Creating the User Interface
Setting up the AoaService
Building the Main Menu Activity
Building the Recording Activity
Building the List Recordings View
Building the Playback View Activity
Making Further Improvements
Mechanics
More Sensors
Making a Better App
Summary

CHAPTER 10: KITCHEN LAMP
The Concept
The Design Brief
The Arduino Side
Hardware
Software
Building The Android App
Sketching the Application Layout
Create the Kitchen Lamp Project
Create the User Interface
Building the Kitchen Timer
Responding to Phone Calls
Listen for SMS Events
Connecting to the WroxAccessory
Further Improvements
Product-ready Embedded System
Making a Better App
Summary

CHAPTER 11: MR. WILEY
The Concept
The Design Brief
The Arduino Side
The Hardware
The Firmware (on the Robot Board)
Creating Software for the Mega ADK Board
Building the Android App
Sketching the Application Layout
Creating the Mr. Wiley Project
Building the Computer Vision Algorithm
Connecting to the WroxAccessory
Making Further Improvements
Electronics
Making a Better App
Summary

INDEX



John Boxall

Arduino Workshop: A Hands-On Introduction with 65 Projects
Мастерская Arduino: практическое знакомство с 65 проектами

— Publisher: No Starch Press, 2013. — 392 pages

ISBN-10: 1593274483
ISBN-13: 978-1593274481

Arduino - недорогая гибкая микроконтроллерная платформа с открытым исходным кодом предлагает бесчисленные способы создания устройств, взаимодействующих с миром вокруг вас.

Среди проектов книги 65 полезных устройств: цифровой термометр, который строит диаграмму изменений температуры на жидкокристаллическом дисплее; регистратор GPS, который записывает данные перемещений, которые могут быть выведены на экран из Google Maps; и многое другое.

CONTENTS

ACKNOWLEDGMENTS

1. GETTING STARTED
The Possibilities Are Endless
Strength in Numbers
Parts and Accessories
Required Software
Mac OS X
Windows XP and Later
Ubuntu Linux 904 and Later
Safety
Looking Ahead

2. EXPLORING THE ARDUINO BOARD AND THE IDE
The Arduino Board
Taking a Look Around the IDE
The Command Area
The Text Area
The Message Window Area
Creating Your First Sketch in the IDE
Comments
The Setup Function
Controlling the Hardware
The Loop Function
Verifying Your Sketch
Uploading and Running Your Sketch
Modifying Your Sketch
Looking Ahead

3. FIRST STEPS
Planning Your Projects
About Electricity
Current
Voltage
Power
Electronic Components
The Resistor
The Light-Emitting Diode
The Solderless Breadboard

Project #1: Creating a Blinking LED Wave
The Algorithm
The Hardware
The Sketch
The Schematic
Running the Sketch
Using Variables

Project #2: Repeating with for Loops
Varying LED Brightness with Pulse-Width Modulation

Project #3: Demonstrating PWM
More Electric Components
The Transistor
The Rectifier Diode
The Relay
Higher-Voltage Circuits
Looking Ahead

4. BUILDING BLOCKS
Using Schematic Diagrams
Identifying Components
Wires in Schematics
Dissecting a Schematic
The Capacitor
Measuring the Capacity of a Capacitor
Reading Capacitor Values
Types of Capacitors
Digital Inputs

Project #4: Demonstrating a Digital Input
The Algorithm
The Hardware
The Schematic
The Sketch
Modifying Your Sketch
Understanding the Sketch
Creating Constants with #define
Reading Digital Input Pins
Making Decisions with if
Making More Decisions with if-then-else
Boolean Variables
Comparison Operators
Making Two or More Comparisons

Project #5: Controlling Traffic
The Goal
The Algorithm
The Hardware
The Schematic
The Sketch
Running the Sketch
Analog vs Digital Signals

Project #6: Creating a Single-Cell Battery Tester
The Goal
The Algorithm
The Hardware
The Schematic
The Sketch
Doing Arithmetic with an Arduino
Float Variables
Comparison Operators for Calculations
Improving Analog Measurement Precision with a Reference Voltage
Using an External Reference Voltage
Using the Internal Reference Voltage
The Variable Resistor
Piezoelectric Buzzers
Piezo Schematic

Project #7: Trying Out a Piezo Buzzer

Project #8: Creating a Quick-Read Thermometer
The Goal
The Hardware
The Schematic
The Sketch
Hacking the Sketch
Looking Ahead

5. WORKING WITH FUNCTIONS

Project #9: Creating a Function to Repeat an Action

Project #10: Creating a Function to Set the Number of Blinks
Creating a Function to Return a Value

Project #11: Creating a Quick-Read Thermometer That Blinks the Temperature
The Hardware
The Schematic
The Sketch
Displaying Data from the Arduino in the Serial Monitor
The Serial Monitor

Project #12: Displaying the Temperature in the Serial Monitor
Debugging with the Serial Monitor
Making Decisions with while Statements
do-while
Sending Data from the Serial Monitor to the Arduino

Project #13: Multiplying a Number by Two
long Variables

Project #14: Using long Variables
Looking Ahead

6. NUMBERS, VARIABLES, AND ARITHMETIC
Generating Random Numbers
Using Ambient Current to Generate a Random Number

Project #15: Creating an Electronic Die
The Hardware
The Schematic
The Sketch
Modifying the Sketch
A Quick Course in Binary
Byte Variables
Increasing Digital Outputs with Shift Registers

Project #16: Creating an LED Binary Number Display
The Hardware
Connecting the 74HC
The Sketch

Project #17: Making a Binary Quiz Game
The Algorithm
The Sketch
Arrays
Defining an Array
Referring to Values in an Array
Writing to and Reading from Arrays
Seven-Segment LED Displays
Controlling the LED

Project #18: Creating a Single-Digit Display
The Hardware
The Schematic
The Sketch
Displaying Double Digits

Project #19: Controlling Two Seven-Segment LED Display Modules
The Hardware
The Schematic
Modulo

Project #20: Creating a Digital Thermometer
The Hardware
The Sketch
LED Matrix Display Modules
The LED Matrix Schematic
Making the Connections
Bitwise Arithmetic
The Bitwise AND Operator
The Bitwise OR Operator
The Bitwise XOR Operator
The Bitwise NOT Operator
Bitshift Left and Right

Project #21: Creating an LED Matrix
Project #22: Creating Images on an LED Matrix
Project #23: Displaying an Image on an LED Matrix
Project #24: Animating an LED Matrix
The Sketch
Looking Ahead

7. LIQUID CRYSTAL DISPLAYS
Character LCD Modules
Using a Character LCD in a Sketch
Displaying Text
Displaying Variables or Numbers

Project #25: Defining Custom Characters
Graphic LCD Modules
Connecting the Graphic LCD
Using the LCD
Controlling the Display

Project #26: Seeing the Text Functions in Action
Creating More Complex Display Effects

Project #27: Creating a Temperature History Monitor
The Algorithm
The Hardware
The Sketch
The Result
Modifying the Sketch
Looking Ahead

8. EXPANDING YOUR ARDUINO
Shields
ProtoShields

Project #28: Creating a Custom Shield with Eight LEDs
The Hardware
The Schematic
The Layout of the ProtoShield Board
The Design
Soldering the Components
Modifying the Custom Shield
Expanding Sketches with Libraries
Importing a Shield's Libraries
MicroSD Memory Cards
Testing Your MicroSD Card

Project #29: Writing Data to the Memory Card

Project #30: Creating a Temperature-Logging Device
The Hardware
The Sketch
Timing Applications with millis() and micros()

Project #31: Creating a Stopwatch
The Hardware
The Schematic
The Sketch
Interrupts
Interrupt Modes
Configuring Interrupts
Activating or Deactivating Interrupts

Project #32: Using Interrupts
The Sketch
Looking Ahead

9. NUMERIC KEYPADS
Using a Numeric Keypad
Wiring a Keypad
Programming for the Keypad
Testing the Sketch
Making Decisions with switch-case

Project #33: Creating a Keypad-Controlled Lock
The Sketch
How It Works
Testing the Sketch
Looking Ahead

10. ACCEPTING USER INPUT WITH TOUCHSCREENS
Touchscreens
Connecting the Touchscreen

Project #34: Addressing Areas on the Touchscreen
The Hardware
The Sketch
Testing the Sketch
Mapping the Touchscreen

Project #35: Creating a Two-Zone On/Off Touch Switch
The Sketch
How It Works
Testing the Sketch

Project #36: Creating a Three-Zone Touch Switch

The Touchscreen Map
The Sketch
How It Works
Looking Ahead

11. MEET THE ARDUINO FAMILY

Project #37: Creating Your Own Breadboard Arduino
The Hardware
The Schematic
Running a Test Sketch
The Many Arduino Boards
Arduino Uno
Freetronics Eleven
The Freeduino
The Boarduino
The Arduino Nano
The Arduino LilyPad
The Arduino Mega
The Freetronics EtherMega
The Arduino Due
Looking Ahead

12. MOTORS AND MOVEMENT
Making Small Motions with Servos
Selecting a Servo
Connecting a Servo
Putting a Servo to Work

Project #38: Building an Analog Thermometer
The Hardware
The Schematic
The Sketch
Using Electric Motors
The TIP120 Darlington Transistor

Project #39: Controlling the Motor
The Hardware
The Schematic
The Sketch

Project #40: Building and Controlling a Tank Robot
The Hardware
The Schematic
The Sketch
Sensing Collisions

Project #41: Detecting Tank Bot Collisions with a Microswitch
The Schematic
The Sketch
Infrared Distance Sensors
Wiring It Up
Testing the IR Distance Sensor

Project #42: Detecting Tank Bot Collisions with IR Distance Sensor
Ultrasonic Distance Sensors
Connecting the Ultrasonic Sensor
Using the Ultrasonic Sensor
Testing the Ultrasonic Distance Sensor

Project #43: Detecting Tank Bot Collisions with an Ultrasonic Distance Sensor
The Sketch
Looking Ahead

13. USING GPS WITH YOUR ARDUINO
What Is GPS?
Testing the GPS Shield

Project #44: Creating a Simple GPS Receiver
The Hardware
The Sketch
Displaying the Position on the LCD

Project #45: Creating an Accurate GPS-based Clock
The Hardware
The Sketch

Project #46: Recording the Position of a Moving Object over Time
The Hardware
The Sketch
Displaying Locations on a Map
Looking Ahead

14. WIRELESS DATA
Using Low-cost Wireless Modules

Project #47: Creating a Wireless Remote Control
The Hardware for the Transmitter Circuit
The Transmitter Schematic
The Hardware for the Receiver Circuit
The Receiver Schematic
The Transmitter Sketch
The Receiver Sketch
Using XBee Wireless Data Modules for Greater Range and Faster Speed

Project #48: Transmitting Data with an XBee
The Sketch
Setting Up the Computer to Receive Data

Project #49: Building a Remote Control Thermometer
The Hardware
The Layout
The Sketch
Operation
Looking Ahead

15. INFRARED REMOTE CONTROL
What Is Infrared?
Setting Up for Infrared
The IR Receiver
The Remote Control
A Test Sketch
Testing the Setup

Project #50: Creating an IR Remote Control Arduino
The Hardware
The Sketch
Expanding the Sketch

Project #51: Creating an IR Remote Control Tank
The Hardware
The Sketch
Looking Ahead

16. READING RFID TAGS
Inside RFID Devices
Testing the Hardware
The Schematic
Testing the Schematic

Project #52: Creating a Simple RFID Control System
The Sketch
How It Works
Storing Data in the Arduino's Built-in EEPROM
Reading and Writing to the EEPROM

Project #53: Creating an RFID Control with "Last Action" Memory
The Sketch
How It Works
Looking Ahead

17. DATA BUSES
The I2C Bus

Project #54: Using an External EEPROM
The Hardware
The Schematic
The Sketch
The Result

Project #55: Using a Port Expander IC
The Hardware
The Schematic
The Sketch
The SPI Bus
Pin Connections
Implementing the SPI
Sending Data to an SPI Device

Project #56: Using a Digital Rheostat
The Hardware
The Schematic
The Sketch
Looking Ahead

18. REAL-TIME CLOCKS
Connecting the RTC Module

Project #57: Adding and Displaying Time and Date with an RTC
The Hardware
The Sketch
How It Works

Project #58: Creating a Simple Digital Clock
The Hardware
The Sketch
How It Works and Results

Project #59: Creating an RFID Time-Clock System
The Hardware
The Sketch
How It Works
Looking Ahead

19. THE INTERNET
What You'll Need

Project #60: Building a Remote-Monitoring Station
The Hardware
The Sketch
Troubleshooting
How It Works

Project #61: Creating an Arduino Tweeter
The Hardware
The Sketch
Controlling Your Arduino from the Web

Project #62: Setting Up a Remote Control for Your Arduino
The Hardware
The Sketch
Controlling Your Arduino Remotely
Looking Ahead

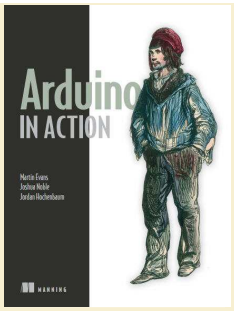
20. CELLULAR COMMUNICATIONS
The Hardware
Preparing the Power Shield
Hardware Configuration and Testing
Changing the Operating Frequency

Project #63: Building an Arduino Dialer
The Hardware
The Schematic
The Sketch
How It Works

Project #64: Building an Arduino Texter
The Sketch
How It Works

Project #65: Setting Up an SMS Remote Control
The Hardware
The Schematic
The Sketch
How It Works
Looking Ahead

index



Martin Evans, Joshua Noble

Arduino in Action
Arduino в действии

— Publisher: Manning Publications, 2013. — 368 pages

ISBN-10: 1617290246
ISBN-13: 978-1617290244

« Arduino в действии» - практическое руководство по анализу прототипа и созданию электроники с использованием платформы Arduino. Подходящая для новичков и опытных пользователей, эта книга начинается с основ и затем систематически ведёт вас через проекты: от простого мигающего светодиода до сложного сопряжения Arduino с вашим iPhone.

CONTENTS

Preface
Acknowledgments
About this book
About the cover illustration

PART 1. GETTING STARTED

1. Hello Arduino
1.1 A brief history of the Arduino
1.2 The Arduino hardware
Arduino Uno
Arduino Duemilanove
Arduino Ethernet
Arduino Mega
Other Arduino boards
Attack of the clones
Getting an Arduino
1.3 Setting up your working environment
Software for Arduino
Basic hardware setup
Your Arduino toolbox
1.4 Make something happen!
Your first blinking LED
Sketch to make an LED blink
Connecting everything
Uploading and testing
1.5 Touring the IDE
The main editor
Serial monitor
Catching errors
Process
1.6 Anatomy of a sketch
A routine called setup
The endless loop
1.7 Commenting code
1.8 Summary

2. Digital input and output
2.1 Getting started
Using a breadboard
Circuit diagram
Adding the LEDs
Connecting the hardware
Sketch to flash five LEDs
Upload and test
2.2 Gaining control
Circuit diagram
Connections
Interrupts butting in Sketch to control the LEDs with a push button
Upload and test
Time for a break
Upload and test
2.3 Reaction tester
Circuit diagram
Connections
Sketch to test reaction speed
Upload and test
2.4 Reactometer: Who really has the fastest reaction time?
Sketch to measure reaction speed
Upload and test
2.5 Summary

3. Simple projects: input and output
3.1 Time to get analog
What's the difference between analog and digital?
Reading a potentiometer
Connecting the hardware
Sketch to read a potentiometer
Upload and test
3.2 A piezoelectric transducer
The circuit diagram
Connecting the hardware
Sketch to measure output from a piezoelectric transducer
Upload and test
Circuit with added speaker
Connecting the hardware
Sketch to generate a tone
Upload and test
3.3 Making a pentatonic or five-tone keyboard
Circuit diagram
Connecting the hardware
Sketch to create a pentatonic keyboard
Upload and test
3.4 Summary

PART 2. PUTTING ARDUINO TO WORK

4. Extending Arduino
4.1 Extending the Arduino with libraries
4.2 Core library
4.3 Standard libraries
Test-driven development with Arduino TestSuite
Storing values using EEPROM
Storing more data with SD
Get connected with Ethernet
Serial communication with Firmata
Displaying data using the LiquidCrystal library
Controlling a servo motor
Turning a stepper motor
Communicating with SPI peripherals
Communicating with the two-wire interface
Get more serial ports with SoftwareSerial
4.4 Contributed libraries
Installing a new library
4.5 Expanding the Arduino with shields
Common shields
Gotchas: will it work with my Arduino?
4.6 Summary

5 Arduino in motion
5.1 Getting up to speed with DC motors
Stopping and starting
Sketch to turn a small DC motor on and off
Connecting the hardware
Upload and test
5.2 Speed control and reverse
PWM to the rescue
The H-bridge for motor control
The L293D dual H driver
Connecting the hardware
Sketch to control a motor with an L293D
Upload and test
Changing motor speed

Upload and test
5.3 Stepper motors: one step at a time
Unipolar or bipolar
Connecting the hardware
Stepper motor library functions
Sketch to control a stepper motor
Upload and test
5.4 Try not to get in a flap with servomotors
Controlling a servomotor
Servomotor functions and methods
Sketch to control a servomotor
Connecting the hardware
Upload and test
5.5 Mighty power comes in small packages with brushless DC motors
Why go brushless
Gaining control
Sketch to control a brushless motor
Connecting the hardware
Upload and test
Reverse
Sketch to reverse a brushless motor
Connecting the hardware
Upload and test
5.6 The motor control shield for more motors
5.7 Summary

6. Object detection
6.1 Object detection with ultrasound
Choosing an ultrasonic sensor
Three wires or four
Sketches for ultrasonic object finding
Connecting the hardware
Upload and test
6.2 Infrared for range finding
Infrared and ultrasound together
The Sharp GP2D12 range finder
Nonlinear algorithm for calculating distance
Sketch for range finding
Connecting the hardware
Upload and test
6.3 Passive infrared to detect movement
Using the Parallax PIR sensor
Sketch for infrared motion detection
Connecting the hardware
Upload and test
6.4 Summary

7. LCD displays
7.1 Introduction to LCDs
String variables: String type vs. char type
7.2 Parallel character LCDs: the Hitachi HD44780
4-bit or 8-bit?
Library and functions
Circuit diagram
Connecting everything up in 4-bit mode
Sketch for writing to the Hitachi HD44780
Upload and test
7.3 Serial LCD weather station
Serial vs. parallel LCDs
SerLCD library and functions
The Maxim IC DS18B20 temperature sensor
OneWire and DallasTemperature libraries
Circuit diagram
Connecting everything up
Sketch for an LCD weather station
Upload and test
7.4 Graphic LCDs: the Samsung KS0108 GLCD
Library and functions
Circuit diagram
Connecting everything up
Sketch for drawing to a GLCD
Upload and test
7.5 Summary

8. Communications
8.1 Ethernet
The Ethernet library
Ethernet Shield with SD data card
8.2 Arduino web server
Setting up the server
Sketch for creating a web server
Upload and test
Troubleshooting
8.3 Tweet tweet: talking to Twitter
Of Twitter and tokens
Libraries and functions
Circuit diagram and connecting the hardware
Sketch for the Twitter button-press tweeter
Upload and test
8.4 Wi-Fi
Arduino Wifi Shield
WiFi library and functions
Gestures: wireless accelerometers
Connecting the hardware
Sketch for Bluetooth communication
Upload and test
8.5 Bluetooth wireless
Arduino BT
Adding Bluetooth
Establishing a Bluetooth connection
Sketch for Bluetooth communication
8.6 Serial peripheral interface (SPI)
SPI library
SPI devices and digital potentiometers
Circuit diagram and connecting the hardware
Sketch for a digital LED dimmer
8.7 Data logging
Types of memory
SD cards and SD library
Sketch for an SD card sensor logger
8.8 Cosm
Sign up for an account and get an API key
Creating a new data feed
Sketch for Cosm sensor logging
Upload and test
8.9 Summary

9. Game on
9.1 Nintendo Wii salutes you
Wii Nunchuk
Nunchuk connections
Wii will talk
Wii will test
9.2 Release the Xbox
Getting connected
USB Host library
Learning about the Xbox controller using the USB Host Shield
Xbox reporting for duty
Let's boot it Interfacing with code Xboxhid.ino
Hardware connections and testing
9.3 Summary

10 Integrating the Arduino with iOS
10.1 Connecting your device to the Arduino
The Redpark serial cable
The final connection
10.2 iOS code
Creating a single-view application in Xcode
Writing the code
10.3 The Arduino gets involved
Sketch to switch LED from iOS device
Testing the sketch
10.4 Doing more with Xcode
Adding a Slider control
10.5 Arduino sliding Arduino slider circuit
Testing the circuit
10.6 Moving data to the iOS device Xcode coding
The GP2D12 IR distance sensor
Testing
10.7 Summary

11. Making wearables
11.1 Introducing the LilyPad
LilyPad accessories
Conductive thread and fabric
11.2 Creating a turn-signal jacket
11.3 Creating a wearable piano
11.4 The Arduino Pro Mini
11.5 Creating a smart headphone
11.6 Creating a jacket with a compass
11.7 Summary

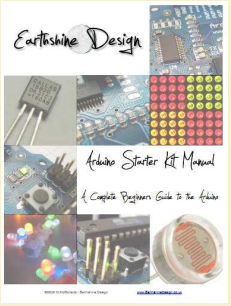
12. Adding shields
12.1 Shield basics
12.2 The Adafruit motor shield
The AFMotor library
Using the motor shield with a stepper motor
Using the motor shield with a DC motor
Getting a motor shield
12.3 Creating your own shield
Memory
Level shifters
The SD card holder
Connecting the SD card to the Arduino
Preparing the perfboard

Testing the shield
12.4 Summary

13 Software integration
13.1 The serial channel
13.2 Servos for face tracking
Assembling the face-tracking hardware
Code for face-tracking
13.3 Using Firmata to create an equalizer
Using Firmata in your application
Audio analysis in Processing
Assembling the equalizer hardware
Code for the equalizer
13.4 Using Pure Data to create a synthesizer
Assembling the synthesizer hardware
Code for the synthesizer
13.5 Using Python to monitor temperatures
The Serial library in Python
Assembling the thermometer hardware
Code for monitoring temperatures
13.6 Summary

Appendix A. Installing the Arduino IDE
Appendix B. Coding primer
Appendix C. Libraries
Appendix D. Components list
Appendix E. Useful links

Index



Mike McRoberts

Arduino Starters Kit Manual - A Complete Beginners guide to the Arduino

Руководство по Arduino Starters Kit – полное руководство для начинающих в Arduino

— Publisher: Earthshine Design, 2009. — 105 pages

Цель этой книги и набора Arduino Starters Kit (ASK) состоит в том, чтобы шаг за шагом дать основные понятия об Arduino, электронике и программировании на Си.

CONTENTS

Introduction

The Starter Kit Contents
What exactly is an Arduino?
Getting Started
Upload your first sketch
The Arduino IDE

The Projects

Chapter 1 - LED Flasher
Code Overview
Hardware Overview

Chapter 2 - SOS Morse Code Signaller
Code Overview

Chapter 3 - Traffic Lights

Chapter 4 - Interactive Traffic Lights
Code Overview

Chapter 5 - LED Chase Effect
Code Overview

Chapter 6 - Interactive LED Chase Effect
Code Overview
Hardware Overview

Chapter 7 - Pulsating Lamp
Code Overview

Chapter 8 - Mood Lamp
Code Overview

Chapter 9 - LED Fire Effect
Code Overview

Chapter 10 - Serial Controlled Mood Lamp
Code Overview

Chapter 11 - Drive a DC Motor
Code Overview
Hardware Overview

Chapter 12 - Piezo Sounder Melody Player
Code Overview
Hardware Overview

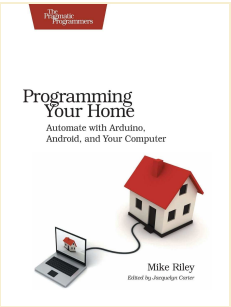
Chapter 13 - Serial Temperature Sensor
Code Overview
Hardware Overview

Chapter 14 - Light Sensor
Code Overview
Hardware Overview

Chapter 15 - Shift Register 8-Bit Binary Counter
The Binary Number System
Hardware Overview
Code Overview
Bitwise Operators
Code Overview (continued)

Chapter 16 - Dial 8-Bit Binary Counters
Code & Hardware Overview

Chapter 17 - LED Dot Matrix - Basic Animation
Code Overview



Mike Riley

Programming Your Home: Automate with Arduino, Android and Your Computer

Программирование вашего дома:
автоматизируйте с Arduino, Android и компьютером

— Publisher: Pragmatic Bookshelf, 2012. — 200 pages
— Series: Pragmatic Programmers

ISBN-10: 1934356905
ISBN-13: 978-1934356906

Возьмите под свой контроль свой дом! Автоматизируйте бытовые приборы и освещение и узнайте об Arduino и Android. В книге энтузиаст Майк Райли делится множеством своих домашних проектов автоматизации.

CONTENTS

Preface

Part I — Preparations

1. Getting Started.
1.1 What Is Home Automation?
1.2 Commercial Solutions
1.3 DIY Solutions
1.4 Justifying the Investment
1.5 Setting Up Your Workbench
1.6 Sketching Out Your Ideas
1.7 Writing, Wiring, and Testing
1.8 Documenting Your Work

2. Requirements
2.1 Knowing the Hardware
2.2 Knowing the Software
2.3 Be Safe, Have Fun!

Part II — Projects

3. Water Level Notifier
3.1 What You Need
3.2 Building the Solution
3.3 Hooking It Up
3.4 Sketching Things Out
3.5 Writing the Web Mailer
3.6 Adding an Ethernet Shield
3.7 All Together Now
3.8 Next Steps

4. Electric Guard Dog
4.1 What You Need
4.2 Building the Solution
4.3 Dog Assembly
4.4 Dog Training
4.5 Testing It Out
4.6 Unleashing the Dog
4.7 Next Steps

5. Tweeting Bird Feeder
5.1 What You Need
5.2 Building the Solution
5.3 The Perch Sensor
5.4 The Seed Sensor
5.5 Going Wireless
5.6 Tweeting with Python
5.7 Putting It All Together
5.8 Next Steps

6. Package Delivery Detector
6.1 What You Need
6.2 Building the Solution
6.3 Hardware Assembly
6.4 Writing the Code
6.5 The Package Delivery Sketch
6.6 Testing the Delivery Sketch
6.7 The Delivery Processor
6.8 Creating the Delivery Database
6.9 Installing the Package Dependencies
6.10 Writing the Script
6.11 Testing the Delivery Processor
6.12 Setting It Up
6.13 Next Steps

7. Web-Enabled Light Switch
7.1 What You Need
7.2 Building the Solution
7.3 Hooking It Up
7.4 Writing the Code for the Web Client
7.5 Testing Out the Web Client
7.6 Writing the Code for the Android Client
7.7 Testing Out the Android Client
7.8 Next Steps

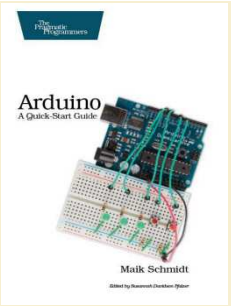
8. Curtain Automation
8.1 What You Need
8.2 Building the Solution
8.3 Using the Stepper Motor
8.4 Programming the Stepper Motor
8.5 Adding the Sensors
8.6 Writing the Sketch
8.7 Installing the Hardware
8.8 Next Steps

9. Android Door Lock
9.1 What You Need
9.2 Building the Solution
9.3 Controlling the Android Door Lock
9.4 Writing the Android Server
9.5 Writing the Android Client
9.6 Test and Install
9.7 Next Steps

10. Giving Your Home a Voice
10.1 What You Need
10.2 Speaker Setup
10.3 Giving Lion a Voice
10.4 Wireless Mic Calibration
10.5 Programming a Talking Lion
10.6 Conversing with Your Home
10.7 Next Steps

Part III — Predictions

11. Future Designs
11.1 Living in the Near
11.2 The Long View
11.3 The Home of the Future



Maik Schmidt

Arduino: A Quick-Start Guide

Arduino: быстрый старт

— Publisher: Pragmatic Bookshelf, 2011. — 272 pages
— Series: Quick-Start Guides

ISBN-10: 1934356662
ISBN-13: 978-1934356661

С помощью книги вы сможете создать свои первые гаджеты в течение нескольких минут.

CONTENTS

Preface
Who Should Read This Book
What's in This Book
Arduino Uno and the Arduino Platform
Code Examples and Conventions
Online Resources

I. Getting Started with Arduino

1. Welcome to the Arduino
1.1 What You Need
1.2 What Exactly Is an Arduino?
1.3 Exploring the Arduino Board
1.4 Installing the Arduino IDE
1.5 Meeting the Arduino IDE
1.6 Compiling and Uploading Programs
1.7 Working with LEDs
1.8 What If It Doesn't Work ?
1.9 Exercises

2. Inside the Arduino
2.1 What You Need
2.2 Managing Projects and Sketches
2.3 Changing Preferences
2.4 Using Serial Ports
2.5 What If It Doesn't Work ?
2.6 Exercises

II. Eight Arduino Projects

3. Building Binary Dice
3.1 What You Need
3.2 Working with Breadboards
3.3 Using an LED on a Breadboard
3.4 First Version of a Binary Die
3.5 Working with Buttons
3.6 Adding Our Own Button
3.7 Building a Dice Game
3.8 What If It Doesn't Work ?
3.9 Exercises

4. Building a Morse Code Generator Library
4.1 What You Need
4.2 Learning the Basics of Morse Code
4.3 Building a Morse Code Generator
4.4 Fleshing Out the Generator's Interface
4.5 Outputting Morse Code Symbols
4.6 Installing and Using the Telegraph Class
4.7 Final Touches
4.8 What If It Doesn't Work ?
4.9 Exercises

5. Sensing the World Around Us
5.1 What You Need
5.2 Measuring Distances with an Ultrasonic Sensor
5.3 Increasing Precision Using Floating-Point Numbers
5.4 Increasing Precision Using a Temperature Sensor
5.5 Transferring Data Back to You r Computer Using Processing
5.6 Representing Sensor Data
5.7 Building the Application's Foundation
5.8 Implementing Serial Communication in Processing
5.9 Visualizing Sensor Data
5.10 What If It Doesn't Work ?
5.11 Exercises

6. Building a Motion-Sensing Game Controller
6.1 What You Need
6.2 Wiring Up the Accelerometer
6.3 Bringing You r Accelerometer to Life
6.4 Finding and Polishing Edge Values
6.5 Building You r Own Game Controller
6.6 Writing You r Own Game
6.7 More Projects
6.8 What If It Doesn't Work ?
6.9 Exercises

7. Tinkering with the Wii Nunchuk
7.1 What You Need
7.2 Wiring a Wi i Nunchuk
7.3 Talking to a Nunchuk
7.4 Building a Nunchuk Class
7.5 Using Our Nunchuk Class
7.6 Rotating a Colorful Cube
7.7 What If It Doesn't Work ?
7.8 Exercises

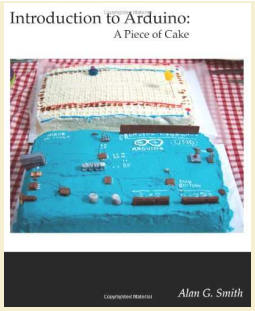
8. Networking with Arduino
8.1 What You Need
8.2 Using You r PC to Transfer Sensor Data to the Internet
8.3 Registering an Application with Twitter174
8.4 Tweeting Messages with Processing
8.5 Networking Using an Ethernet Shield
8.6 Emailing from the Command Line
8.7 Emailing Directly from an Arduino
8.8 Detecting Motion Using a Passive Infrared Sensor192
8.9 Bringing It All Together
8.10 What If It Doesn't Work ?
8.11 Exercises

9. Creating You r Own Universal Remote Control
9.1 What You Need
9.2 Understanding Infrared Remote Controls204
9.3 Grabbing Remote Control Codes
9.4 Building You r Own Apple Remote
9.5 Controlling Devices Remotely with Your Browser.
9.6 Building an Infrared Proxy
9.7 What If It Doesn't Work ?
9.8 Exercises

10. Controlling Motors with Arduino
10.1 What You Need
10.2 Introducing Motors
10.3 First Steps with a Servo Motor
10.4 Building a Blaminatr
10.5 What If It Doesn't Work ?
10.6 Exercises

III. Appendixes

A. Basics of Electronics
A.1 Current, Voltage, and Resistance
A.2 Learning How to Solder
B. Advanced Arduino Programming
B.1 The Arduino Programming Language
B.2 Bit Operations
C. Advanced Serial Programming
C.1 Learning More About Serial Communication
C.2 Serial Communication Using Various Programming Languages



Alan G Smith

Introduction to Arduino: A piece of cake
Введение в Arduino: пара пустяков

— Publisher: CreateSpace Independent Publishing Platform, 2011. — 170 pages

ISBN-10: 1463698348
ISBN-13: 978-1463698348

Эта книга отличается от других книг по Arduino тем, что не требует глубоких познаний в электронике и программировании.

С этой книгой вы сможете быстро освоить работу со звуком, светом, LCD, и т.п.

CONTENTS

1 Getting Started
1.1 What is a Microcontroller?
1.2 Install the Software
1.3 The Integrated Development Environment (IDE)
1.4 Our first circuit
1.5 Updated Circuit
1.6 Our First Program
1.7 Comments
1.8 Gotchas
1.9 Exercises

2 Making Light Patterns
2.1 "Blinky"
2.2 IF Statements
2.3 ELSE Statements
2.4 WHILE statements
2.5 What is truth(true)?
2.6 Combinations
2.7 FOR statements
2.8 Our New Circuit
2.9 Introducing Arrays
2.10 Exercises

3 Input
3.1 Pushbuttons
3.2 Potentiometers
3.3 RGB LEDs
3.4 Exercises

4 Sound
4.1 Our Circuit
4.2 Simple note
4.3 Music
4.4 Music with functions
4.5 Exercises

5 Making a digital thermometer
5.1 Serial Monitor
5.2 Measuring the temperature
5.3 Hooking up the LCD
5.4 Talking to the LCD
5.5 Bringing it all together
5.6 Exercises

6 Graphics (Pictures) on our LCD
6.1 Binary and Hex
6.2 Using graphics
6.3 Making a Chart
6.4 Exercises

7 Sensors Galore
7.1 Introduction
7.2 Photo Cell (Light Sensor)
7.3 Tilt Sensor
7.4 Reed Switch (Magnetic Field Detector)
7.5 Piezo Element (Vibration sensor)
7.6 Exercises

8 Making a rubber band gun
8.1 One Servo
8.2 Joystick
8.3 Pan/Tilt bracket
8.4 Adding a firing mechanism
8.5 Exercises

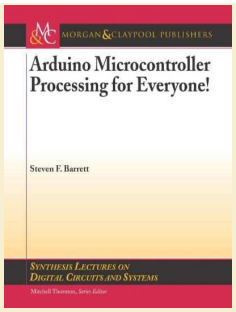
9 Make your own project!

10 Next Steps

A Arduino Reference
A.1 Structure
A.2 Variables
A.3 Functions
A.4 PCD8544 (LCD Controller) Library

B Parts in Kit
B.1 First used in Chapter 1
B.2 First used in Chapter 2
B.3 First used in Chapter 3
B.4 First used in Chapter 4
B.5 First used in Chapter 5
B.6 First used in Chapter 6
B.7 First used in Chapter 7
B.8 First used in Chapter 8

C Sample Solutions to Selected Exercises
C.1 Chapter 1 Solutions
C.2 Chapter 2 Solutions
C.3 Chapter 3 Solutions
C.4 Chapter 4 Solutions
C.5 Chapter 5 Solutions
C.6 Chapter 6 Solutions
C.7 Chapter 7 Solutions
C.8 Chapter 8 Solutions



Steven F. Barrett, Mitchell Thornton

Arduino Microcontroller Processing for Everyone!

Обработка данных в Arduino для всех!

— Publisher: Morgan and Claypool Publishers, 2010. — 344 pages
— Series: Synthesis Lectures on Digital Circuits and Systems

ISBN-10: 1608454371
ISBN-13: 978-1608454372

Книга содержит множество примеров программного и аппаратного обеспечения, чтобы помочь читателю в разработке разнообразных систем на Arduino. Предназначена для широкого круга студентов, любителей, ученых и инженеров.

CONTENTS

Preface

1. Getting Started
1.1 Overview
1.2 Getting Started
1.3 Arduino Duemilanove
1.3.1 Arduino host processor — the ATmega328
1.4 Example: Autonomous Maze Navigating Robot
1.4.1 Structure chart
1.4.2 UML activity diagrams
1.4.3 Arduino Duemilanove Systems
1.5 Arduino open source schematic
1.6 Other Arduino-based platforms
1.7 Extending the hardware features of the Arduino platform
1.8 Arduino Software
1.9 Arduino Duemilanove/ATmega328 hardware features
1.9.1 Memory
1.9.2 Port System
1.9.3 Internal Systems
1.10 Summary
1.11 References
1.12 Chapter Problems

2. Programming
2.1 Overview
2.2 The Big Picture
2.3 Anatomy of a Program
2.3.1 Comments
2.3.2 Include files
2.3.3 Functions
2.3.4 Program constants
2.3.5 Interrupt handler definitions
2.3.6 Variables
2.3.7 Main program
2.4 Fundamental programming concepts
2.4.1 Operators
2.4.2 Programming constructs
2.4.3 Decision processing
2.5 Arduino Development Environment
2.5.1 Background
2.5.2 Arduino Development Environment overview
2.5.3 Sketchbook concept
2.5.4 Arduino software, libraries, and language references
2.6 Application 1: Robot IR sensor
2.7 Application 2: Art piece illumination system
2.8 Summary
2.9 References
2.10 Chapter Problems

3. Embedded SystemsDesign
3.1 What is an embedded system?
3.2 Embedded system design process
3.2.1 Project Description
3.2.2 Background Research
3.2.3 Pre-Design
3.2.4 Design
3.2.5 Implement Prototype
3.2.6 Preliminary Testing
3.2.7 Complete and Accurate Documentation
3.3 Example: Blinky 602A autonomous maze navigating robot system design
3.4 Application: Control algorithm for the Blinky 602A Robot
3.5 Summary
3.6 References
3.7 Chapter Problems

4. Serial Communication Subsystem
4.1 Overview
4.2 Serial Communications
4.3 Serial Communication Terminology
4.4 Serial USART
4.4.1 System Overview
4.5 System Operation and Programming using Arduino Development Environment features
4.6 System Operation and Programming in C
4.6.1 Serial Peripheral Interface—SPI
4.7 SPI Programming in the Arduino Development Environment
4.8 SPI Programming in C
4.9 Two-wire Serial Interface—TWI
4.10 Application 1: SD/MMC card module extension via the USART
4.11 Application 2: Programming the Arduino Duemilanove ATmega328 via the ISP
4.11.1Programming Procedure
4.12 Summary
4.13 References
4.14 Chapter Problems

5. Analog toDigital Conversion (ADC)
5.1 Overview
5.2 Sampling, Quantization and Encoding
5.2.1 Resolution and Data Rate
5.3 Analog-to-Digital Conversion (ADC) Process
5.3.1 Transducer Interface Design (TID) Circuit
5.3.2 Operational Amplifiers
5.4 ADC Conversion Technologies
5.4.1 Successive-Approximation
5.5 The Atmel ATmega328 ADC System
5.5.1 Block Diagram
5.5.2 Registers
5.6 Programming the ADC using the Arduino Development Environment
5.7 Programming the ADC in C
5.8 Example: ADC Rain Gage Indicator
5.8.1 ADC Rain Gage Indicator using the Arduino Development Environment
5.8.2 ADC Rain Gage Indicator in C
5.8.3 ADC Rain Gage using the Arduino Development Environment—Revisited
5.9 One-bit ADC - Threshold Detector
5.10 Digital-to-Analog Conversion (DAC)
5.10.1DAC with the Arduino Development Environment
5.10.2DAC with external converters
5.10.3Octal Channel, 8-bit DAC via the SPI
5.11 Application: Art piece illumination system – Revisited
5.12 Summary
5.13 References
5.14 Chapter Problems

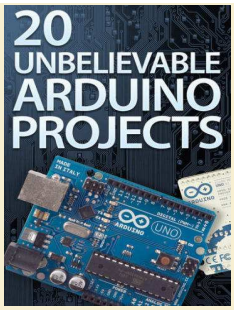
6. Interrupt Subsystem
6.1 Overview
6.2 ATmega328 Interrupt System

6.3 Interrupt Programming
6.4 Programming Interrupts in C and the Arduino Development Environment
6.4.1 External Interrupt Programming
6.4.2 Internal Interrupt Programming
6.5 Foreground and Background Processing
6.6 Interrupt Examples
6.6.1 Real Time Clock in C
6.6.2 Real Time Clock using the Arduino Development Environment
6.6.3 Interrupt Driven USART in C
6.7 Summary
6.8 References
6.9 Chapter Problems

7. Timing Subsystem
7.1 Overview
7.2 Timing related terminology
7.2.1 Frequency
7.2.2 Period
7.2.3 Duty Cycle
7.3 Timing System Overview
7.4 Applications
7.4.1 Input Capture — Measuring External Timing Event
7.4.2 Counting Events
7.4.3 Output Compare — Generating Timing Signals to Interface External Devices
7.4.4 Industrial Implementation Case Study (PWM)
7.5 Overview of the Atmel ATmega328 Timer System
7.6 Timer 0 System
7.6.1 Modes of Operation
7.6.2 Timer 0 Registers
7.7 Timer 1
7.7.1 Timer 1 Registers
7.8 Timer 2
7.9 Programming the Arduino Duemilanove using the built-in Arduino Development Environment Timing Features
7.10 Programming the Timer System in C
7.10.1 Precision Delay in C
7.10.2 PulseWidth Modulation in C
7.10.3 Input Capture Mode in C
7.11 Servo Motor Control with the PWM System in C
7.12 Summary
7.13 References
7.14 Chapter Problems

8. Atmel AVROperating Parameters and Interfacing
8.1 Overview
8.2 Operating Parameters
8.3 Battery Operation
8.3.1 Embedded system voltage and current drain specifications
8.3.2 Battery characteristics
8.4 Input Devices
8.4.1 Switches
8.4.2 Pullup resistors in switch interface circuitry
8.4.3 Switch Debouncing
8.4.4 Keypads
8.4.5 Sensors
8.4.6 LM34 Temperature Sensor Example
8.5 Output Devices
8.5.1 Light Emitting Diodes (LEDs)
8.5.2 Seven Segment LED Displays
8.5.3 Code Example
8.5.4 Tri-state LED Indicator
8.5.5 Dot Matrix Display
8.5.6 Liquid Crystal Character Display (LCD) in C
8.5.7 Liquid Crystal Character Display (LCD) using the Arduino Development Environment
8.5.8 High Power DC Devices
8.6 DC Solenoid Control
8.7 DC Motor Speed and Direction Control
8.7.1 DC Motor Operating Parameters
8.7.2 H-bridge direction control
8.7.3 Servo motor interface
8.7.4 Stepper motor control
8.7.5 AC Devices
8.8 Interfacing to Miscellaneous Devices
8.8.1 Sonalerts, Beepers, Buzzers
8.8.2 Vibrating Motor
8.9 Extended Example 1: Automated Fan Cooling System
8.10 Extended Example 2: Fine Art Lighting System
8.11 Extended Example 3: Flight Simulator Panel
8.12 Summary
8.13 References
8.14 Chapter Problems

A. ATmega328 Register Set
B. ATmega328Header File
Author's Biography
Index



www.instructables.com

20 Unbelievable Arduino Projects
20 невероятных проектов на Arduino

— Publisher: www.instructables.com, 2012. — 430 pages

Здесь двадцать удивительных проектов на Arduino: светодиодный куб 8x8x8; светодиодная матрица 24x6; робот, лазающий по деревьям; робот, играющий в шахматы и др.

Следуя шаг за шагом по главам книги, вы тоже сможете их повторить.

CONTENTS

Author and Copyright Notices
Disclaimer

LED Cube 8x8x8
Intro: LED Cube 8x8x8
Step 1: Skills required
Step 2: Component list
Step 3: Ordering components
Step 4: What is a LED cube
Step 5: How does a LED cube work
Step 6: The anatomy of a LED cube
Step 7: Cube size and IO port requirements
Step 8: IO port expansion, more multiplexing
Step 9: IO port expansion, alternative solution
Step 10: Power supply considerations
Step 11: Buy a power supply
Step 12: Build a power supply
Step 13: Choose your LEDs
Step 14: Choose your resistors
Step 15: Choose the size of your cube
Step 16: How to make straight wire
Step 17: Practice in small scale
Step 18: Build the cube: create a jig
Step 19: Build the cube: soldering advice
Step 20: Build the cube: test the LEDs
Step 21: Build the cube: solder a layer
Step 22: Build the cube: test the layer
Step 23: Build the cube: straigthen the pins
Step 24: Build the cube: bend the pins
Step 25: Build the cube: solder the layers together
Step 26: Build the cube: create the base
Step 27: Build the cube: mount the cube
Step 28: Build the cube: cathode risers
Step 29: Build the cube: attach cables
Step 30: Build the controller: layout
Step 31: Build the controller: clock frequency
Step 32: Build the controller: protoboard soldering advice
Step 33: Build the controller: Power terminal and filtering capacitors
Step 34: Build the controller: IC sockets, resistors and connectors
Step 35: Build the controller: Power rails and IC power
Step 36: Build the controller: Connect the ICs, 8bit bus + OE
Step 37: Build the controller: Address selector
Step 38: Build the controller: AVR board
Step 39: Build the controller: Transistor array
Step 40: Build the controller: Buttons and status LEDs
Step 41: Build the controller: RS-232
Step 42: Build the controller: Make an RS-232 cable
Step 43: Build the controller: Connect the boards
Step 44: Build the controller: Connect the cube
Step 45: Program the AVR: Set the fuse bits
Step 46: Program the AVR with test code
Step 47: Test the cube
Step 48: Program the AVR with real code
Step 49: Software: Introduction
Step 50: Software: How it works
Step 51: Software: IO initialization
Step 52: Software: Mode selection and random seed
Step 53: Software: Interrupt routine
Step 54: Software: Low level functions
Step 55: Software: Cube virtual space
Step 56: Software: Effect launcher
Step 57: Software: Effect 1, rain
Step 58: Software: Effect 2, plane boing
Step 59: Software: Effect 3, sendvoxels random Z
Step 60: Software: Effect 4, box shrinkgrow and woopwoop
Step 61: Software: Effect 5, axis updown randsuspend
Step 62: Software: Effect 6, stringfly
Step 63: Software: RS-232 input
Step 64: PC Software: Introduction
Step 65: PC Software: Cube updater thread
Step 66: PC Software: Effect 1, ripples
Step 67: PC Software: Effect 2, sidewaves
Step 68: PC Software: Effect 3, fireworks
Step 69: PC Software: Effect 4, Conway's Game of Life 3D
Step 70: Run the cube on an Arduino
Step 71: Hardware debugging: Broken LEDs
Step 72: Feedback

Power Laces- the Auto lacing shoe
Intro: Power Laces- the Auto lacing shoe
Step 1: Parts & Tools
Step 2: The Laces pt. 1
Step 3: The Laces pt. 2
Step 4: Servo Mounting Plate
Step 5: Construct the Motor Shield
Step 6: Mount the Servos, Battery, and Arduino
Step 7: Adding some electronics to the motor shield
Step 8: Connect the Laces to the Servos
Step 9: Upload the Arduino Sketch

Plantduino Greenhouse
Intro: Plantduino Greenhouse
Step 1: Plant Science 101
Step 2: Build a Garden/ Plant Seeds
Step 3: Build a Greenhouse: Step 1 materials
Step 4: Build a Greenhouse: Step 2 Build the Frame
Step 5: Build a Greenhouse: Step 4 Lay the plastic
Step 6: Build the Greenhouse: Step 5 Add the back and the door
Step 7: Build the Greenhouse: Step 7 Make it airtight/waterproof
Step 8: Build a Greenhouse: Step 8 Dig a Trench
Step 9: Watering System: Step 1 Materials
Step 10: Watering System: Step 2 Build a Relay Box
Step 11: Watering System: Step 3 Connect the Valve
Step 12: Watering System: Step 4 moisture sensors
Step 13: Watering System: Step 5 Write the Code
Step 14: Watering System: Step 6 Bring It All Together
Step 15: Plantduino: Step 1 materials
Step 16: Plantduino: Step 2 Schematics
Step 17: Plantduino: Step 3 Assembly Tips and Tricks
Step 18: Birdhouse: Creation and Installation
Step 19: Creating the Birdhouse Motherboard
Step 20: Video
Step 21: Final Thoughts/ Additional Reading

The EyeWriter 2.0
Intro: The EyeWriter 2.0
Step 1: Overview
Step 2: Parts list
Step 3: Software - openFrameworks & EyeWriter
Step 4: Software - Camera & Arduino
Step 5: Load Arduino sketch
Step 6: Hardware: Power Adapter
Step 7: Hardware: Infrared LED's
Step 8: Hacking the PS Eye camera - preparing

Step 9: Hacking the PS Eye camera - VSync
Step 10: Hacking the PS Eye camera - finishing
Step 11: Full Circuit
Step 12: Building a wood base
Step 13: Using EyeWriter Software - Setup & Tracking Screen
Step 14: Using EyeWriter Software - Calibration Screen
Step 15: Using EyeWriter Software - Catch Me
Step 16: Using EyeWriter Software - Drawing
Step 17: Using EyeWriter Software - Typing
Step 18: Using EyeWriter Software - Pong

Twitter Mood Light - The World's Mood in a Box
Intro: Twitter Mood Light - The World's Mood in a Box
Step 1: How it works
Step 2: All you need is..
Step 3: Connect the Arduino and WiFly to a computer
Step 4: Connecting the LED
Step 5: Choosing good search terms
Step 6: Download the code
Step 7: Programming step 1: SPI UART
Step 8: Programming step 2: Connecting to a Wireless Network
Step 9: Programming step 3: Searching Twitter with TCP/IP port 80
Step 10: Programming step 4: RGB LED
Step 11: Programming 5: Computing the World Mood
Step 12: Building the Box
Step 13: Enjoy!

Flamethrowing Jack-O'-Lantern
Intro: Flamethrowing Jack-O'-Lantern
Step 1: Go get stuff
Step 2: Cut a cap
Step 3: Gut it
Step 4: Design a face
Step 5: Trace
Step 6: Cut
Step 7: Bend
Step 8: Brackets
Step 9: Drill holes
Step 10: Attach things
Step 11: Candle mount
Step 12: Battery adapter
Step 13: Program the Receiver
Step 14: Program the transmitter
Step 15: Switch
Step 16: Antenna
Step 17: Wire the transmitter
Step 18: Power
Step 19: Case closed
Step 20: Wire the reciever
Step 21: Put it together
Step 22: Wire the motor
Step 23: Put it in the pumpkin
Step 24: Candle
Step 25: Fire!

Make a 24X6 LED matrix
Intro: Make a 24X6 LED matrix
Step 1: Getting All The Right Things
Step 2: How it works?
Step 3: Schematics
Step 4: Soldering The LEDs
Step 5: Programming The Display
Step 6: We Are Done!

Secret Knock Detecting Door Lock
Intro: Secret Knock Detecting Door Lock
Step 1: Tools, Supplies, And Skills
Skills
Tools:
Materials
Electronics:
Case:
Step 2: Program The Arduino
Step 3: Lay Out And Test The Circuit
Step 4: Prepare The Case
Step 5: Make The Lock Turning Clamp
Step 6: Make The Knock Detector Spring
Step 7: Soldering The Circuits
Step 8: Assembling The Case
Step 9: Mounting, Testing, and Use
Step 10: Epilog: Changes And Improvements
Did you build this?
Masters of Secret Knocks:

turn signal biking jacket
Intro: Turn signal biking jacket
Step 1: Supplies
Step 2: Design
Step 3: Sew your power supply and LilyPad to your jacket
Step 4: Test your stitching
Step 5: Sew on your turn signal LEDs
Step 6: Sew in your control switches
Step 7: Sew in your indicator LEDs
Step 8: Program your jacket

Tree Climbing Robot
Intro: Tree Climbing Robot
Step 1: Design
Step 2: Tools and Materials
Step 3: Motor Controller
Step 4: Power
Step 5: Power, cont
Step 6: Legs
Step 7: Feet
Step 8: Motor Hubs
Step 9: Building the Frame
Step 10: Frame, cont
Step 11: Electronics Platform
Step 12: Rotation Sensors
Step 13: Backbone Motor
Step 14: Mounting the Spine
Step 15: Mounting the Spine, cont
Step 16: Linear Slides
Step 17: Wiring the Robot
Step 18: Limit Switches
Step 19: Battery Holders
Step 20: Programming

Rave Rover - Mobile Dance Stage
Intro: Rave Rover - Mobile Dance Stage
Step 1: Starting the Build
Step 2: Cutting Parts
Step 3: Fitting the floor
Step 4: Getting LEDs ready
Step 5: Installing the LEDs
Step 6: Adding the Frame
Step 7: LED Color Check and Testing
Step 8: Gathering More Materials
Step 9: Frame Building
Step 10: Getting frames to fit..
Step 11: Mounting Components
Step 12: More Mounting..
Step 13: Pole Mounting
Step 14: Finishing the Electronics..
Step 15: Drive Test!
Step 16: Installing Floor
Step 17: Final touches
Step 18: Speaker Install
Step 19: Finally Done!
Step 20: Where to find parts..
Step 21: Party Time!

Type Case, the making of a low-resolution display
Intro: Type Case, the making of a low-resolution display
Step 1: The idea
Step 2: Simulations
Step 3: Development = solving problems
Step 4: The build
Step 5: The documentation process

Sigh Collector
Intro: Sigh Collector
Step 1: Material Needed
Step 2: Build and Program Circuit. Hack into Air Pump
Step 3: Build the Sigh Collector main unit
Step 4: Make the air bladder
Step 5: Combine electronics with main unit. Install Check Valve and Pump
Step 6: Build carrying case, Sew handle
Step 7: Build and Program circuit for sigh detection. Assemble electronics into carrying case
Step 8: Cut and Sew chest strap and attach the stretch sensor
Step 9: A word on Wireless
Step 10: Finished

Make a Fire Breathing Animetronic Pony from FurReal Butterscotch or S'Mores
Intro: Make a Fire Breathing Animetronic Pony from FurReal Butterscotch or S'Mores
Step 1: Get it before you hack it
Step 2: What you will need
Step 3: Removing the skin: Head first
Step 4: Removing Skin: ENT
Step 5: Remove Skin: Straight from the horses mouth
Step 6: Remove Skin: The body

Step 7: Removing the skin: The legs
Step 8: Removing the skin: the Neck
Step 9: Removing the face
Step 10: Getting access to the Circuit board in the lower body
Step 11: Cutting the power to the Microcontroler
Step 12: Tapping power for the Arduino
Step 13: Tapping the lines into the motor control circuit
Step 14: Taping into the encoders
Step 15: Getting the morors and sensors connected to the arduino
Step 16: Connecting a wii nunchuck into the system
Step 17: The Arduino Code
Step 18: Getting the fuel to the head
Step 19: Building an ignition system
Step 20: Remote fuel trigger
Step 21: Follow up

Tweet-a-watt - How to make a twittering power meter
Intro: Tweet-a-watt - How to make a twittering power meter
Step 1: Make it!
Step 2: Prep
Step 3: Make the Receiver
Step 4: Configure
Step 5: Solder the Transmitter - parts list
Step 6: Transmitter Schematic
Step 7: Assemble and create the transmitter - 1
Step 8: Assemble and create the transmitter - 2
Step 9: Assemble and create the transmitter - 3
Step 10: Assemble and create the transmitter - 4
Step 11: Assemble and create the transmitter - 5
Step 12: Software
Step 13: Expand
Step 14: Design - overview
Step 15: Design - listen
Step 16: Design - store
Step 17: Design - graph
Step 18: Resources
Step 19: Download

Bubblesteen Bubble Machine
Intro: Bubblesteen Bubble Machine
Step 1: Things you will need
Step 2: Dealing with the micro controller
Step 3: Putting it together
Step 4: Arduino & motor shield platform
Step 5:
Step 6: Additional photos

Arduino R/C Lawnmower (painted)
Intro: Arduino R/C Lawnmower (painted)
Step 1: Setting up
Step 2: The Motor Driver
Step 3: The Wheels
Step 4: The Frame part A
Step 5: The Frame part B
Step 6: Mounting the motors
Step 7: Mounting the mower deck
Step 8: Select and Install the batteries
Step 9: Mount the electronics
Step 10: The Code
Step 11: More Videos

How to Build an Arduino Powered Chess Playing Robot
Intro: How to Build an Arduino Powered Chess Playing Robot
Step 1: Parts and Materials
Step 2: Design and Code Explanation
Step 3: Mounting the Drawer Bearings (Y Axis)
Step 4: Building the Motor Mount (Y Axis)
Step 5: Installing the Rack Gears (Y Axis)
Step 6: Wiring and Mounting the Motor (Y Axis)
Step 7: Mounting the Crossbars (X Axis)
Step 8: Mounting the Drawer Bearing and Rack Gears (X Axis)
Step 9: Attaching the Magnet to the Servo (X Axis)
Step 10: Wiring and Mounting the Motor (X Axis)
Step 11: Wiring the Sensors
Step 12: Place the Magnets
Step 13: Code, Final Assembly + Reflection

SITWAY
Intro: SITWAY
Step 1: MATERIALS AND COSTS
Step 2: Salvaging parts from the donor wheelchair
Step 3: Build the frame and mount the wheels and motors
Step 4: STEERING CONTROLLER
Step 5: ELECTRONICS
Step 6: WIRING
Step 7: MOTOR TEST
Step 8: THE FIRST TEST RIDE
Step 9: ADDING 3D PRINTED OBJECTS
Step 10: CONCLUSION

A Makers Wedding - Photo booth
Intro: A Makers Wedding - Photo booth
Step 1: How it works
Step 2: Software and Trigger Button
Step 3: Booth Design
Step 4: Cut The Panels
Step 5: Bottom Panel - Tripod Mount
Step 6: Box Construction
Step 7: Adding Components
Step 8: Testing
Step 9: Details and Finishing - Part 1
Step 10: Details and Finishing - Part 2
Step 11: Usage

издательство
БХВ-Петербург
www.bhv.ru

Отдел оптовых поставок
E-mail: opt@bhv.spb.su



Соммер У.

Программирование микроконтроллерных плат Arduino/Freeduino

256 стр., ISBN 978-5-9775-0727-1

Рассмотрено программирование микроконтроллерных плат Arduino/Freeduino. Описана структура и функционирование микроконтроллеров, среда программирования Arduino, необходимые инструменты и комплектующие для проведения экспериментов.

Подробно рассмотрены основы программирования плат Arduino: структура программы, команды, операторы и функции, аналоговый и цифровой ввод/вывод данных. Изложение материала сопровождается более 80 примерами по разработке различных устройств: реле температуры, школьных часов, цифрового вольтметра, сигнализации с датчиком перемещения, выключателя уличного освещения и др. Для каждого проекта приведен перечень необходимых компонентов, монтажная схема и листинги программ.

На FTP-сервере издательства выложены исходные коды примеров из книги, технические описания, справочные данные, среда разработки, утилиты и драйверы.

В книге рассмотрены:

- Общие сведения о микроконтроллерах
- Программирование микроконтроллера на платах Arduino/Freeduino
- Разработка и тестирование собственных приложений
- Более 80 практических проектов: настройка и применение микросхемы-адаптера FT232RL, задержка включения и выключения, реле температуры, измеритель емкости, школьные часы, 6-канальный цифровой вольтметр, управление вентилятором, обмен данными между компьютером и платой Arduino, сенсорный датчик, запись данных с помощью программы построения графиков Stamp PLOT, цифровой осциллограф с памятью, сигнализация с датчиком перемещения, выключатель уличного освещения, свет свечи, музыкальный проигрыватель, графопостроитель, последовательный ввод/вывод данных, эксперименты с ЖК-дисплеями и многое другое.

УДК 681.3.068
ББК 32.973.26-018.1
С61

Соммер У.

С61 Программирование микроконтроллерных плат Arduino/Freeduino. — СПб.: БХВ-Петербург, 2012. — 256 с.: ил. — (Электроника)

ISBN 978-5-9775-0727-1

Рассмотрено программирования микроконтроллерных плат Arduino/Freduino. Описана структура и функционирование микроконтроллеров, среда программирования Arduino, необходимые инструменты и комплектующие для проведения экспериментов. Подробно рассмотрены основы программирования плат Arduino: структура программы, команды, операторы и функции, аналоговый и цифровой ввод/вывод данных. Изложение материала сопровождается более 80 примерами по разработке различных устройств: реле температуры, школьных часов, цифрового вольтметра, сигнализации с датчиком перемещения, выключателя уличного освещения и др. Для каждого проекта приведен перечень необходимых компонентов, монтажная схема и листинги программ. На FTP-сервере издательства выложены исходные коды примеров из книги, технические описания, справочные данные, среда разработки, утилиты и драйверы.

Для радиолюбителей

УДК 681.3.068
ББК 32.973.26-018.1

Die berechtigte Übersetzung von deutschsprachiges Buch Arduino. Mikrocontroller-Programmierung mit Arduino/Freeduino, ISBN: 978-3-645-65034-2. Copyright © 2010 Franzis Verlag GmbH, 85586 Poing. Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträger oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt. Die Russische Übersetzung ist von BHV St. Petersburg verbreitet, Copyright © 2011.

Авторизованный перевод немецкой редакции книги Arduino. Mikrocontroller-Programmierung mit Arduino/Freeduino, ISBN: 978-3-645-65034-2. Copyright © 2010 Franzis Verlag GmbH, 85586 Poing. Все права защищены, включая любые виды копирования, в том числе фотомеханического, а также хранение и тиражирование на электронных носителях. Изготовление и распространение копий на бумаге, электронных носителях данных и публикация в Интернете, в том числе в формате PDF, возможны только при наличии письменного согласия Издательства Franzis. Нарушение этого условия преследуется в уголовном порядке. Перевод на русский язык "БХВ-Петербург" © 2011.

Arduino™ является зарегистрированной торговой маркой Arduino LLC и аффилированных компаний.

Группа подготовки издания:

Николай Тверских

Подписано в печать 30.09.11.
Формат 70×100$^{1}/_{16}$. Печать офсетная. Усл. печ. л. 20,32.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

ISBN 978-3-645-65034-2 (нем.)
ISBN 978-5-9775-0727-1 (рус.)

© 2010 Franzis Verlag GmbH, 85586 Poing
© Перевод на русский язык "БХВ-Петербург", 2011

Оглавление

Предисловие. 1

Введение 3

Глава 1. Общие сведения о микроконтроллерах 5

1.1. Структура и принцип работы контроллера. 6
1.1.1. Центральный процессор. 6
1.1.2. Оперативная память и память программ 7
1.2. Внешние устройства. 8
1.3. Сравнение технологий RISC и CISC 8
1.3.1. Технология CISC. 8
1.3.2. Технология RISC. 10
1.3.3. Выводы 10

Глава 2. Программирование микроконтроллеров. 11

2.1. Что такое программа?. 11
2.2. Программирование на C. 11

Глава 3. Краткий обзор семейства микроконтроллеров Arduino 13

3.1. Плата Arduino Mega. 14
3.2. Плата Arduino Duemilanove. 15
3.3. Плата Arduino Mini 15
3.4. Плата Arduino Nano 16
3.5 Плата Arduino Pro Mini. 16
3.6. Плата Arduino Pro 17
3.7. Плата LilyPad. 17
3.8. USB-адаптер. 18

Глава 4. Платы расширения Arduino 19

4.1. Плата расширения Arduino ProtoShield. 19
4.2. Плата расширения Ardumoto 20
4.3. Плата расширения TellyMate 21
4.4. Плата расширения ArduPilot. 22
4.5. Модули XBeeZNet. 22
4.6. Плата расширения Ethernet 23

Глава 5. Комплектующие изделия 25

5.1. Список основных комплектующих 25
5.2. Список деталей для дополнительных экспериментов 25

5.3. Экспериментальная плата Freeduino 26
5.4. Экспериментальная плата микроконтроллера Freeduino. 26
5.5. Электропитание. 27
5.6. Кнопка Reset. 27
5.7. ISP-подключение 27
5.8. Замечания по технике безопасности 29

Глава 6. Электронные компоненты и их свойства 31

6.1. Светодиоды 31
6.2. Резисторы 32
6.3. Конденсаторы. 33
6.4. Транзисторы. 34
6.5. Диод 34
6.6. Акустический пьезопреобразователь ("пищалка") 35
6.7. Монтажный провод. 35
6.8. Кнопка. ... 35
6.9. Потенциометр. 36
6.10. Фоторезистор 36
6.11. Монтажная панель с контактными гнездами. ... 37

Глава 7. Предварительная подготовка 39

7.1. Установка драйвера 39
7.2. Вспомогательная программа MProg для FT232RL. 40
7.3. Программирование микросхемы FT232R с помощью MProg 44
7.4. Установка программного обеспечения Arduino. ... 45

Глава 8. Среда разработки Arduino. ... 47

8.1. Установки в Arduino-IDE 48
8.2. Наша первая программа "ES_Blinkt" 50
8.3. Что мы сделали?. 52

Глава 9. Основы программирования Arduino 55

9.1. Биты и байты. 55
9.2. Базовая структура программы 56
9.2.1. Последовательное выполнение программы. ... 56
9.2.2. Прерывание выполнения программы. ... 57
9.3. Структура программы Arduino 57
9.4. Первая программа с Arduino. ... 58
9.5. Команды Arduino и их применение. ... 59
9.5.1. Комментарии в исходном тексте 59
9.5.2. Фигурные скобки *{ }*. 60
9.5.3. Точка с запятой *;*. 60
9.5.4. Типы данных и переменные. ... 60
9.5.5. Имя переменной. 60
9.5.6. Локальные и глобальные переменные 61
9.5.7. Различные типы данных. ... 61
9.5.8. Операторы 65
9.5.9. Директива *#define*. 66

9.5.10. Управляющие конструкции . . 66
9.5.11. Циклы. . 71
9.5.12. Функции и подпрограммы . 75
9.5.13. Функции преобразования типа . 78
9.5.14. Математические функции . . 79
9.5.15. Последовательный ввод/вывод. . 86
9.5.16. Как функционирует последовательный интерфейс?. . . 93
9.5.17. Программная эмуляция UART. . 96
9.5.18. Конфигурация входа/выхода и установка порта . . . 97
9.5.19. Аналоговый ввод данных и АЦП . . 103
9.5.20. Аналоговый выход ШИМ . . 105
9.6. Некоторые специальные функции. . 110
Установка паузы с помощью *delay*. . 110
Функции случайных чисел . . . 110
Сколько времени прошло? . . . 113

Глава 10. Дальнейшие эксперименты с Arduino . . 115

10.1. Регулятор уровня яркости светодиода с транзистором . . . 115
10.2. Плавное мигание . . . 117
10.3. Подавление дребезга контактов кнопок. . 120
10.4. Задержка включения . . . 124
10.5. Задержка выключения . . . 126
10.6. Светодиоды и Arduino . . . 127
10.7. Подключение больших нагрузок . . 130
10.8. ЦАП на основе ШИМ-порта . . 132
10.9. С музыкой все веселей. . . 136
10.10. Романтичный свет свечи с помощью микроконтроллера . . . 139
10.11. Контроль персонала на проходной. . 140
10.12. Часы реального времени . . 143
10.13. Программа школьных часов . . 144
10.14. Управление вентилятором . . 148
10.15. Автомат уличного освещения. . 151
10.16. Сигнализация. . . 153
10.17. Кодовый замок . . . 155
10.18. Измеритель емкости с автоматическим выбором диапазона . . . 159
10.19. Профессиональное считывание сопротивления потенциометра . . 162
10.20. Сенсорный датчик. . . 164
10.21. Конечный автомат . . . 166
10.22. 6-канальный вольтметр на основе Arduino . . 169
10.23. Программирование самописца напряжения . . 171
10.24. Осциллограф с памятью на основе Arduino. . 173
10.25. Программа StampPlot — бесплатный профессиональный регистратор данных . . . 175
10.26. Управление через VB.NET. . 179
10.27. Реле температуры. . . 181

Глава 11. Шина I^2C. . . 185

11.1. Передача бита. . . 186
11.2. Состояние "СТАРТ" . . . 186
11.3. Состояние "СТОП" . . . 186

11.4. Передача байта . . . 186
11.5. Подтверждение. . . . 187
11.6. Адресация . . . 187
11.7. 7-битовая адресация. . . . 187

Глава 12. Arduino и температурный датчик LM75 с I²C-шиной . . . 189

Глава 13. Расширитель порта I²C с PCF8574 . . . 193

Глава 14. Ультразвуковой датчик для определения дальности . . . 197

14.1. Ультразвуковой датчик SRF02 . . . 197
14.2. Считывание данных . . . 198

Глава 15. Сопряжение платы Arduino с GPS . . . 201

15.1. Сколько требуется спутников? . . . 202
15.2. Как подключить GPS к Arduino? . . . 202
15.3. GPS-протокол . . . 203

Глава 16. Сервопривод с платой Servo для Arduino . . . 209

16.1. Как функционирует сервопривод? . . . 209
16.2. Подключение привода к Arduino . . . 210

Глава 17. Жидкокристаллические дисплеи . . . 213

17.1. Поляризация дисплеев . . . 214
17.2. Статическое управление и мультиплексный режим . . . 214
17.3. Угол обзора . . . 215
17.4. Отражающие, пропускающие и полупрозрачные ЖКИ . . . 215
17.5. Установка контрастности дисплея . . . 216
17.6. Набор отображаемых символов. . . . 217
17.7. Расположение выводов распространенных ЖКИ . . . 218
17.8. Управление дисплеем от микроконтроллера. . . . 220
17.9. Инициализация дисплеев . . . 220
17.10. Подключение дисплея к Arduino . . . 222
17.11. Первый эксперимент с ЖКИ. . . . 223
17.12. Как же все работает? . . . 226

ПРИЛОЖЕНИЯ . . . 229

Приложение 1. Соответствие выводов Arduino и ATmega . . . 231

Приложение 2. Escape-последовательности. . . . 232

Приложение 3. Таблица ASCII . . . 234

Приложение 4. Перечень фирм-поставщиков компонентов . . . 239

Приложение 4. Перечень фирм-поставщиков компонентов . . . 239

Приложение 5. Описание компакт-диска . . . 240

Предметный указатель . . . 241

Предисловие

Многим нелегко сделать первый шаг в области электроники и программирования микроконтроллеров. Для освоения большинства систем микроконтроллеров новичку сначала требуется перевернуть горы литературы, а также прочитать и понять непростые технические паспорта. Среды программирования, как правило, довольно сложны и рассчитаны на профессиональных программистов. Таким образом, доступ в мир микроконтроллеров остается для некоторых навсегда закрытым.

Arduino — это простая для освоения платформа с открытым кодом на основе встроенного микроконтроллера и среды разработки с программным интерфейсом API для микроконтроллеров. Для взаимодействия между человеком и микроконтроллером могут присоединяться различные аналоговые и цифровые датчики, которые регистрируют состояние окружающей среды и передают данные в микроконтроллер. Микроконтроллер обрабатывает входящие данные, а программа выдает новые данные в виде аналоговых или цифровых значений. В результате открываются широкие горизонты для творчества.

В распоряжении разработчика предоставлены готовые программы и библиотеки функций среды программирования Arduino. Комбинируя аппаратные и программные средства, вы сможете с помощью этой книги связать наш реальный мир с миром микроконтроллера, который состоит из битов и байтов.

Желаю приятно провести время при чтении книги и проведении экспериментов!

Улли Соммер (Ulli Sommer)

Введение

Книга представляет собой учебный курс программирования микроконтроллеров. Вы познакомитесь со структурой и принципом действия микроконтроллера, изучите среду программирования Arduino, узнаете о необходимых инструментах и комплектующих для проведения экспериментов. Целая глава книги посвящена основам программирования плат Arduino. Здесь подробно описывается структура программы, команды, операторы и функции, аналоговый и цифровой ввод и вывод данных. Изложение материала сопровождается многочисленными практическими примерами. Вполне вероятно, что экспериментируя, вы сможете изобрести что-то новое в области микроконтроллерной технологии. Не останавливайтесь на достигнутом, старайтесь усовершенствовать конструкцию устройства и код программы.

Книга состоит из 17 глав и 5 приложений. Описано более 80 различных устройств на основе платы Arduino. Для каждого эксперимента приведен перечень необходимых компонентов, монтажная схема макета и листинги программ.

Подготовка к экспериментам. Для предлагаемых экспериментов потребуется всего несколько простых и доступных компонентов из ящика с радиодеталями. Кое-что, возможно, придется приобрести специально. В *приложении 4* указан перечень фирм-поставщиков электронных компонентов.

Для большинства экспериментов не нужны ни батареи, ни внешний источник питания.

Очень полезным в работе будет многофункциональный измерительный прибор (мультиметр) и/или интерфейс к компьютеру. С этими средствами вы сможете провести дополнительные эксперименты и узнать много полезного. Пригодится и стандартный аккумулятор типоразмера AA (Mignon) или AAA (Micro).

Лицензия GPL. Вы можете поделиться своими собственными программами с другими пользователями Интернета. Приведенные примеры программ имеют открытую лицензию GPL (General Public License). Таким образом, вы имеете право на изменение программ, в соответствии с условиями GPL, их публикацию и предоставление другим пользователям, если распространяете свои программы также под лицензией GPL.

Требования к системе. Персональный компьютер не ниже Pentium III, ОС Windows 98SE/ME/XP/Vista/Windows 7, Linux, Macintosh, дисковод компакт-дисков, платформа Java.

Обновления и поддержка. Платформа Arduino постоянно совершенствуется. Обновления можно загрузить бесплатно с Web-сайта **www.arduino.cc** (вы платите только за доступ в Интернет).

О компакт-диске к книге. Материал прилагаемого к книге компакт-диска можно скачать по ссылке **ftp://85.249.45.166/9785977507271.zip**. Ссылка также доступна на странице книги на сайте **www.bhv.ru**.

Компакт-диск содержит различные программы, инструменты для программирования, технические паспорта и принципиальные схемы, а также коды примеров из книги. Использование этих материалов облегчает работу с книгой. Описание компакт-диска приведено в *приложении 5*.

Глава 1

Общие сведения о микроконтроллерах

Перед тем как начать работать с аппаратной вычислительной платформой Arduino, важно получить общие сведения о микроконтроллерах. Микроконтроллеры применяются, прежде всего, для автоматизации в метрологии, технике управления и автоматического регулирования. Преимущество микроконтроллеров состоит в том, что можно эффективно и с малыми затратами измерять и интерпретировать физические величины, чтобы потом принимать требуемые решения и выполнять необходимые действия.

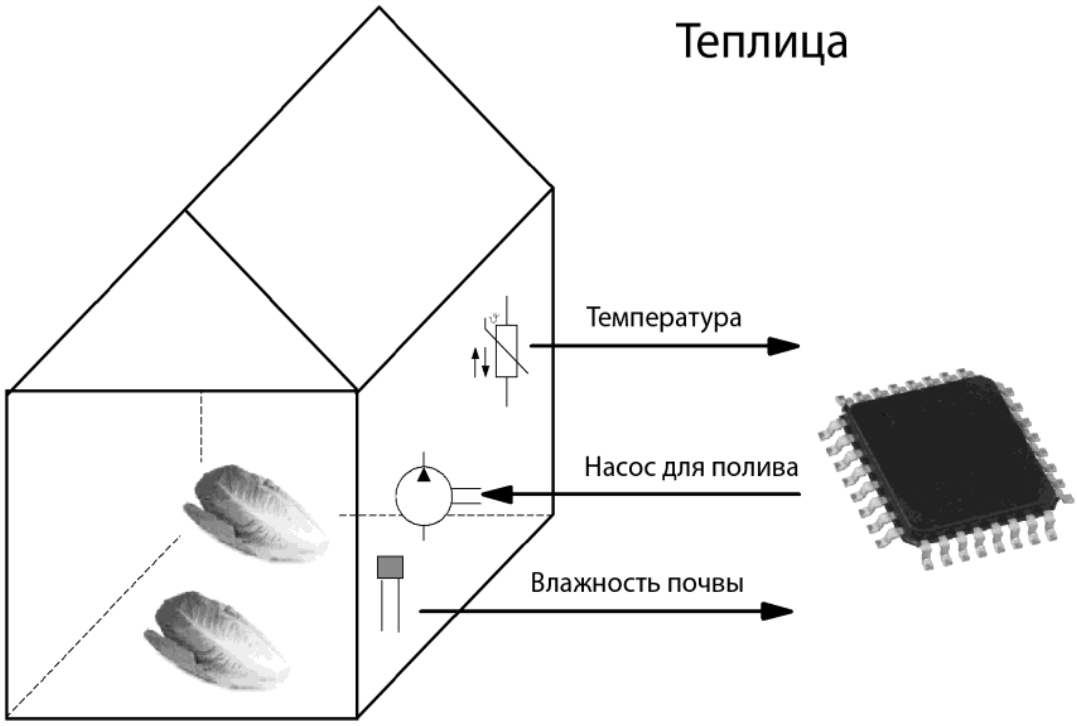


Рис. 1.1. Система управления климатом теплицы

Область возможных приложений микроконтроллеров чрезвычайно обширна: от частного домохозяйства (например, для управления теплицей или освещением) до промышленного производства, где могут обслуживаться и эксплуатироваться комплексные устройства, управляемые системами микроконтроллеров. На рис. 1.1

приведен типичный пример обработки данных для управления оросительной установкой теплицы. Контроллер фиксирует данные о температуре окружающей среды и влажности почвы, полученные от датчиков. Результаты измерения далее подвергаются логической обработке в микроконтроллере. Затем формируются сигналы управления насосом для полива.

1.1. Структура и принцип работы контроллера

Контроллер представляет собой, по сути, микрокомпьютер и содержит все присущие ему основные модули (рис. 1.2). Стандартные блоки каждого микроконтроллера — это центральный процессор (CPU), оперативная память (RAM), а также память программ (Flash-память) и внешние устройства.

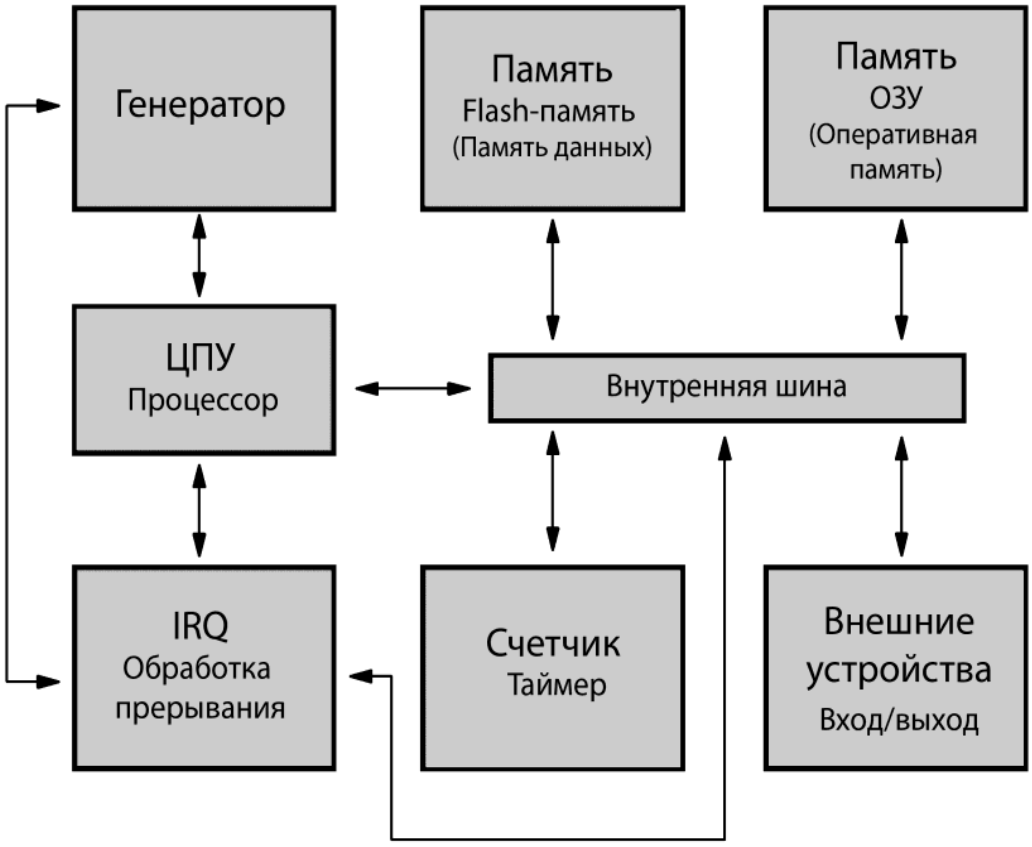


Рис. 1.2. Упрощенная структура микроконтроллера

1.1.1. Центральный процессор

Основное функциональное устройство микроконтроллера — центральный процессор (CPU — Central Processing Unit). Его можно сравнить с "мозгом" микроконтроллера. Сигналы в нем представлены в цифровой форме и над ними выполняются арифметические и логические операции.

1.1.2. Оперативная память и память программ

Оперативная память и память программ традиционно рассматриваются отдельно. Программа пользователя, т. е. наша собственная программа, которую мы сами писали, сохраняется в энергонезависимой Flash-памяти программ (рис. 1.3). В зависимости от типа контроллера память программ может занимать объем от нескольких килобайт до мегабайт. Кроме того, в некоторых вычислительных системах можно увеличить память программ, подключая внешние Flash-накопители.

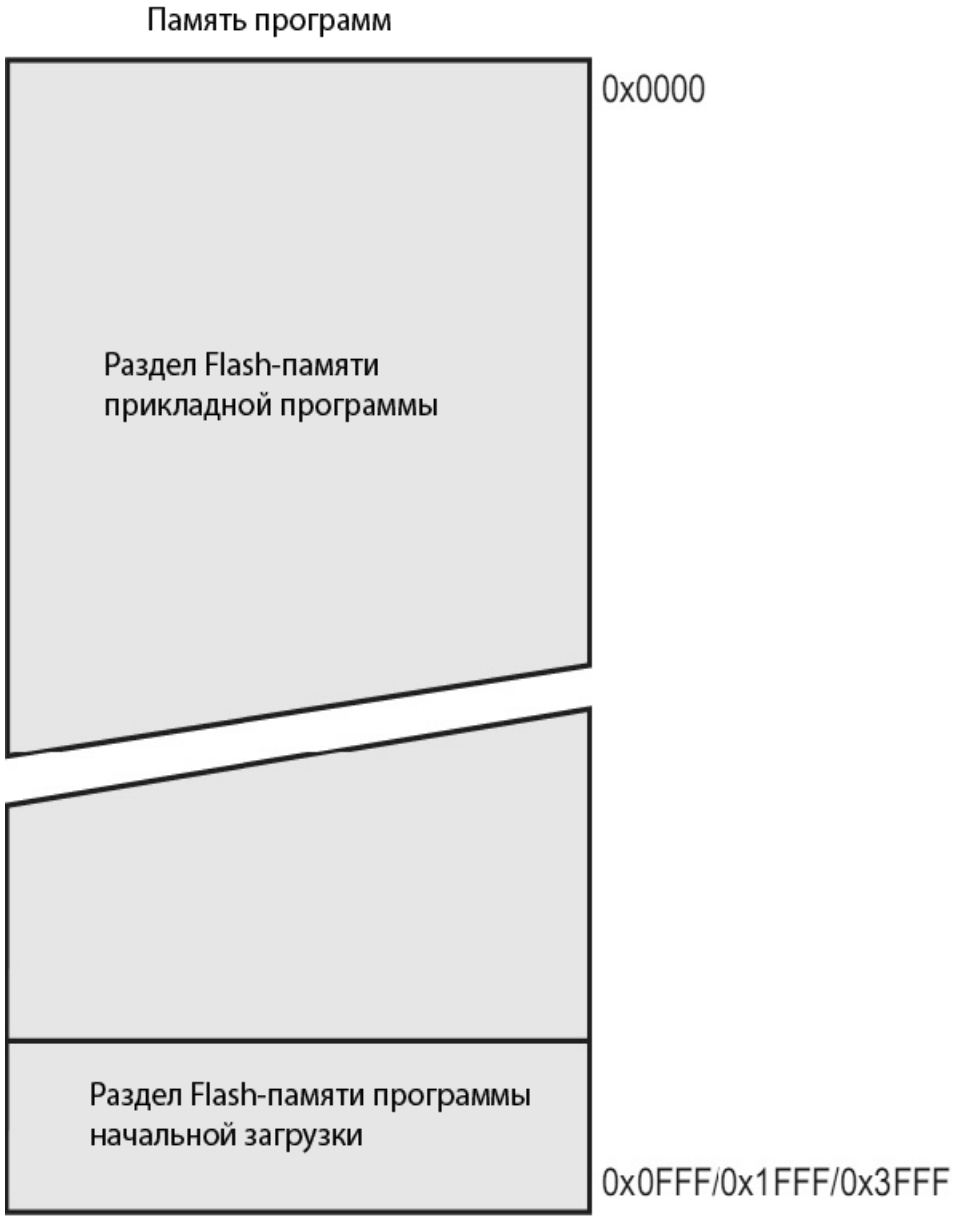


Рис. 1.3. Flash-память микроконтроллера ATmega168PA (источник: технический паспорт компании ATMEL)

Оперативная память служит для временного хранения различных промежуточных данных. Здесь хранятся и результаты вычислений, полученные во время выполнения программы.

Назначение оперативной памяти ОЗУ (RAM — Random Access Memory) — возможность быстрого обращения к ограниченному количеству данных. Ее объем, как правило, значительно меньше, а быстродействие намного больше, чем Flash-памяти.

Значения записываются и хранятся в ОЗУ во время выполнения и энергозависимы в отличие от Flash-памяти, т. е. после перезагрузки контроллера содержимое ОЗУ полностью стирается. На рис. 1.4 изображена структура ОЗУ микроконтроллера ATmega168PA.

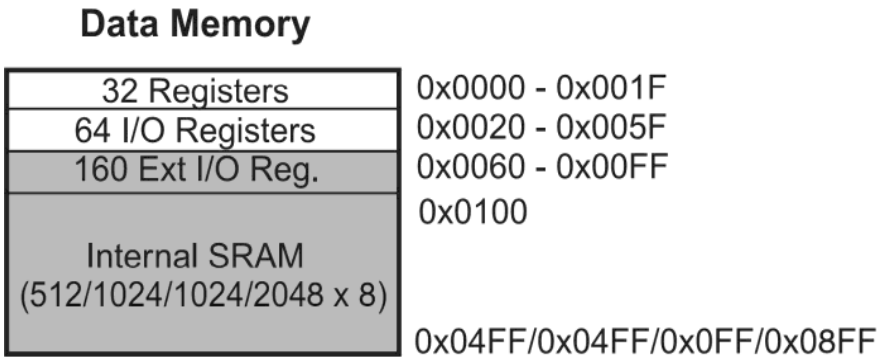


Рис. 1.4. RAM-память микроконтроллера ATmega168PA (источник: технический паспорт ATMEL)

1.2. Внешние устройства

Внешними (периферийными) устройствами часто называют все компоненты микроконтроллера, кроме центрального процессора и памяти. В частности, к ним относятся внешние интерфейсы, например, цифровые входы и выходы (Input/Output — сокращенно I/O). Большинство микроконтроллеров снабжены также различными цифровыми и аналоговыми входами и выходами.

1.3. Сравнение технологий RISC и CISC

Рассмотрение RISC- и CISC-технологий — это уже более глубокий взгляд на цифровую и микроконтроллерную технику. Сразу заметим, что контроллеры AVR для Arduino базируются на технологии RISC. Кратко опишем технологии RISC и CISC.

1.3.1. Технология CISC

При технологии CISC- в ОЗУ-загружается и программа, и данные. Говорят также о том, что код программы и данные делят между собой одну и ту же область памяти. Это имело смысл, в частности, в первых вычислительных системах, т. к. оперативная память была дорога.

Для микроконтроллера гораздо более важный отличительный признак — это структура команд. Компьютер с CISC-технологией располагает большим ассортиментом очень узкоспециализированных команд. В цифровой технике команда — это последовательность определенных байтов. Один байт может принимать 256 (от 0 до 255) различных состояний. Чтобы реализовать более 256 различных команд, нужны дополнительные байты. Таким образом, специальная команда состоит из нескольких, например пяти, байтов. Загрузка этой команды продолжается дольше, чем короткой однобайтовой команды.

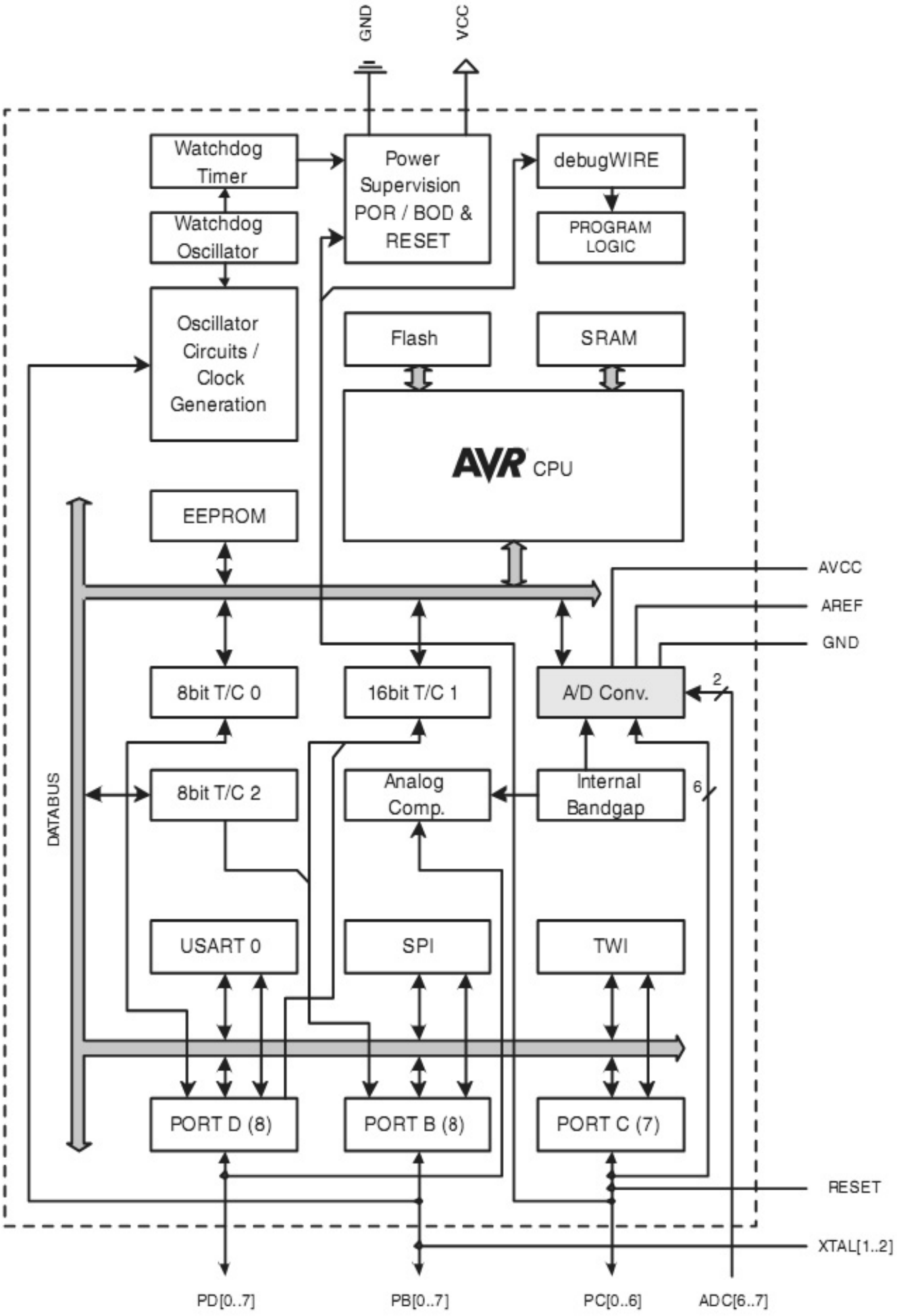


Рис. 1.5. Блок-схема микроконтроллера
(источник: технический паспорт компании ATMEL)

1.3.2. Технология RISC

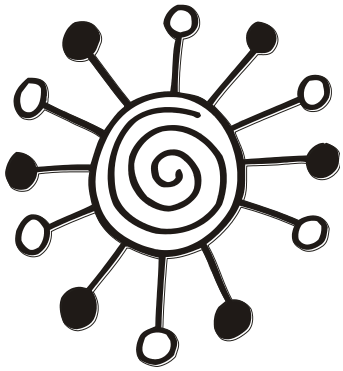
Было установлено, что в CISC-компьютерах, как правило, около 90% исходного кода программ состоит только из примерно 30 различных команд. Так возникла мысль реализовать в центральном процессоре систему ограниченного числа коротких и быстро выполняющихся команд. Таким образом, в RISC-микроконтроллерах команды, как правило, состоят не более чем из 1–2 байтов. Длинную специальную команду приходится составлять их из нескольких коротких. Чтобы достичь равной производительности с CISC-компьютерами, большинство RISC-компьютеров располагают бóльшим числом регистров. Регистр — это встроенная в центральный процессор сверхскоростная память. Еще один отличительный признак RISC-систем — четкое физическое и логическое разделение между областями памяти программ и данных.

1.3.3. Выводы

CISC-компьютер обладает множеством специальных команд, которые занимают большой объем памяти и требуют, как правило, длительного времени выполнения. Команды RISC-компьютера потребляют меньше памяти и выполняются значительно быстрее. Тем не менее, недостаток RISC-технологии состоит в том, что здесь специальные команды приходится заменять цепочками из нескольких основных команд. Таким образом, и CISC- и RISC-технология имею свои преимущества и недостатки. Следует отметить, что не существует ни полностью RISC-, ни полностью CISC-систем.

На рис. 1.5 приведена подробная блок-схема микроконтроллера для изучения его внутреннего устройства.

Глава 2



Программирование микроконтроллеров

Степень интеграции микропроцессоров и микроконтроллеров все возрастает и они все шире проникают в прикладные области метрологии, техники управления и автоматического регулирования. Даже в обычной жизни микроконтроллеры становятся все популярнее. Так происходит, с одной стороны, оттого, что сегодня сложные аналоговые схемы заменяются более простыми цифровыми микроконтроллерами. Но решающее преимущество микроконтроллеров — непревзойденное соотношение цена/производительность.

2.1. Что такое программа?

Программа — это описание процесса обработки информации. При выполнении программы рассчитывается совокупность выходных значений исходя из совокупности переменных или постоянных входных значений. Цель выполнения программы — сбор данных либо получение отклика на входные значения. Наряду с собственно вычислениями программа может содержать команды для доступа к аппаратным средствам компьютера и для управления ходом выполнения алгоритма. Программа состоит из нескольких строк так называемого исходного текста. При этом каждая строка содержит один или несколько арифметических или управляющих операторов. Не только сами команды, но и последовательность их выполнения существенно влияет на результат обработки информации. Выполнение соответствующих операций происходит последовательно (по очереди). Упорядоченную определенным образом последовательность инструкций программы называют также алгоритмом.

2.2. Программирование на C

Язык программирования C (ANSI C) прост для изучения. C — это язык программирования высокого уровня, который создал Деннис Ричи (Dennis Ritchie) в начале 1970-х годов в Bell Laboratories для операционной системы UNIX. С тех пор этот язык очень широко распространен. Области применения языка C весьма различны. Он используется, например, в системном и прикладном программировании.

Основные модули всех систем UNIX и ядро многих операционных систем запрограммированы на языке C.

Многочисленные другие языки, например C ++, Objective-C, C #, Java, PHP и Perl ориентируются на синтаксис и свойства языка C. Изучение этого языка программирования очень выгодно, т. к. в дальнейшем легче освоить многие системы микроконтроллеров. Почти для всех микроконтроллеров существует бесплатный компилятор C, предлагаемый производителем микроконтроллера. Компилятор C от Arduino несколько проще, чем профессиональные C-компиляторы, но весьма эффективен. C компилятором Arduino не нужно заботиться о программировании сложных аппаратных средств, поскольку в среде разработки есть соответствующие встроенные команды.



Петин В.А.

Проекты с использованием контроллера Arduino

— СПб.: БХВ-Петербург, 2014. — 400 с.: ил.
— (Электроника)

ISBN 978-5-9775-3337-9

Рассмотрены основные платы Arduino и платы расширения (шилды), добавляющие функциональность основной плате. Подробно описан язык и среда программирования Arduino IDE. Тщательно разобраны проекты с использованием контроллеров семейства Arduino. Это проекты в области робототехники, создания погодных метеостанций, "умного дома", вендинга, телевидения, Интернета, беспроводной связи (bluetooth, радиоуправление). Для всех проектов представлены схемы и исходный код. Также приведен исходный код для устройств Android, используемых в проектах для связи с контроллерами Arduino. На сайте издательства размещен архив с исходными кодами программ проектов из книги, исходными кодами библиотек, описаниями и спецификациями электронных компонентов, схемами из книги в формате spl7

СОДЕРЖАНИЕ

Предисловие
Для кого и о чем эта книга?
Структура книги
Благодарности

ЧАСТЬ I. ARDUINO — ОБЩИЙ ОБЗОР

Глава 1. Введение в Arduino
1.1. Arduino — что это?
1.2. В чем преимущество Arduino?
1.3. История создания Arduino

Глава 2. Обзор контроллеров семейства Arduino
2.1. Arduino Pro Mini
2.2. Arduino Duemilanove
2.3. Arduino Nano
2.4. Arduino LilyPad
2.5. Arduino Uno
2.6. Arduino Mega2560
2.7. Arduino Leonardo
2.8. Arduino Due

Глава 3. Платы расширения Arduino

ЧАСТЬ II. СРЕДА РАЗРАБОТКИ И ЯЗЫК ПРОГРАММИРОВАНИЯ КОНТРОЛЛЕРОВ ARDUINO

Глава 4. Среда программирования Arduino IDE
4.1. Установка Arduino IDE в Windows
4.2. Установка Arduino IDE в Linux
4.3. Настройка средв Arduino IDE

Глава 5. Программирование в Arduino
5.1. Базовые знания
5.1.1. Цифровые выводы
5.1.2. Аналоговые входы
5.1.3. Широтно-импульсная модуляция
5.1.4. Память в Arduino
5.2. Структура программы
5.2.1. Функции setup() и loop()
5.3. Синтаксис и операторы
5.3.1. Управляющие операторы
5.3.1.1. Оператор if (условие) и операторы сравнения ==, !=, < , >
5.3.1.2. Оператор if..else
5.3.1.3. Оператор for
5.3.1.4. Оператор switch
5.3.1.5. Оператор while
5.3.1.6. Оператор do ... while
5.3.1.7. Оператор break
5.3.1.8. Оператор continue
5.3.1.9. Оператор return
5.3.2. Синтаксис
5.3.2.1. ; (точка с запятой) ; (semicolon)
5.3.2.2. {} (фигурные скобки) {} (curly braces)
5.3.2.3. Комментарии // (single line comment), /* */ (multi-line comment)
5.3.3. Арифметические операторы
5.3.3.1. = (assignment) = оператор присваивания
5.3.3.2. + (сложение), - (вычитание), * (умножение) , / (деление)
5.3.3.3. % (modulo)
5.3.4. Операторы сравнения
5.3.5. Логические операторы
5.3.5.1. && (логическое И)
5.3.5.2. || (логическое ИЛИ)
5.3.5.3. ! (логическое отрицание)
5.3.6. Унарные операторы
5.3.6.1. ++ (увеличение значения) / -- (уменьшение значения)
5.3.6.2. += , -= , *= , /=
5.4. Данные
5.4.1. Типы данных
5.4.1.1. boolean
5.4.1.2. char
5.4.1.3. byte
5.4.1.4. int
5.4.1.5. unsigned int
5.4.1.6. long
5.4.1.7. unsigned long
5.4.1.8. float
5.4.1.9. double
5.4.1.10. string — текстовые строки
5.4.1.11. Массивы
5.4.1.12. void
5.4.2. Константы
5.4.3. Переменные
5.4.3.1. Объявление переменных
5.4.3.2. Границы переменных
5.4.4. Преобразование типов данных
5.4.4.1. char()
5.4.4.2. byte()
5.4.4.3. int()
5.4.4.4. long()
5.4.4.5. float()
5.5. Функции
5.5.1. Цифровой ввод/вывод
5.5.1.1. Функция pinMode
5.5.1.2. Функция digitalWrite()
5.5.1.3. Функция digitalRead()
5.5.2. Аналоговый ввод/вывод
5.5.2.1. Функция analogRead()
5.5.2.2. Функция analogReference()
5.5.2.3. Функция analogWrite()
5.5.3. Дополнительные фунции ввода/вывода
5.5.3.1. Функция tone()
5.5.3.2. Функция noTone()
5.5.3.3. Функция shiftOut()
5.5.3.4. Функция pulseIn()

5.5.4. Работа со временем
5.5.4.1. Функция millis()
5.5.4.2. Функция micros()
5.5.4.3. Функция delay()
5.5.4.4. Функция delayMicroseconds()
5.5.5. Математические функции
5.5.5.1. Функция min(x,yx)
5.5.5.2. Функция max(x, y)
5.5.5.3. Функция abs()
5.5.5.4. Функция constrain(x, a, b)
5.5.5.5. Функция map(value, fromLow, fromHigh, toLow, toHigh)
5.5.5.6. Функция pow(base, exponent)
5.5.5.7. Функция sq(x)
5.5.5.8. Функция sqrt(x)
5.5.6. Тригонометрические функции
5.5.6.1. Функция sin(rad)
5.5.6.2. Функция cos(rad)
5.5.6.3. Функция tan(rad)
5.5.7. Генераторы случайных значений
5.5.7.1. Функция randomSeed(seed)
5.5.7.2. Функция random()
5.5.8. Операции с битами и байтами
5.5.8.1. Функция lowByte()
5.5.8.2. Функция highByte()
5.5.8.3. Функция bitRead()
5.5.8.4. Функция bitWrite()
5.5.8.5. Функция bitSet()
5.5.8.6. Функция bitClear()
5.5.8.7. Функция bit()
5.5.9. Внешние прерывания
5.5.9.1. Функция attachInterrupt
5.5.9.2. Функция detachInterrupt

ЧАСТЬ III. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ARDUINO

Глава 6. Arduino и набор функций Serial
6.1. Функции библиотеки Serial
6.1.1. Функция Serial.begin()
6.1.2. Функция Serial.end()
6.1.3. Функция Serial.available()
6.1.4. Функция Serial.read()
6.1.5. Функция Serial.flush()
6.1.6. Функция Serial.print()
6.1.7. Функция Serial.println()
6.1.8. Функция Serial.write()
6.1.9. Функция Serial.peek()
6.2. Библиотека SoftwareSerial
6.2.1. Функция SoftwareSerial()
6.2.2. Функция SoftwareSerial.listen()
6.2.3. Функция SoftwareSerial.isListening()
6.2.4. Функция SoftwareSerial.overflow()

Глава 7. Arduino и знакосинтезирующие жидкокристаллические индикаторы
7.1. Принцип работы модулей ЖКИ WINSTAR WH1604
7.2. Библиотека LiquidCrystal
7.2.1. Функция LiquidCrystal()
7.2.2. Функция begin()
7.2.3. Функция clear()
7.2.4. Функция home()
7.2.5. Функция setCursor()
7.2.6. Функция write()
7.2.7. Функция print()
7.2.8. Функция cursor()
7.2.9. Функция noCursor()
7.2.10. Функция blink()
7.2.11. Функция noBlink()
7.2.12. Функция display()
7.2.13. Функция noDisplay()
7.2.14. Функция scrollDisplayLeft()
7.2.15. Функция scrollDisplayRight()
7.2.16. Функция autoscroll()
7.2.17. Функция noAutoscroll()
7.2.18. Функция leftToRight()
7.2.19. Функция rightToLeft()
7.2.20. Функция createChar()
7.3. Библиотека LiquidCrystalRus

Глава 8. Библиотека EEPROM
8.1. Функции библиотек EEPROM
8.1.1. Функция чтения EEPROM.read
8.1.2. Функция записи EEPROM.write
8.2. Примеры использования памяти EEPROM
8.2.1. Воспроизведение звука
8.2.2. Звонок с мелодиями

Глава 9. Подключение клавиатуры и мыши
9.1. Обмен данными по протоколу PS/2
9.2. Библиотека ps2dev
9.3. Подключение клавиатуры
9.4. Редактор текста на дисплее WH1604
9.5. Подключение мыши
9.6. Опрос состояния мыши

Глава 10. Arduino и сенсорная панель
10.1. Как работает резистивный экран?
10.2. Программа чтения координат сенсорного экрана
10.3. Библиотека Arduino TouchScreen

Глава 11. Arduino и 1-Wire
11.1. Что такое 1-Wire?
11.2. Применение 1-Wire
11.3. Протокол 1-Wire
11.3.1. Описание интерфейса 1-Wire
11.3.2. Обмен информацией по 1-Wire
11.3.3. Протокол обмена информацией 1-Wire
11.4. Библиотека OneWire
11.4.1. Функция begin()
11.4.2. Функция requestFrom()
11.4.3. Функция beginTransmission()
11.4.4. Функция endTransmission()
11.4.5. Функция write()
11.4.6. Функция available()
11.4.7. Функция read()
11.4.8. Функция onReceive()
11.4.9. Функция onRequest()
11.5. Устройство iButton и программирование электронного замка
11.5.1. Поиск устройств 1-Wire и получение уникального кода
11.5.2. Режимы работы
11.5.3. Режим пользователя
11.5.4. Первоначальный ввод ключа в пустую базу
11.5.5. Просмотр, запись и удаление кодов ключей в режиме администратора
11.5.6. Блок-реле. Открывание замка
11.5.7. Проигрывание мелодий

Глава 12. Arduino и цифровой датчик температуры DS18B20
12.1. Описание датчика DS18B20
12.2. Использование библиотеки OneWire
12.3. Библиотека DallasTemperature

Глава 13. Arduino и датчики температуры и влажности DHT
13.1. Характеристики датчиков DH11, DH22
13.2. Подключение к Arduino
13.3. Библиотека DHT

Глава 14. Сетевой обмен с помощью Arduino
14.1. Устройство Arduino Ethernet shield
14.2. Библиотека Ethernet library
14.2.1. Класс Ethernet (Ethernet class)
Функция Ethernet.begin()
Функция Ethernet.localIP()
14.2.2. Класс IPAddress (IPAddress class)
Функция IPAddress()
14.2.3. Класс Server (Server class)
Функция ethernetServer()
Функция begin()
Функция available()
Функция write()
Функция print()
Функция println()
14.2.4. Класс Client (Client class)
Функция client()
Функция EthernetClient()
Функция connected()
Функция connect()
Функция write()
Функция print()
Функция println()
Функция available()
Функция read()
Функция flush()
Функция stop()
14.2.5. Класс EthernetUDP (EthernetUDP class)
Функция begin()
Функция read()
Функция write()
Функция beginPacket()
Функция endPacket()
Функция parsePacket()
Функция available()

Функция remoteIP()
Функция remotePort()
14.3. Домашняя метеостанция с доступом через Интернет
14.3.1. Устройство, настройка и отладка метеостанции
14.3.2. Создание виджета для планшетов с ОС Android
14.3.3. Размещение данных метеостанции на сайте

Глава 15. Arduino и карта памяти SD
15.1. Arduino-библиотека SD
15.1.1. Класс SD
Функция begin()
Функция exists()
Функция mkdir()
Функция rmdir()
Функция open()
Функция remove()
15.1.2. Класс File
Функция available()
Функция close()
Функция flush()
Функция peek()
Функция position()
Функция print()
Функция println()
Функция seek()
Функция size()
Функция read()
Функция write()
Функция isDirectory()
Функция openNextFile()
Функция rewindDirectory()
15.2. Запись показаний датчиков на SD-карту

Глава 16. Arduino и светодиодные матрицы
16.1. Светодиоды и светодиодные матрицы
16.2. Светодиодная матрица FYM-23881BUG-11
16.3. SPI-расширитель выходов 74HC595
16.4. Игра "Тетрис" на светодиодных матрицах FYM-23881BUG-11
16.4.1. Управление изображением на светодиодных матрицах
16.4.2. Фигуры игры "Тетрис"
16.4.3. Управление фигурами игры "Тетрис"
16.4.4. Проверка столкновения фигур
16.5. Светодиодная матрица RGB
16.6. RGB-ночник, управляемый с помощью движения рук

Глава 17. Работа Arduino с купюроприемником
17.1. Купюроприемник ICT серий A7 и V7
17.2. Подключение купюроприемника ICT V7 к Arduino
17.3. Скетч для получения номинала принимаемой купюры

Глава 18. Arduino и радиочастотная идентификация (RFID)
18.1. Радиочастотная идентификация
18.2. Датчик считывания RFID-карт
18.3. Скетч для считывания RFID-карт

Глава 19. Arduino и датчики расстояния
19.1. Ультразвуковые дальномеры HC-SR04
19.2. Принцип работы ультразвукового дальномера HC-SR04
19.3. Библиотека Ultrasonic
19.4. Инфракрасные датчики расстояния Sharp
19.5. Подключение датчиков Sharp к Arduino

Глава 20. Arduino и передача данных в инфракрасном диапазоне
20.1. Обмен данными в инфракрасном диапазоне
20.2. Протоколы для ИК-пультов
20.2.1. Протокол RC5
20.2.2. Протокол NEC
20.2.3. Протокол JVC
20.2.4. Протокол Sony
20.3. Подключение ИК-приемника
20.4. Библиотека IRremote
20.5. Скетч для получения кодов ИК-пульта

Глава 21. Создаем робота
21.1. Ходовая часть
21.2. Драйвер двигателей L293D
21.3. Массив возможных состояний моторов
21.4. Разработка скетча движений робота
21.5. Движение робота по линии в автономном режиме

Глава 22. Arduino и шаговые двигатели
22.1. Управление шаговым двигателем
22.2. Arduino-библиотека Stepper
22.2.1. Функция Stepper()
22.2.2. Функция setSpeed(rpm)
22.2.3. Функция step(steps)
22.3. Пример использования библиотеки Stepper
22.4. Arduino-библиотека AccelStepper

Глава 23. Arduino и сервоприводы
23.1. Сервоприводы
23.2. Arduino-библиотека Servo для управления сервоприводом
23.2.1. Функция attach()
23.2.2. Функция detach()
23.2.3. Функция write(int)
23.2.4. Функция writeMicrosconds(int)
23.2.5. Функция read()
23.2.6. Функция attached()
23.3. Робот-паук на сервоприводах
23.4. Скетч для управления роботом-пауком

Глава 24. Arduino и Bluetooth
24.1. "Голубой зуб"
24.2. Модуль Bluetooth HC-05
24.3. Управление роботом с Android-устройства по Bluetooth

Глава 25. TV-выход на Arduino
25.1. Схема подключения
25.2. Arduino-библиотека TVOut
25.2.1. Функция установки режима begin()
25.2.2. Функции задержки
Функция delay()
Функция delay_frame()
25.2.3. Функции получения параметров
Функция hres()
Функция vres()
Функция char_line()
25.2.4. Основные графические функции
Функция set_pixel()
Функция get_pixel()
Функция fill()
Функция clear_screen()
Функция invert()
Функция shift_direction()
Функция draw_line()
Функция draw_row()
Функция draw_column()
Функция draw_rect()
Функция draw_circle()
Функция bitmap()
25.2.5. Функции вывода текстовой информации
Функция select_font()
Функция print_char()
Функция set_cursor()
Функция print()
Функция println()
25.2.6. Функции вывода аудио
Функция tone()
Функция noTone()
25.3. Создание пользовательских шрифтов
25.4. Создание графических примитивов
25.5. Создание простейшей игровой консоли
25.6. Разработка игры для игровой консоли
25.6.1. Создание переменных игры
25.6.2. Управление положением "игрока" с помощью джойстика
25.6.3. Генерация и перемещение объектов-цифр
25.6.4. Проверка столкновения "игрока" и объектов-цифр
25.6.5. Счетчик баллов "игрока"
25.6.6. Переход на новый уровень
25.6.7. Отображение данных игры на табло
25.6.8. Звуковое сопровождение игры
25.6.9. Основной цикл игры
25.6.10. Добавляем меню для выбора игр

Глава 26. Arduino и радиоуправление
26.1. Принципы формирования радиосигнала
26.2. Установка связи приемника с передатчиком
26.3. Разработка скетча приема команд для Arduino

Глава 27. Arduino и беспроводной радиомодуль NRF24L01
27.1. Радиомодуль NRF24L01
27.2. Библиотека для работы с модулем NRF24L01
27.2.1. Параметры библиотеки Mirf
27.2.2. Функции библиотеки Mirf
Функция init()
Функция setRADDR()
Функция setTADDR()
Функция config()
Функция dataReady()
Функция getData()

Функция send()
Функция isSending()
27.3. Пример соединения двух плат Arduino с помощью модуля NRF24L01

Глава 28. Работа Arduino с USB-устройствами
28.1. Интерфейс USB
28.2. USB Host Shield
28.3. HID-устройства USB
28.4. Подключение HID-мыши USB
28.5. Использование HID-устройства (руль Defender) для управления роботом
28.6. Подключение к Arduino Android-устройства через USB Host Shield

Глава 29. Arduino и ROS
29.1. Что такое ROS?
29.2. Дистрибутивы ROS
29.3. Установка ROS
29.4. Узлы и темы в ROS
29.5. Пакет rosserial
29.6. Подготовка сообщения (publisher) на Arduino
29.7. Создание подписки (subscriber) на Arduino
29.8. Связь через ROS двух плат Arduino

Глава 30. Arduino и "умный дом" X10
30.1. Система домашней автоматизации X10
30.2. Двусторонний интерфейс TW523
30.3. Arduino-библиотека X10
30.3.1. Функция begin()
30.3.2. Функция write()
30.4. Блок на Arduino для голосового управления приборами X10

Приложение 1. Список использованных источников
Приложение 2. Описание электронного архива

издательство
БХВ-Петербург
www.bhv.ru

Отдел оптовых поставок
E-mail: opt@bhv.spb.su



Петин В.

Проекты с использованием контроллера Arduino, 2-е изд.

464 стр., ISBN 978-5-9775-3550-2

Arduino — проекты любой сложности легко и быстро!

- **Arduino и роботы**
- **Arduino и радиоуправление**
- **Arduino и метеостанции**
- **Arduino и вендинг**
- **Arduino и телевизоры**
- **Голосовое управление умным домом**
- **Управление iRobot Create на Arduino**
- **Arduino Leonardo**

Платформа Arduino совершила маленькую «революцию» в деле демократизации процесса разработки микроэлектронных устройств. «Пятьдесят лет назад для разработки программного обеспечения вам требовалась команда людей в белых халатах, которые знали все об электронных лампах. Но теперь даже моя мама может программировать. Мы позволили множеству людей создавать электронные устройства самостоятельно», — утверждает Массимо Банци, один из создателей платы Arduino. Легкость программирования, доступная новичкам, большое количество плат расширения, программных наработок в виде библиотек позволяют разработчику сосредоточиться не на написании громоздкого кода, а на творчестве и фантазии. Данная книга — подтверждение этого утверждения. Вы увидите, как достаточно просто на Arduino создаются достаточно сложные проекты в различных областях. Робототехника, домашние погодные станции, системы «умного дома», вендинговые системы, радиоуправление, телевидение, Интернет — вот неполный перечень рассмотренных в книге проектов. Вы поймете, что в ваших силах создать подобное и еще больше…

На сайте издательства размещен архив с исходными кодами программ и библиотек, описаниями и спецификациями электронных компонентов и др. Во втором издании добавлены проекты голосового управления с помощью Arduino, работа с адресуемыми RGB-лентами, управление iRobot Create на Arduino. Рассмотрены проекты с использованием платы Arduino Leonardo. Приведены пошаговые уроки для начинающих разработчиков.

Петин Виктор Александрович, профессиональный программист. Круг интересов: робототехника, электроника, программирование. Имеет более 50 публикаций в сетевых изданиях. Автор книги «Микрокомпьютеры Raspberry Pi. Практическое руководство» и двух книг в области практического веб-программирования.

УДК 004.4
ББК 32.973.26-018.2
П29

Петин В. А.

П29 Проекты с использованием контроллера Arduino. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2015. — 464 с.: ил. — (Электроника)

ISBN 978-5-9775-3550-2

Рассмотрены основные платы Arduino и платы расширения (шилды), добавляющие функциональность основной плате. Подробно описан язык и среда программирования Arduino IDE. Тщательно разобраны проекты с использованием контроллеров семейства Arduino. Это проекты в области робототехники, создания погодных метеостанций, "умного дома", вендинга, телевидения, Интернета, беспроводной связи (bluetooth, радиоуправление). Для всех проектов представлены схемы и исходный код. Также представлен исходный код для устройств Android, используемых в проектах для связи с контроллерами Arduino. На сайте издательства размещен архив с исходными кодами программ и библиотек, описаниями и спецификациями электронных компонентов и др. Во втором издании добавлены проекты голосового управления с помощью Arduino, работа с адресуемыми RGB-лентами, управление iRobot Create на Arduino. Рассмотрены проекты с использованием платы Arduino Leonardo. Приведены пошаговые уроки для начинающих разработчиков.

Для читателей, интересующихся современной электроникой

УДК 004.4
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Екатерина Капалыгина*
Редактор	*Григорий Добин*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Марины Дамбиевой*

Подписано в печать 09.12.14.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 37,41.
Тираж 1500 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3550-2

© Петин В. А., 2015
© Оформление, издательство "БХВ-Петербург", 2015

Оглавление

Предисловие 15
Для кого и о чем эта книга? 15
Структура книги 15
Благодарности 16

ЧАСТЬ I. ARDUINO — ОБЩИЙ ОБЗОР 17

Глава 1. Введение в Arduino 19
1.1. Arduino — что это? 19
1.2. В чем преимущество Arduino? 20
1.3. История создания Arduino 20

Глава 2. Обзор контроллеров семейства Arduino 22
2.1. Arduino Pro Mini. 23
2.2. Arduino Duemilanove. 23
2.3. Arduino Nano 25
2.4. Arduino LilyPad 26
2.5. Arduino Uno 27
2.6. Arduino Mega2560 28
2.7. Arduino Leonardo. 29
2.8. Arduino Due 30

Глава 3. Платы расширения Arduino. 31

ЧАСТЬ II. СРЕДА РАЗРАБОТКИ И ЯЗЫК ПРОГРАММИРОВАНИЯ КОНТРОЛЛЕРОВ ARDUINO 37

Глава 4. Среда программирования Arduino IDE 39
4.1. Установка Arduino IDE в Windows. 39
4.2. Установка Arduino IDE в Linux 41

Глава 5. Программирование в Arduino 45
5.1. Базовые знания. 45
5.1.1. Цифровые выводы. 45
5.1.2. Аналоговые входы. 46
5.1.3. Широтно-импульсная модуляция 46
5.1.4. Память в Arduino 46
5.2. Структура программы 48
5.2.1. Функции *setup()* и *loop()* 48
5.3. Синтаксис и операторы. 49
5.3.1. Управляющие операторы.....49
5.3.1.1. Оператор *if* (условие) и операторы сравнения ==, *!*=, < , > 49
5.3.1.2. Оператор *if..else* 49
5.3.1.3. Оператор *for*.....50
5.3.1.4. Оператор *switch* 51
5.3.1.5. Оператор *while*.....51
5.3.1.6. Оператор *do ... while*.....52
5.3.1.7. Оператор *break* 52
5.3.1.8. Оператор *continue* 52
5.3.1.9. Оператор *return* 53
5.3.2. Синтаксис. 53
5.3.2.1. *;* (точка с запятой) *;* (semicolon) 53
5.3.2.2. *{}* (фигурные скобки) *{}* (curly braces) 53
5.3.2.3. Комментарии *//* (single line comment), /* */ (multi-line comment). 54
5.3.3. Арифметические операторы.....54
5.3.3.1. = (assignment) = оператор присваивания 54
5.3.3.2. + (сложение), - (вычитание), * (умножение) , / (деление) 55
5.3.3.3. *%* (modulo) 55
5.3.4. Операторы сравнения. 55
5.3.5. Логические операторы. 55
5.3.5.1. *&&* (логическое И) 55
5.3.5.2. || (логическое ИЛИ) 55
5.3.5.3. *!* (логическое отрицание).....56
5.3.6. Унарные операторы 56
5.3.6.1. ++ (увеличение значения) / -- (уменьшение значения) 56
5.3.6.2. += , -= , *= , /=.....56
5.4. Данные. 56
5.4.1. Типы данных. 56
5.4.1.1. *boolean*.....57
5.4.1.2. *char* 57
5.4.1.3. *byte*.....57
5.4.1.4. *int* 57
5.4.1.5. *unsigned int*.....58
5.4.1.6. *long* 58
5.4.1.7. *unsigned long*.....58
5.4.1.8. *float* 59
5.4.1.9. *double*.....59
5.4.1.10. *string* — текстовые строки.....59
5.4.1.11. Массивы.....60
5.4.1.12. *void*. 61

5.4.2. Константы 61
5.4.3. Переменные 62
5.4.3.1. Объявление переменных 62
5.4.3.2. Границы переменных....62
5.4.4. Преобразование типов данных....63
5.4.4.1. *char()* 63
5.4.4.2. *byte()* 63
5.4.4.3. *int()*....63
5.4.4.4. *long()*....63
5.4.4.5. *float()* 64
5.5. Функции. 64
5.5.1. Цифровой ввод/вывод. 64
5.5.1.1. Функция *pinMode* 64
5.5.1.2. Функция *digitalWrite()*....64
5.5.1.3. Функция *digitalRead()* 65
5.5.2. Аналоговый ввод/вывод....66
5.5.2.1. Функция *analogRead()* 66
5.5.2.2. Функция *analogReference()*....67
5.5.2.3. Функция *analogWrite()* 67
5.5.3. Дополнительные фунции ввода/вывода 69
5.5.3.1. Функция *tone()*....69
5.5.3.2. Функция *noTone()*....69
5.5.3.3. Функция *shiftOut()* 69
5.5.3.4. Функция *pulseIn()* 71
5.5.4. Работа со временем 72
5.5.4.1. Функция *millis()* 72
5.5.4.2. Функция *micros()* 72
5.5.4.3. Функция *delay()* 73
5.5.4.4. Функция *delayMicroseconds()* 74
5.5.5. Математические функции....75
5.5.5.1. Функция *min(x,yx)* 75
5.5.5.2. Функция *max(x, y)*....75
5.5.5.3. Функция *abs()* 75
5.5.5.4. Функция *constrain(x, a, b)* 76
5.5.5.5. Функция *map(value, fromLow, fromHigh, toLow, toHigh)*....76
5.5.5.6. Функция *pow(base, exponent)* 77
5.5.5.7. Функция *sq(x)* 77
5.5.5.8. Функция *sqrt(x)*....77
5.5.6. Тригонометрические функции 78
5.5.6.1. Функция *sin(rad)*....78
5.5.6.2. Функция *cos(rad)* 78
5.5.6.3. Функция *tan(rad)* 78
5.5.7. Генераторы случайных значений....78
5.5.7.1. Функция *randomSeed(seed)*....78
5.5.7.2. Функция *random()* 79
5.5.8. Операции с битами и байтами....79
5.5.8.1. Функция *lowByte()* 80
5.5.8.2. Функция *highByte()*....80

5.5.8.3. Функция *bitRead()* 80
5.5.8.4. Функция *bitWrite()* 80
5.5.8.5. Функция *bitSet()* 81
5.5.8.6. Функция *bitClear()* 81
5.5.8.7. Функция *bit()* 81
5.5.9. Внешние прерывания 81
5.5.9.1. Функция *attachInterrupt* 82
5.5.9.2. Функция *detachInterrupt* 82
5.6. Управление портами через регистры ATmega 83

ЧАСТЬ III. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ARDUINO 87

Глава 6. Arduino и набор функций *Serial*. 89
6.1. Функции библиотеки *Serial* 90
6.1.1. Функция *Serial.begin()* 90
6.1.2. Функция *Serial.end()* 90
6.1.3. Функция *Serial.available()* 90
6.1.4. Функция *Serial.read()* 91
6.1.5. Функция *Serial.flush()* 92
6.1.6. Функция *Serial.print()* 92
6.1.7. Функция *Serial.println()* 93
6.1.8. Функция *Serial.write()* 93
6.1.9. Функция *Serial.peek()* 93
6.2. Библиотека *SoftwareSerial* 94
6.2.1. Функция *SoftwareSerial()* 94
6.2.2. Функция *SoftwareSerial.listen()* 94
6.2.3. Функция *SoftwareSerial.isListening()* 95
6.2.4. Функция *SoftwareSerial.overflow()* 96

Глава 7. Arduino и знакосинтезирующие жидкокристаллические индикаторы 97
7.1. Принцип работы модулей ЖКИ WINSTAR WH1604 98
7.2. Библиотека *LiquidCrystal* 103
7.2.1. Функция *LiquidCrystal()* 104
7.2.2. Функция *begin()* 105
7.2.3. Функция *clear()* 105
7.2.4. Функция *home()* 105
7.2.5. Функция *setCursor()* 105
7.2.6. Функция *write()* 106
7.2.7. Функция *print()* 106
7.2.8. Функция *cursor()* 107
7.2.9. Функция *noCursor()* 107
7.2.10. Функция *blink()* 107
7.2.11. Функция *noBlink()* 107
7.2.12. Функция *display()* 107
7.2.13. Функция *noDisplay()* 108
7.2.14. Функция *scrollDisplayLeft()* 108
7.2.15. Функция *scrollDisplayRight()* 108

7.2.16. Функция *autoscroll()* 108
7.2.17. Функция *noAutoscroll()* 108
7.2.18. Функция *leftToRight()* 109
7.2.19. Функция *rightToLeft()* 109
7.2.20. Функция *createChar()* 109
7.3. Библиотека *LiquidCrystalRus* 111

Глава 8. Библиотека *EEPROM*. 116
8.1. Функции библиотек *EEPROM* 116
8.1.1. Функция чтения *EEPROM.read* 116
8.1.2. Функция записи *EEPROM.write* 117
8.2. Примеры использования памяти EEPROM 118
8.2.1. Воспроизведение звука 118
8.2.2. Звонок с мелодиями 121

Глава 9. Использование Arduino Leonardo в качестве USB-устройства. 126
9.1. Имитация клавиатуры на Arduino Leonardo 127
9.2. Имитация компьютерной мыши на Arduino Leonardo 129

Глава 10. Arduino и 1-Wire 133
10.1. Что такое 1-Wire? 133
10.2. Применение 1-Wire. 136
10.3. Протокол 1-Wire 137
10.3.1. Описание интерфейса 1-Wire 137
10.3.2. Обмен информацией по 1-Wire 138
10.3.3. Протокол обмена информацией 1-Wire 141
10.4. Библиотека *OneWire* 143
10.4.1. Функция *begin()* 144
10.4.2. Функция *requestFrom()* 144
10.4.3. Функция *beginTransmission()* 144
10.4.4. Функция *endTransmission()* 145
10.4.5. Функция *write()* 145
10.4.6. Функция *available()* 145
10.4.7. Функция *read()* 146
10.4.8. Функция *onReceive()* 146
10.4.9. Функция *onRequest()* 146
10.5. Устройство iButton и программирование электронного замка 146
10.5.1. Поиск устройств 1-Wire и получение уникального кода 148
10.5.2. Режимы работы. 150
10.5.3. Режим пользователя 151
10.5.4. Первоначальный ввод ключа в пустую базу 151
10.5.5. Просмотр, запись и удаление кодов ключей в режиме администратора 153
10.5.6. Блок-реле. Открывание замка. 156
10.5.7. Проигрывание мелодий 157

Глава 11. Arduino и цифровой датчик температуры DS18B20 159
11.1. Описание датчика DS18B20 159
11.2. Использование библиотеки *OneWire* 162
11.3. Библиотека *DallasTemperature* 163

Глава 12. Arduino и датчики температуры и влажности DHT 165
12.1. Характеристики датчиков DH11, DH22.....165
12.2. Подключение к Arduino 166
12.3. Библиотека *DHT*.....167

Глава 13. Сетевой обмен с помощью Arduino 169
13.1. Устройство Arduino Ethernet shield 169
13.2. Библиотека *Ethernet library*.....171
13.2.1. Класс Ethernet (*Ethernet class*).....171
Функция *Ethernet.begin()*.....172
Функция *Ethernet.localIP()*.....172
13.2.2. Класс IPAddress (*IPAddress class*).....172
Функция *IPAddress()*.....172
13.2.3. Класс Server (*Server class*).....173
Функция *ethernetServer()*.....173
Функция *begin()*.....173
Функция *available()*.....173
Функция *write()*.....174
Функция *print()*.....175
Функция *println()*.....175
13.2.4. Класс Client (*Client class*).....175
Функция *client()*.....176
Функция *EthernetClient()*.....176
Функция *connected()*.....176
Функция *connect()*.....177
Функция *write()*.....177
Функция *print()*.....177
Функция *println()*.....177
Функция *available()*.....178
Функция *read()*.....178
Функция *flush()*.....178
Функция *stop()*.....179
13.2.5. Класс EthernetUDP (*EthernetUDP class*).....179
Функция *begin()*.....179
Функция *read()*.....179
Функция *write()*.....180
Функция *beginPacket()*.....180
Функция *endPacket()*.....181
Функция *parsePacket()*.....181
Функция *available()*.....182
Функция *remoteIP()*.....182
Функция *remotePort()*.....182
13.3. Домашняя метеостанция с доступом через Интернет.....182
13.3.1. Устройство, настройка и отладка метеостанции.....182
13.3.2. Создание виджета для планшетов с ОС Android.....187
13.3.3. Размещение данных метеостанции на сайте.....188

Глава 14. Arduino и карта памяти SD 190
14.1. Arduino-библиотека *SD*.....190
14.1.1. Класс *SD*.....190
Функция *begin()*.....191

Функция *exists()* 191
Функция *mkdir()* 191
Функция *rmdir()* 191
Функция *open()* 191
Функция *remove()* 192
14.1.2. Класс *File* 192
Функция *available()* 192
Функция *close()* 193
Функция *flush()* 193
Функция *peek()* 193
Функция *position()* 194
Функция *print()* 194
Функция *println()* 194
Функция *seek()* 195
Функция *size()* 195
Функция *read()* 195
Функция *write()* 195
Функция *isDirectory()* 196
Функция *openNextFile()* 196
Функция *rewindDirectory()* 196
14.2. Запись показаний датчиков на SD-карту 196

Глава 15. Arduino и светодиодные матрицы 202
15.1. Светодиоды и светодиодные матрицы 202
15.2. Светодиодная матрица FYM-23881BUG-11 203
15.3. SPI-расширитель выходов 74HC595 204
15.4. Игра "Тетрис" на светодиодных матрицах FYM-23881BUG-11 206
15.4.1. Управление изображением на светодиодных матрицах 208
15.4.2. Фигуры игры "Тетрис" 209
15.4.3. Управление фигурами игры "Тетрис" 211
15.4.4. Проверка столкновения фигур 213
15.5. Светодиодная матрица RGB 215
15.6. RGB-ночник, управляемый с помощью движения рук 216

Глава 16. Arduino и управляемые светодиодные ленты RGB 221
16.1. RGB-светодиодная лента WS2812 221
16.2. Arduino-библиотека Adafruit_Neopixel 223
16.3. Создание табло результатов ЧМ-2018 по футболу на RGB-ленте WS2812 224

Глава 17. Работа Arduino с вендинговыми аппаратами 230
17.1. Купюроприемник ICT серий A7 и V7 230
17.2. Подключение купюроприемника ICT V7 к Arduino 234
17.3. Скетч для получения номинала принимаемой купюры 236
17.4. Монетоприемник CH-926 237
Настройка монетоприемника 238
Калибровка монетоприемника 239
17.5. Подключение монетоприемника к Arduino 239
17.6. Написание скетча. 240

Глава 18. Arduino и радиочастотная идентификация (RFID). 242
18.1. Радиочастотная идентификация 242
18.2. Датчик считывания RFID-карт. 245
18.3. Скетч для считывания RFID-карт....247

Глава 19. Arduino и датчики расстояния 250
19.1. Ультразвуковые дальномеры HC-SR04 250
19.2. Принцип работы ультразвукового дальномера HC-SR04 251
19.3. Библиотека *Ultrasonic* 251
19.4. Инфракрасные датчики расстояния Sharp 252
19.5. Подключение датчиков Sharp к Arduino 255

Глава 20. Arduino и передача данных в инфракрасном диапазоне. 256
20.1. Обмен данными в инфракрасном диапазоне....256
20.2. Протоколы для ИК-пультов. 258
20.2.1. Протокол RC5. 258
20.2.2. Протокол NEC 259
20.2.3. Протокол JVC 261
20.2.4. Протокол Sony 262
20.3. Подключение ИК-приемника 263
20.4. Библиотека *IRremote*....263
20.5. Скетч для получения кодов ИК-пульта....265

Глава 21. Создаем робота. 271
21.1. Ходовая часть. 271
21.2. Драйвер двигателей L293D. 274
21.3. Массив возможных состояний моторов 277
21.4. Разработка скетча движений робота 279
21.5. Движение робота по линии в автономном режиме....282

Глава 22. Arduino и шаговые двигатели 291
22.1. Управление шаговым двигателем 292
22.2. Arduino-библиотека *Stepper* 293
22.2.1. Функция *Stepper()* 293
22.2.2. Функция *setSpeed(rpm)* 293
22.2.3. Функция *step(steps)* 294
22.3. Пример использования библиотеки *Stepper* 294
22.4. Arduino-библиотека *AccelStepper* 296

Глава 23. Arduino и сервоприводы. 297
23.1. Сервоприводы. 297
23.2. Arduino-библиотека *Servo* для управления сервоприводом 300
23.2.1. Функция *attach()* 301
23.2.2. Функция *detach()* 301
23.2.3. Функция *write(int)* 302
23.2.4. Функция *writeMicrosconds(int)* 302
23.2.5. Функция *read()* 302
23.2.6. Функция *attached()* 302
23.3. Робот-паук на сервоприводах. 303
23.4. Скетч для управления роботом-пауком 307

Глава 24. Arduino и Bluetooth 312
24.1. "Голубой зуб". 312
24.2. Модуль Bluetooth HC-05 312
24.3. Управление роботом с Android-устройства по Bluetooth 318

Глава 25. TV-выход на Arduino. 327
25.1. Схема подключения. 327
25.2. Arduino-библиотека *TVOut* 327
25.2.1. Функция установки режима *begin()* 328
25.2.2. Функции задержки. 328
Функция *delay()* 328
Функция *delay_frame()* 328
25.2.3. Функции получения параметров 329
Функция *hres()* 329
Функция *vres()* 329
Функция *char_line()* 329
25.2.4. Основные графические функции. 329
Функция *set_pixel()* 329
Функция *get_pixel()* 330
Функция *fill()* 330
Функция *clear_screen()* 330
Функция *invert()* 331
Функция *shift_direction()* 331
Функция *draw_line()* 331
Функция *draw_row()* 331
Функция *draw_column()* 332
Функция *draw_rect()* 332
Функция *draw_circle()* 333
Функция *bitmap()* 333
25.2.5. Функции вывода текстовой информации 333
Функция *select_font()* 334
Функция *print_char()* 334
Функция *set_cursor()* 334
Функция *print()* 334
Функция *println()* 335
25.2.6. Функции вывода аудио 335
Функция *tone()* 335
Функция *noTone()* 335
25.3. Создание пользовательских шрифтов 336
25.4. Создание графических примитивов 337
25.5. Создание простейшей игровой консоли 340
25.6. Разработка игры для игровой консоли. 342
25.6.1. Создание переменных игры 342
25.6.2. Управление положением "игрока" с помощью джойстика. 343
25.6.3. Генерация и перемещение объектов-цифр 345
25.6.4. Проверка столкновения "игрока" и объектов-цифр 346
25.6.5. Счетчик баллов "игрока" 347
25.6.6. Переход на новый уровень 348

25.6.7. Отображение данных игры на табло 348
25.6.8. Звуковое сопровождение игры 349
25.6.9. Основной цикл игры 349
25.6.10. Добавляем меню для выбора игр 350

Глава 26. Arduino и радиоуправление 352
26.1. Принципы формирования радиосигнала 353
26.2. Установка связи приемника с передатчиком 355
26.3. Разработка скетча приема команд для Arduino 356

Глава 27. Arduino и беспроводной радиомодуль NRF24L01. 359
27.1. Радиомодуль NRF24L01. 359
27.2. Библиотека для работы с модулем NRF24L01 360
27.2.1. Параметры библиотеки *Mirf* 361
27.2.2. Функции библиотеки *Mirf* 361
Функция *init()* 361
Функция *setRADDR()* 361
Функция *setTADDR()* 361
Функция *config()* 361
Функция *dataReady()* 362
Функция *getData()* 362
Функция *send()* 362
Функция *isSending()* 362
27.3. Пример соединения двух плат Arduino с помощью модуля NRF24L01 363

Глава 28. Работа Arduino с USB-устройствами. 366
28.1. Интерфейс USB. 366
28.2. USB Host Shield. 367
28.3. HID-устройства USB 368
28.4. Подключение HID-мыши USB 371
28.5. Использование HID-устройства (руль Defender) для управления роботом 371
28.6. Управление iRobot Create с помощью беспроводного геймпада через Arduino 381

Глава 29. Arduino и ROS. 388
29.1. Что такое ROS? 388
29.2. Дистрибутивы ROS. 389
29.3. Установка ROS. 389
29.4. Узлы и темы в ROS. 390
29.5. Пакет *rosserial* 391
29.6. Подготовка сообщения (publisher) на Arduino 392
29.7. Создание подписки (subscriber) на Arduino 395
29.8. Связь через ROS двух плат Arduino 397

Глава 30. Голосовое управление 400
30.1. Управление электроприборами с помощью радиорозеток UNIEL 400
30.2. Радиомодули FS1000A 402
30.3. Модуль распознавания голоса Voice Recognition Module V2 405
30.3.1. Обучение модуля 405
30.4. Система голосового управления. 408

ПРИЛОЖЕНИЯ 415
Приложение 1. Список использованных источников 417
Приложение 2. Начальная школа 418
Урок 1. Управление светодиодом. Включение/выключение. Плавное регулирование яркости. 418
Урок 2. Подключение кнопки 421
Урок 3. Работа с монитором порта 425
Урок 4. Подключение датчика температуры и влажности, работа с монитором порта 428
Урок 5. Подключение модуля часов 431
Урок 6. Работа с жидкокристаллическим экраном 433
Урок 7. Подключение RGB-светодиода, использование широтно-импульсной модуляции. 435
Урок 8. Подключение фоторезистора 438
Урок 9. Подключение фоторезистора и 8-ми светодиодов 440
Проект 1. Бегущий огонек 443
Проект 2. Секундомер 444
Проект 3. Тренер 448
Проект 4. Цветная температура 453
Проект 5. Метеостанция. 455

Приложение 3. Описание электронного архива 462

Предисловие

Для кого и о чем эта книга?

Предлагаемая книга ориентирована на читателей, желающих быстро войти в темы программирования микроконтроллеров и использования микроконтроллеров для связи с внешними системами в проектах автоматизации и робототехники.

Книга содержит описание языка программирования плат Arduino в среде Arduino IDE и предлагает изучение предмета на реальных проектах, имеющих практическое значение. В ней вы найдете множество примеров и проектов использования Arduino, представляющих собой законченные решения, пригодные для использования в ваших проектах.

Для тех, у кого нет опыта разработки проектов Arduino, а также написания программного кода для электронных устройств, к книге предлагается специальный набор деталей и компонентов, с помощью которого вы сможете быстро адаптироваться в мире Arduino и получить удовольствие, видя, как оживают ваши творения[1]! Уроки учебного набора *(приложение 2)* тематически привязаны к соответствующим главам книги, и вы найдете в этих главах необходимые ссылки. Используя детали и компоненты набора, вы сможете реализовать более сложные проекты, описание которых также содержится в *приложении 2*.

Книга сопровождается электронным архивом, содержащим исходный код всех рассмотренных примеров и проектов, а также используемые в проектах необходимые библиотеки *(см. приложение 3)*. Этот электронный архив можно скачать с FTP-сервера издательства "БХВ-Петербург" по ссылке **ftp://ftp.bhv.ru/9785977535502.zip**, а также со страницы книги на сайте **www.bhv.ru**.

Структура книги

Книга состоит из трех частей и включает предисловие, тридцать глав и три приложения.

Часть I содержит описание Arduino, обзор контроллеров семейства Arduino и плат расширения для Arduino.

---

[1] Описание набора см. на **www.bhv.ru/books/192858**.

В *части II* книги рассмотрены среда разработки и язык программирования для контроллеров Arduino.

Часть III посвящена созданию конкретных устройств на основе контроллера Arduino. Проекты содержат электрические схемы и листинги программ. Рассмотрено использование плат расширения (шилдов). В книге широко используются библиотеки Arduino.

В *приложениях* приведены перечень использованной литературы и интернет-ресурсов, уроки и проекты учебного набора и описание электронного архива, сопровождающего книгу.

Благодарности

Хочу поблагодарить родных и близких, которые с пониманием относились к потраченному на книгу (за счет общения с ними) времени.

Большая благодарность издательству "БХВ-Петербург", где поверили в необходимость этой книги, и всем сотрудникам издательства, которые помогали мне в ее создании.

Благодарю также всех читателей, купивших эту книгу, — надеюсь, она поможет вам в разработке собственных проектов на основе Arduino.

ЧАСТЬ I



Arduino — общий обзор

Глава 1. Введение в Arduino

Глава 2. Обзор контроллеров семейства Arduino

Глава 3. Платы расширения Arduino



ГЛАВА 1

Введение в Arduino

1.1. Arduino — что это?

Появление первых микроконтроллеров ознаменовало начало новой эры в развитии микропроцессорной техники. Наличие в одном корпусе большинства системных устройств сделало микроконтроллер подобным обычному компьютеру. В отечественной литературе они даже назывались однокристальными микроЭВМ. Соответственно и желание использовать микроконтроллеры как обычные компьютеры появилось практически с их появлением. Но желание это сдерживалось многими факторами. Например, чтобы собрать устройство на микроконтроллере, необходимо знать основы схемотехники, устройство и работу конкретного процессора, уметь программировать на ассемблере и изготавливать электронную технику. Потребуются также программаторы, отладчики и другие вспомогательные устройства. В итоге без огромного объема знаний и дорогостоящего оборудования не обойтись. Такая ситуация долго не позволяла многим любителям использовать микроконтроллеры в своих проектах. Сейчас, с появлением устройств, дающих возможность работать с микроконтроллерами без наличия серьезной материальной базы и знания многих предметов, все изменилось. Примером такого устройства может служить проект Arduino итальянских разработчиков.

Arduino и его клоны представляют собой наборы, состоящие из готового электронного блока и программного обеспечения. Электронный блок здесь — это печатная плата с установленным микроконтроллером и минимумом элементов, необходимых для его работы. Фактически электронный блок Arduino является аналогом материнской платы современного компьютера. На нем имеются разъемы для подключения внешних устройств, а также разъем для связи с компьютером, по которому и осуществляется программирование микроконтроллера. Особенности используемых микроконтроллеров ATmega фирмы Atmel позволяют производить программирование без применения специальных программаторов. Все, что нужно для создания нового электронного устройства, — это плата Arduino, кабель связи и компьютер. Второй частью проекта Arduino является программное обеспечение для создания управляющих программ. Оно объединило в себе простейшую среду разработки и язык программирования, представляющий собой вариант языка C/C++ для микро-

контроллеров. В него добавлены элементы, позволяющие создавать программы без изучения аппаратной части. Так что для работы с Arduino практически достаточно знания только основ программирования на C/C++. Создано для Arduino и множество библиотек, содержащих код, работающий с различными устройствами.

1.2. В чем преимущество Arduino?

Пользователь современного компьютера не задумывается о функционировании отдельных частей ПК. Он просто запускает нужные программы и работает с ними. Точно так же и Arduino позволяет пользователю сосредоточиться на разработке проектов, а не на изучении устройства и принципов функционирования отдельных элементов. Нет надобности и в создании законченных плат и модулей. Разработчик может использовать готовые платы расширения или просто напрямую подключить к Arduino необходимые элементы. Все остальные усилия будут направлены на разработку и отладку управляющей программы на языке высокого уровня. В итоге доступ к разработке микропроцессорных устройств получили не только профессионалы, но и просто любители что-то сделать своими руками. Наличие готовых модулей и библиотек программ позволяет непрофессионалам в электронике создавать готовые работающие устройства для решения своих задач. А варианты использования Arduino ограничены только возможностями микроконтроллера и имеющегося варианта платы, ну и, конечно, фантазией разработчика.

1.3. История создания Arduino

В 2002 году программист Массимо Банци (Massimo Banzi) был принят на работу в должности доцента в Институт проектирования взаимодействий города Иврея (Interaction Design Institute Ivrea, IDII) для продвижения новых способов разработки интерактивных проектов. Однако крошечный бюджет и ограниченное время доступа к лабораторной базе сводили его усилия практически на нет. В проектах Банци использовал устройство BASIC Stamp, разработанное калифорнийской компанией Parallax. Stamp представлял собой небольшую печатную плату с размещенными на ней источником питания, микроконтроллером, памятью и портами ввода/вывода для соединения с различной аппаратурой. Программирование микроконтроллера осуществлялось на языке BASIC. BASIC Stamp имел две проблемы: недостаток вычислительной мощности и достаточно высокую цену — плата с основными компонентами стоила около 100 долларов. И команда Банци решила самостоятельно создать плату, которая удовлетворяла бы всем их потребностям.

Банци и его сотрудники поставили себе целью создать устройство, представляющее собой простую, открытую и легкодоступную платформу для разработки, с ценой — не более 30 долларов — приемлемой для студенческого кармана. Хотели они и выделить чем-то свое устройство на фоне прочих. Поэтому в противовес другим производителям, экономящим на количестве выводов печатной платы, они решили добавить их как можно больше, а также сделали свою плату синей, в отличие от обычных зеленых плат.

Продукт, который создала команда, состоял из дешевых и доступных компонентов — например, базировался он на микроконтроллере ATmega328. Но главная задача состояла в том, чтобы гарантировать работу устройства по принципу plug-and-play, — чтобы пользователь, достав плату из коробки и подключив к компьютеру, мог немедленно приступить к работе.

Первый прототип платы был сделан в 2005 году, она имела простейший дизайн и еще не называлась Arduino. Чуть позже Массимо Банци придумал назвать ее так — по имени принадлежащего ему бара, расположенного в городе Ивреа. Бренд "Arduino" без какой-либо рекламы и привлечения средств маркетинга быстро приобрел высокую популярность в Интернете. С начала распространения продано более 250 тыс. комплектов Arduino, и это не учитывая множества клонов. В мире насчитывается более двухсот дистрибьюторов продукции Arduino — от крупных фирм, таких как SparkFun Electronics, до мелких компаний, работающих на местный рынок. На сегодня платформа Arduino представлена не одной платой, а целым их семейством. В дополнение к оригинальному проекту, называемому Arduino Uno, новые модели, имеющие на плате более мощные средства, носят название Arduino Mega, компактные модели — Arduino Nano, платы в водонепроницаемом корпусе — LilyPad Arduino, а новая плата с 32-разрядным процессором Cortex-M3 ARM — Arduino Due.

Своим успехом проект Arduino обязан существовавшим до него языку Processing и платформе Wiring. От этих проектов Arduino унаследовал одну сильную черту — удобную для пользователя среду разработки. До появления Arduino программирование микроконтроллеров требовало сложного и рутинного предварительного обучения. А с Arduino даже те, кто не имеет опыта работы с электронными устройствами, теперь могут приобщиться к миру электроники. Начинающим уже не приходится тратить много времени на изучение сопутствующего материала — они могут быстро разработать прототип, который будет полноценно рабочим.

По словам Массимо Банци, пятьдесят лет назад, чтобы написать программное обеспечение требовалась команда людей в белых халатах, которые знали все об электронных лампах. Теперь же, с появлением Arduino, множество людей получили возможность создавать электронные устройства самостоятельно. Как утверждает один из членов команды Банци, инженер по телекоммуникациям Дэвид Куартилльз, философия Arduino как раз и состоит в том, что желающий разобраться в электронике может сразу же приступить к ее изучению, вместо того чтобы сначала учиться алгебре.

Начните экспериментировать уже сегодня!

набор «Метеостанция»

Разработан по материалам книги «Проекты с использованием контроллера Arduino. — 2-е изд.».

Набор включает:

- плату с микроконтроллером Arduino™ Nano, макетную плату и электронные компоненты для выполнения экспериментов, описанных в книге, и создания рабочих проектов;
- провода и кабель USB для подключения к ПК;
- пошаговое руководство;
- детали корпуса для создания метеостанции;
- пластмассовый органайзер для компонентов.



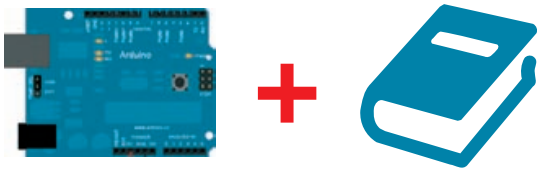
набор «Метеостанция» + книга в одной коробке!

Включает в себя учебный набор «Метеостанция» и книгу «Проекты с использованием контроллера Arduino. — 2-е изд.».



набор «Базовый» плата Arduino UNO + книга в одной коробке!

Содержит плату Arduino Uno, кабель для подключения к порту USB, краткое руководство и книгу «Проекты с использованием контроллера Arduino. — 2-е изд.». Предназначен для получения первичных навыков программирования и работы с Arduino.

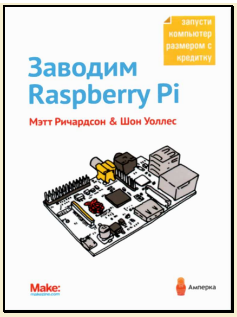


Адреса для заказа наборов

www.bhv.ru/books/192858

www.electronicparts.ru/produkts/kits/meteo.html

дерзай!



Мэтт Ричардсон и Шон Уоллес

amperka.ru

Амперка

Заводим Raspberry Pi

— М.: пер. с англ., Амперка, 2013. — 250 с.: ил.

ISBN 978-5-4465-0280-6
http://amperka.ru/product/getting-started-with-raspberry-pi

Что можно сделать с Raspberry Pi, полноценным компьютером размером с кредитную карточку? Кучу интересных вещей!

Если вы учитесь программировать или собираетесь создать проект, завязанный на электронике, эта книжка расскажет насколько удобной и гибкой может быть эта маленькая платформа. Через серии интересных экспериментов вы узнаете о возможностях Pi, о предустановленных языках программирования, использовании вместе с Arduino, создании онлайн, мультимедиа-проектов и игр. Всё это и многое другое возможно с Raspberry Pi.

Для того, чтобы попрактиковаться вместе с автором по ходу прочтения, с этой книгой мы подготовили специальный набор «Малина Z». Он не только поможет повторить большинство экспериментов из этой книги, но и позволит комфортно двигаться вперёд, к своим удивительным проектам.

Raspberry Pi поражает не только своими возможностями управления электронными устройствами. Это в первую очередь настоящий компьютер. Если вам интересны только огромные программные возможности этой платы, обратите внимание на наш набор «Малина Y», который содержит всё, что нужно для начала работы с Raspberry Pi.

Ну и наконец, набор «Малина» в сочетании с одним из наборов серии «Матрёшка», позволит вашему маленькому гению не только повторить все эксперименты из этой книги, но и начать творить в действительно впечатляющих масштабах.

СОДЕРЖАНИЕ

Глава 1. Берём и запускаем
Знакомство с платой
Правильная периферия
Корпус
Выберите дистрибутив
Запись SD-карты
Загрузка
Настраиваем свою Pi
Выключение
Устранение неисправностей
Дополнительная информация
Глава 2. Linux в Raspberry Pi
Использование командной строки
Другие команды Linux
Директория /etc
Установка даты и времени
Установка нового ПО
Для самостоятельного изучения
Глава 3. Python на Raspberry Pi
Hello Python
Углубляемся в Python
Объекты и модули
Углубляемся в модули
Поиск и устранение ошибок
Для самостоятельного изучения
Глава 4. Мультимедиа и анимация в Python
Hello Pygame
Поверхности в Pygame
Рисуем на поверхностях
Обработка событий и входных данных
Спрайты
Воспроизведение звука
Воспроизведение видео
Самостоятельное изучение
Глава 5. Scratch в Raspberry Pi
Hello Scratch
Сцена
Ещё две вещи, которые нужно знать о спрайтах
Пример помощнее: «Астральные нарушители»
Scratch для взаимодействия с реальным миром
Публикация своих программ
Глава 6. Arduino и Raspberry Pi
Установка Arduino в Raspbian
Взаимодействие по последовательному порту
Самостоятельное изучение
Глава 7. Ввод и вывод на низком уровне
Использование GPIO
Проект: автоматическое освещение через «Cron»
Самостоятельное изучение
Глава 8. Использование GPIO в паре с Python
Установка и работа с GPIO в Python
Мигающий светодиод
Считывание кнопки
Проект: простой проигрыватель
Самостоятельное изучение
Глава 9. Веб-камеры
Проверка веб-камеры
Установка и тестирование SimpleCV
Отображение картинки
Изменяем изображение
Доступ к камере
Распознавание лиц
Проект: фотобудка на Raspberry Pi
Самостоятельное изучение
Глава 10. Python и интернет
Загрузка данных с веб-сервера
Создание веб-сервера на Pi
Взаимодействие сети с реальным миром
Проект: веб-лампа
Самостоятельное изучение
Приложение А. Записываем образ на SD-карту
Создаём загрузочный диск в OS X
Создаём загрузочный диск в Windows *
Создаём загрузочный диск в Linux
Приложение В. Астральные нарушители. Полная версия
Приложение С. Аналоговые сигналы
Конвертируем аналоговый сигнал в цифровой

amperka.ru
Амперка

Глава 6

Arduino и Raspberry Pi

В следующих главах мы поговорим об использовании контактов GPIO на Raspberry Pi для подключения сенсоров, светодиодов и моторов. Однако, если у вас уже есть опыт использования платформы Arduino, то можно использовать его в отношении Raspberry Pi.

Когда Raspberry Pi была анонсирована, многие спрашивали: является ли она убийцей Arduino? За ту же цену вы получаете компьютер гораздо большей мощности. Тогда зачем использовать Arduino? Но, на самом деле, эти платформы прекрасно дополняют друг друга. Вот несколько причин использования Raspberry Pi и Arduino вместе:

- для Arduino существует множество библиотек и примеров;
- у вас есть готовый проект, построенный на Arduino, и вы хотите расширить его возможности, добавив вычислительной мощности. Например, у вас есть MIDI-контроллер, подключённый к синтезатору, а теперь вы хотите синтезировать звук прямо на Raspberry Pi;

уровне в 3,3 вольта, и её контакты не выдержат 5 вольт;

- если вы собираете что-то, чего раньше не собирали, вы можете повредить микроконтроллер. Я видел, как студенты пытались подключить моторы прямо к контактам Arduino (никогда так не делайте). К счастью, на Arduino очень легко поменять микроконтроллер, и стоит это порядка 300 рублей, а вот на Raspberry Pi так сделать не получится;

- когда требуется управлять чем-либо, очень точно контролируя время, например, 3D-принтером. В главе 3 мы говорили, что Raspbian не является системой реального времени, поэтому программы не могут полагаться на железобетонную зависимость времени от количества тактов процессора, как на микроконтроллерах.

Примеры в этой главе предполагают, что у вас есть навык работы с Arduino и Arduino IDE. Если же такого навыка у вас нет, то мы советуем вам прочитать книгу Массимо Банци «Arduino для начинающих волшебников» (http://amperka.ru/rpi/banzi-book). Официальный сайт Arduino (http://arduino.cc/en/Tutorial/HomePage) тоже может помочь в освоении платформы. Также оттуда можно скопировать различные готовые примеры кода.



Много интересных материалов на русском языке про Arduino можно найти на вики «Амперки» (http://wiki.amperka.ru).

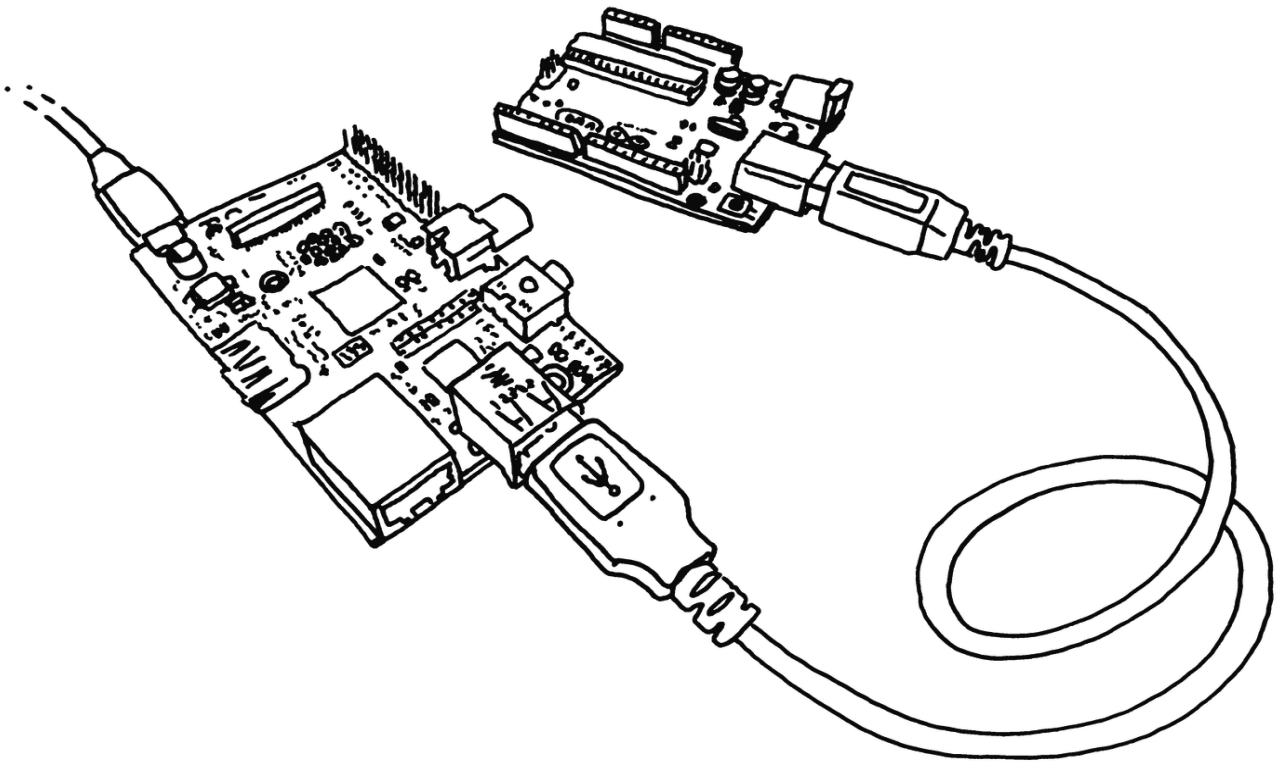


Рис. 6.1. Arduino и Raspberry Pi — лучшие друзья

6.1 Установка Arduino в Raspbian

Для того, чтобы запрограммировать плату Arduino, её нужно подключить к компьютеру при помощи USB-кабеля, а затем загрузить на неё программу при помощи Arduino IDE. Это можно сделать на любом компьютере, включая Raspberry Pi.

Использование Raspberry Pi для программирования Arduino ускоряет процесс отладки кода, но скорость компиляции будет значительно ниже,чем на современных ПК. Однако, всё не так уж и плохо. Самой долгой будет первая компиляция, так как Arduino IDE компилирует только изменённую часть программы.



Программировать Arduino в любимом редакторе и работать с ней из командной строки, минуя медленную среду Arduino IDE, поможет утилита `ino` (http://inotool.org).

Чтобы установить Arduino IDE на Raspberry Pi, выполните следующие команды в терминале:

```
sudo apt-get update ❶
sudo apt-get install arduino ❷
```

❶ Обновляем список пакетов.

❷ Устанавливаем пакет arduino.

Эта команда установит также Java и множество других пакетов, от которых зависит Arduino IDE. После установки Arduino IDE появится в разделе *Электроника* главного меню. Не запускайте её пока что.

Если вы работаете с Raspberry Pi по SSH, можете просто подключить плату через свободный USB-порт. Если же свободных портов не осталось, можно использовать порт на клавиатуре или подключить хаб. USB-порт должен предоставлять достаточное количество энергии для платы, однако хорошей идеей является использование внешнего источника питания для лучшей стабильности.



Важно подключать плату Arduino после загрузки Raspberry Pi, поскольку, если она будет подключена во время загрузки, Raspberry Pi может зависнуть, пытаясь определить, что это за устройство.

При запуске Arduino IDE опрашивает все USB-устройства и составляет список, который вы видите в меню *Сервис → Последовательный порт*. Для доступа к последовательному порту нужно убедиться, что пользователь *pi* имеет права на это. Для этого добавьте пользователя *pi* в группу *tty* и *dialout*. Сделать это нужно до запуска среды.

```
sudo usermod❶  -a -G❷  tty pi
sudo usermod -a -G dialout pi
```

❶ `usermod` — это команда Linux для управления пользователями.

❷ Ключи `-a -G` добавляют пользователя *pi* в указанные группы: *tty*, затем — *dialout*.

Теперь вы можете работать с Arduino. В меню *Сервис → Последовательный порт* выберите порт, к которому подключена плата (скорее всего — */dev/ttyACM0*). Затем в меню *Сервис → Плата* выберите ту плату, которую хотите использовать (например, *Uno*). Зайдите в меню *Файл → Примеры → 01.Basics* и загрузите скетч-пример *Blink*. Нажмите кнопку загрузки на панели инструментов или в меню *Файл* и скетч загрузится на плату. Если всё прошло успешно, то один светодиод на Arduino начнёт мигать.

6.1.1 Определение последовательного порта

Если по каким-то причинам порт */dev/ttyACM0* не работает, придётся провести небольшое расследование. Чтобы понять к какому порту подключена плата, отключите её и выполните следующую команду в терминале:

```
ls /dev/tty*
```

Затем подключите плату и повторите команду. Посмотрите, что изменилось. На моей Raspberry Pi сначала было только устройство `/dev/ttyAMA0` (это встроенный хаб). А после подключения Arduino появилось устройство `/dev/ttyACM0`.

Делаем использование удобнее

Пока вы всё настраивали, вы наверняка заметили, что качество шрифта в редакторе Arduino оставляет желать лучшего. Это можно исправить установкой шрифта *Inconsolata*:

```
sudo apt-get install ttf-inconsolata
```

Затем откройте файл настроек Arduino:

```
nano ~/.arduino/preferences.txt
```

И измените следующие строки:

```
editor.font=Inconsolata,medium,14
editor.antialias=true
```

После перезапуска Arduino IDE, редактор будет использовать новый шрифт.

6.2 Взаимодействие по последовательному порту

Для взаимодействия Raspberry Pi и Arduino через последовательное соединение можно использовать встроенную библиотеку *Serial* на Arduino, и модуль *pySerial* (http://pyserial.sourceforge.net/) на Raspberry Pi. Эта команда установит его:

```
sudo apt-get install python-serial python3-serial
```

Откройте Arduino IDE и загрузите этот код на плату:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  for (byte n = 0; n < 255; n++) {
   Serial.write(n);
   delay(50);
  }
}
```

Этот код посылает по последовательному соединению все числа от нуля до 255.



Имейте в виду, что в Arduino комада `Serial.write()` посылает число как таковое. Чтобы отправить строку «123» вместо числа 123, используйте `Serial.print()`.

Теперь нужно узнать, к какому последовательному порту подключена плата Arduino (рис. 6.1.1). В следующем скрипте на Python’е (мы говорили о Python’е в главе 3) установите нужное значение переменной `port`, если плата подключена к порту, отличному от */dev/ttyACM0*. Сохраните его в файл *SerialEcho.py* и запустите командой `python SerialEcho.py`:

```
import serial

port = "/dev/ttyACM0"
serialFromArduino = serial.Serial(port,9600) ❶
serialFromArduino.flushInput() ❷
while True:
```

```
    if (serialFromArduino.inWaiting() > 0):
        input = serialFromArduino.read(1) ❸
        print(ord(input)) ❹
```

❶ Открываем порт, к которому подключена плата Arduino.

❷ Выбрасываем все данные из буфера входящего потока.

❸ Читаем один байт из буфера последовательного соединения.

❹ Полученный байт интерпретируем как обычное число функцией `ord()`.



Вы не сможете загружать скетчи на плату, пока Python использует подключение. Так что сначала нужно завершить программу горячей комбинацией `ctrl-C`. На деле вы сможете загружать скетчи на платы Leonardo и Micro, но при этом прервётся соединение с Python-скриптом, и программу всё равно придётся перезапустить.

Arduino посылает Python-скрипту число, которое тот интерпретирует как строку. Переменная `input` получит значение в соответствии с тем, на какой символ указывает переданное число в таблице ASCII-символов (http://en.wikipedia.org/wiki/ASCII). Для того, чтобы лучше понять, что тут происходит, попробуйте заменить последнюю строку скрипта на следующую:

```
print(str(ord(input)) + " = the ASCII character " + input)
```

Установка последовательного порта через аргумент

Если вы хотите передавать порт в скрипт при запуске в качестве аргумента командной строки, используйте модуль `sys`, чтобы получить значение первого аргумента:

```
import serial, sys

if (len(sys.argv) != 2):
    print("Usage: python ReadSerial.py port")
    sys.exit()
port = sys.argv[1]
```

Теперь вы можете запускать скрипт следующей командой: `python SerialEcho.py /dev/ttyACM0`.

В первом примере мы посылали всего лишь один байт. Этого может быть достаточно, если вам всего-навсего нужно сообщать о каких-то событиях на Arduino. Например, при нажатии левой кнопки посылать 1, а при нажатии правой — 2. Такой подход сработает, если в программе есть не более 255 событий. Однако чаще требуется передавать большие произвольные числа и строки. При чтении аналоговых сенсоров, например, числа будут в диапазоне от нуля до 1023.

Обработка произвольных чисел бит за битом может вызвать множество затруднений во многих языках программирования. Но то, как Python и PySerial управляются со строками, делает задачу тривиальной. В качестве примера давайте заставим нашу плату посылать числа от нуля до 1024:

```
void setup() {
 Serial.begin(9600);
}
```

```
void loop() {
 for (int n = 0; n < 1024; n++)
   Serial.println(n, DEC);
   delay(50);
  }
}
```

Главное отличие от предыдущего примера в том, что здесь мы используем функцию `println()`. В предыдущем примере функция `Serial.write()` посылала Raspberry Pi само число в сыром виде. Здесь же функция `println()` представляет число в виде строки символов и посылает ASCII-коды символов этой строки. Таким образом, вместо числа 254 посылается строка `254\r\n`. Символ `\r` — это возврат каретки в начало строки, а символ `\n` — это переход на новую строку. Этот принцип унаследован от печатных машинок.

На стороне Python'а теперь можно использовать функцию `readline()`, вместо `read()`, которая считывает все символы до (и включая) символы возврата каретки и новой строки. Python очень гибок с точки зрения конвертации между различными типами данных и строк. Оказывается, можно просто воспользоваться фунуцией `int()` для перевода строки в число:

```
import serial

port = "/dev/ttyACM0"
serialFromArduino = serial.Serial(port,9600)
serialFromArduino.flushInput()
while True:
    input = serialFromArduino.readline() ❶
    inputAsInteger = int(input) ❷
    print(inputAsInteger * 10) ❸
```

❶ Считываем всю строку в переменную `input`.

❷ Конвертируем её в число.

❸ Выводим её, но сначала умножаем на 10, чтобы доказать, что это действительно число, а не строка.

Кстати, этот пример можно легко изменить таким образом, что данные будут считываться с аналогового сенсора. Для этого достаточно чуть изменить `loop()`:

```
void setup() {
 Serial.begin(9600);
}

void loop() {
  int n = analogRead(A0);
  Serial.println(n, DEC);
  delay(100);
}
```

Если вы будете выводить из скрипта на Python просто значение переменной `inputAsInteger` без умножения её на 10, вы должны получить некое плавающее значение в районе 200 при условии, что к аналоговому входу 0 ничего не подключено. С помощью провода подключите этот контакт к земле (GND) и вы получите значение 0, подключите к 3,3 вольтам и получите примерно 715, подключите к 5 вольтам — получите 1023.

6.3 Самостоятельное изучение

Всё вполне тривиально, когда у вас имеется нужный софт. Многие ваши проекты будут содержать похожий код, пока вы занимаетесь простой передачей данных. Однако, как и всегда, всё становится сложнее, как только вы пытаетесь сделать что-то круче «hello world». Вам придётся разрабатывать *протоколы* (или находить готовые и применять их), чтобы обе стороны понимали друг друга. Детали разработки протоколов выходят за рамки этой книги, но вы сможете найти много примеров того,

как различные люди решали эти проблемы на сайте Arduino, в разделе Interfacing with Software (http://www.arduino.cc/playground/Main/InterfacingWithSoftware).

Firmata

Firmata (http://arduino.cc/en/Reference/Firmata) — это последовательный протокол общего назначения, созданный Гансом-Кристофером Штейнером (Hans-Christoph Steiner). Он прост и понятен, хотя, возможно, подходит и не для всех случаев, но с него можно начать.

MIDI

Если у вас музыкальный проект, то можно использовать MIDI-команды как протокол. Поскольку MIDI по сути своей является протоколом последовательной передачи данных, он точно подойдёт.

Платы расширения с Arduino для Raspberry Pi

Существует парочка плат расширения (ещё называемых «шилдами»), которые подключаются к Raspberry Pi через контакты GPIO и несут на своём борту микроконтроллер, совместимый с Arduino. A la Mode (http://baldwisdom.com/projects/alamode/) — хорошее решение: на нём также установлена пара дополнительных примочек вроде часов реального времени.



Аналогичными платами также являются Gertboard (http://amperka.ru/rpi/gertboard) и PiFace (http://amperka.ru/rpi/piface)

Взаимодействие через сеть

Вы можете вообще забыть про последовательное соединение и взаимодействовать через сеть. Многие интересные проекты используют протокол WebSocket (http://www.websocket.org/) вместе с JavaScript'овой платформой Node.js (http://nodejs.org/). Проект Noduino (http://semu.github.com/noduino/) — хорошее место для изучения этой технологии.

Использование аппаратного последовательного порта на Raspberry Pi

На Raspberry Pi имеются контакты GPIO, и пара из них может использоваться как аппаратный сериал порт, в обход USB-порта. Но для того, чтобы их использовать, прочитайте сначала главу 8 и обязательно используйте схему, которая преобразует 5 вольт от Arduino в 3,3 вольта для Raspberry Pi.

Для погружения в тему коммуникации между устройствами вы можете прочитать книгу Тома Иго (Tom Igoe) Making Things Talk (http://shop.oreilly.com/product/9780596510510.do).



Амперка

Конспект хакера

amperka.ru

Амперка

— М.: Амперка, 2014. — 84 с.: ил.

Электронная версия: http://amperka.ru/product/hacker-workbook

Хотите попробовать Arduino, но нет времени на чтение книжек? С этим руководством вы сможете в кратчайшие сроки опробовать в действии бóльшую часть функций Arduino.

В брошюре кратко, но в то же время доступно описываются все основные понятия, необходимые для реализации собственных электронных идей. Брошюра идеально подходит в качестве руководства по использованию наборов «Матрёшка». На самом деле конспект создан специально для них и уже включён в комплект «Матрёшек». Однако, если у вас уже есть Arduino и радиодетали, можете приобрести конспект отдельно.

Первая часть брошюры представляет собой вводную в электричество. В ней объясняются такие простые понятия, как резистор, электрическая схема, закон Ома и т.д. Чтобы теория не забылась, а прочно укоренилась в уме начинающего, во второй части брошюры приведены 20 простых заданий.

При написании брошюры приоритетными задачами были краткость и доступность, а не точность и общность. На 84-х страницах описаны, конечно, далеко не все аспекты схемотехники и функции Arduino. Брошюра призвана заинтересовать начинающего, избавить от рутины и в максимально сжатые сроки помочь определиться: нужно ему это или нет.

СОДЕРЖАНИЕ

Электричество
Схемы
Основные законы электричества
Управление электричеством
Сборка схем
Резистор
Делитель напряжения
Диод
Светодиод
Светодиодные сборки
Кнопка
Биполярный транзистор
Полевой транзистор
Широтно-импульсная модуляция
Конденсатор
Пьезодинамик
Мотор
Сервопривод
Микросхема
Сдвиговый регистр
Триггер Шмитта

Начало работы с Arduino
Пример №1. Маячок
Пример №2. Маячок с нарастающей яркостью
Пример №3. Светильник с управляемой яркостью
Пример №4. Терменвокс
Пример №5. Ночной светильник
Пример №6. Пульсар
Пример №7. Бегущий огонёк
Пример №8. Мерзкое пианино
Пример №9. Миксер
Пример №10. Кнопочный переключатель
Пример №11. Светильник с кнопочным управлением
Пример №12. Кнопочные ковбои
Пример №13. Секундомер
Пример №14. Счётчик нажатий
Пример №15. Комнатный термометр
Пример №16. Метеостанция
Пример №17. Пантограф
Пример №18. Тестер батареек
Пример №19. Светильник, управляемый по USB
Пример №20. Перетягивание каната

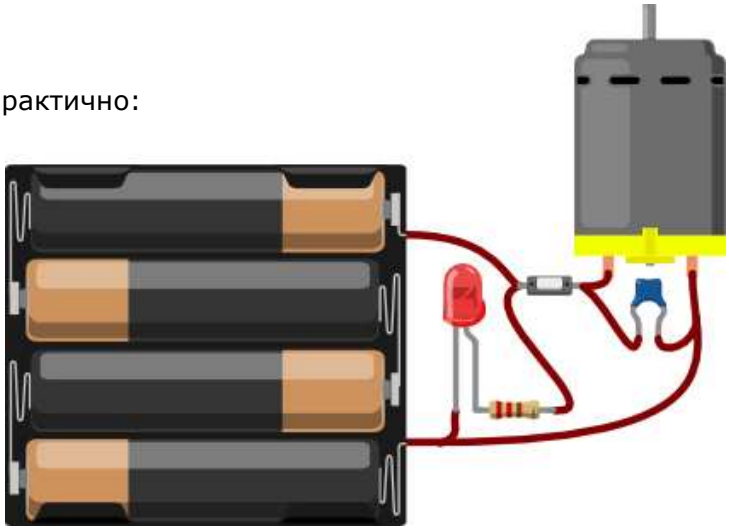
Принципиальные схемы

Чтобы изобразить на бумаге как должна выглядеть та или иная электрическая цепь используют *схемы*. Схемы бывают разных видов со своими преимуществами и недостатками.

Ниже приведена одна и та же электрическая схема, изображённая по-разному, в четырёх вариациях.

Рисованная схема

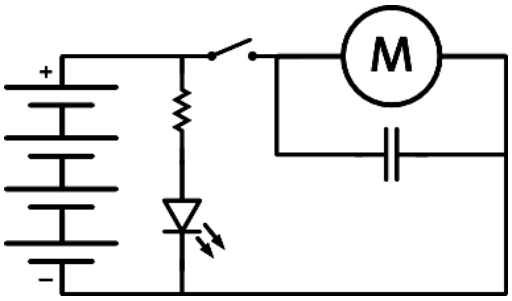
Красиво, но громоздко и непрактично:



Принципиальная схема

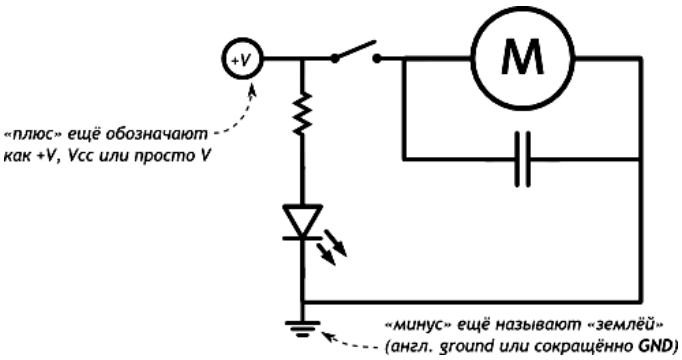
Компактно и наглядно:

- То, что соединено линией, в реальности должно быть соединено проводником
- то, что не соединено линией, в реальности должно быть электрически изолировано



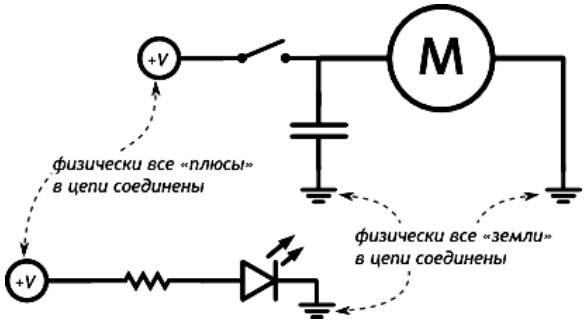
Принципиальная схема без явного источника питания

Источник питания зачастую не рисуют в явном виде, а используют отдельные символы для плюса и минуса. Такая схема ещё более компактна.



Принципиальная схема с отдельными контурами

Часто для удобства одну цепь на схемах разбивают на отдельные части. В сложных проектах так добиваются наглядности и делят зоны ответственности между несколькими инженерами-разработчиками.

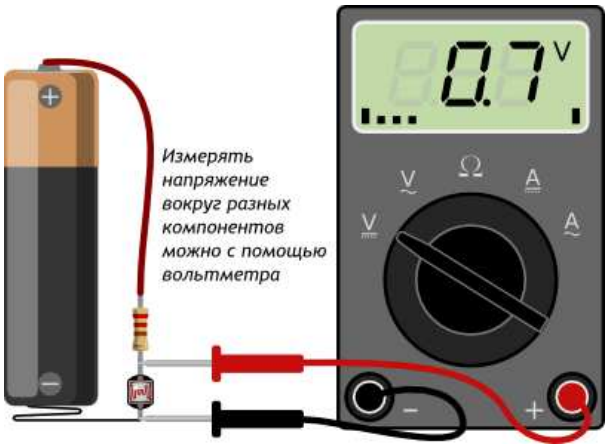


Управление электричеством

Если постоянно и монотонно трансформировать электроэнергию в другую форму, область применения электричества будет сильно ограничена. Огромный мир разнообразных полезных устройств открывается, если научиться контролировать и взаимодействовать с электричеством. Для этого существует несколько способов.

Управление вручную



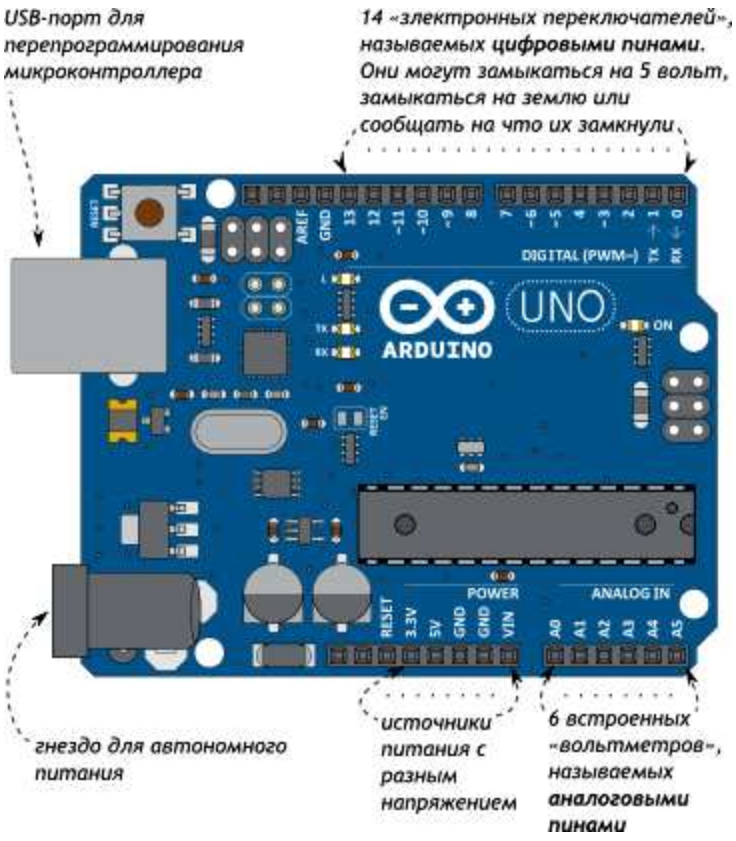


Автоматическое управление

Замыкать и размыкать цепь, измерять напряжение также можно, не вручную, а автоматически, по заданному алгоритму при помощи запрограммированного микроконтроллера.

Существуют «сырые» микроконтроллеры, выполненные в виде одной микросхемы. Они дёшевы при массовом производстве, но их программирование и правильное подключение — нетривиальная задача для новичка.

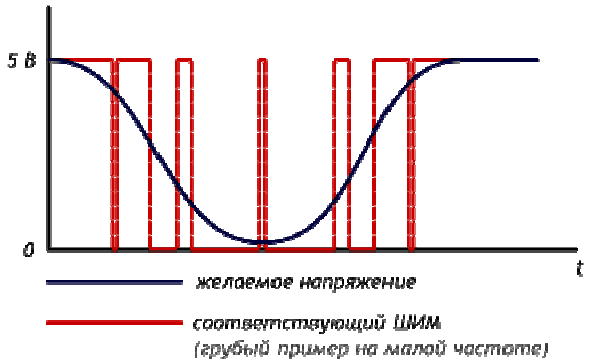
Чтобы решить эту проблему, существуют готовые платы или, как ещё говорят, вычислительные платформы. Они делают процесс взаимодействия с микроконтроллером очень простым. Типичным представителем этого семейства являются платы Arduino.



Широтно-импульсная модуляция

Микроконтроллеры обычно не могут выдавать произвольное напряжение. Они могут выдать либо напряжение питания (например, 5 В), либо землю (т.е. 0 В)

Но уровнем напряжения управляется многое: например, яркость светодиода или скорость вращения мотора. Для симуляции неполного напряжения используется *ШИМ* (Широтно-Импульсная Модуляция, англ. Pulse Width Modulation или просто *PWM*)



Применение

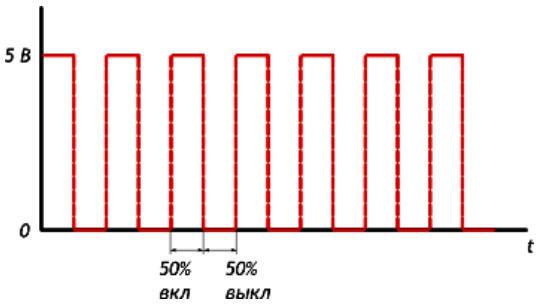
Выход микроконтроллера переключается между землёй и Vcc тысячи раз в секунду. Или, как ещё говорят, имеет частоту в тысячи герц. Глаз не замечает мерцания более 50 Гц, поэтому нам кажется, что светодиод не мерцает, а горит в полсилы.

Аналогично, разогнанный мотор не может остановить вал за миллисекунды, поэтому ШИМ-сигнал заставит вращаться его в неполную силу.

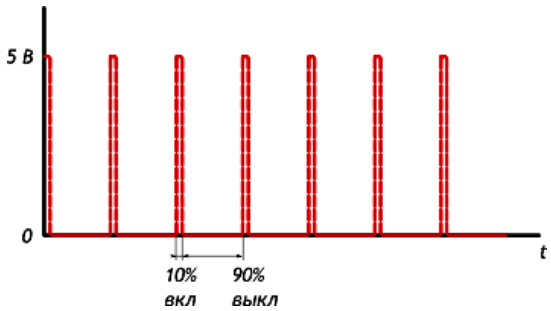
Скважность

Отношение времени включения и выключения называют *скважностью* (англ. duty cycle). Рассмотрим несколько сценариев при напряжении питания *Vcc* равным 5 вольтам.

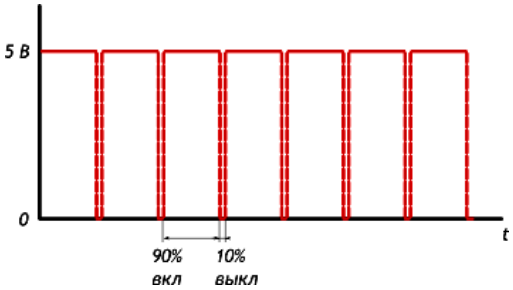
50% — эквивалент 2,5 В



10% — эквивалент 0,5 В



90% — эквивалент 4,5 В



Практикум

- Управление яркостью светодиода с помощью ШИМ в эксперименте «Маячок с нарастающей яркостью»
- Управление большим током с помощью ШИМ и транзистора в эксперименте «Пульсар»

Эксперимент 2. Маячок с нарастающей яркостью

В этом эксперименте мы задаем различные уровни яркости светодиода.

Список деталей для эксперимента
- 1 плата Arduino Uno
- 1 беспаечная макетная плата
- 1 светодиод
- 1 резистор номиналом 220 Ом
- 2 провода «папа-папа»

Для дополнительного задания
- еще 1 светодиод
- еще 1 резистор номиналом 220 Ом
- еще 2 провода

Принципиальная схема

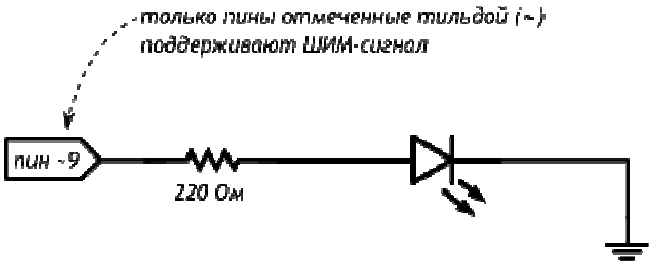
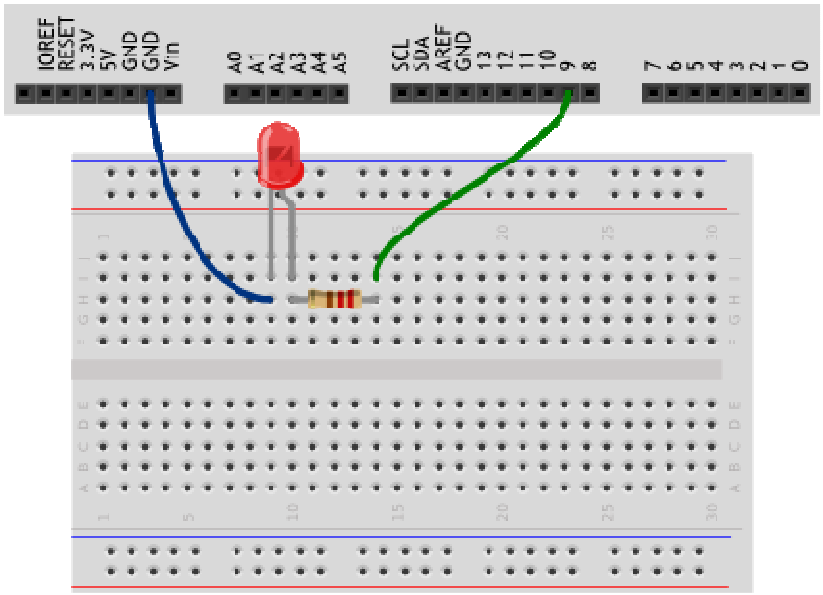


Схема на макетке



Обратите внимание

- Не любой порт Arduino поддерживает широтно-импульсную модуляцию, если вы хотите регулировать напряжение, вам подойдут пины, помеченные символом тильда «~». Для Arduino Uno это пины 3, 5, 6, 9, 10, 11

Скетч

p020_pulse_light.ino

```
// даём разумное имя для пина №9 со светодиодом
// (англ. Light Emitting Diode или просто «LED»)
// Так нам не нужно постоянно вспоминать куда он подключён
#define LED_PIN 9

void setup()
{
  // настраиваем пин со светодиодом в режим выхода,
  // как и раньше
  pinMode(LED_PIN, OUTPUT);
```

```
}

void loop()
{
  // выдаём неполное напряжение на светодиод
  // (он же ШИМ-сигнал, он же PWM-сигнал).
  // Микроконтроллер переводит число от 0 до 255 к напряжению
  // от 0 до 5 В. Например, 85 — это 1/3 от 255,
  // т.е. 1/3 от 5 В, т.е. 1,66 В.
  analogWrite(LED_PIN, 85);
  // держим такую яркость 250 миллисекунд
  delay(250);

  // выдаём 170, т.е. 2/3 от 255, или иными словами — 3,33 В.
  // Больше напряжение — выше яркость!
  analogWrite(LED_PIN, 170);
  delay(250);

  // все 5 В — полный накал!
  analogWrite(LED_PIN, 255);
  // ждём ещё немного перед тем, как начать всё заново
  delay(250);
}
```

Пояснения к коду

- Идентификаторы переменных, констант, функций (в этом примере идентификатор `LED_PIN`) являются одним словом (т.е. нельзя создать идентификатор `LED PIN`).
- Идентификаторы могут состоять из латинских букв, цифр и символов подчеркивания `_`. При этом идентификатор не может начинаться с цифры.

```
PRINT       // верно
PRINT_3D    // верно
MY_PRINT_3D // верно
_PRINT_3D   // верно
3D_PRINT    // ошибка
ПЕЧАТЬ_3Д   // ошибка
PRINT:3D    // ошибка
```

- Регистр букв в идентификаторе имеет значение. Т.е. `LED_PIN`, `LED_pin` и `led_pin` с точки зрения компилятора — различные идентификаторы
- Идентификаторы, создаваемые пользователем, не должны совпадать с предопределенными идентификаторами и стандартными конструкциями языка; если среда разработки подсветила введенный идентификтор каким-либо цветом, замените его на другой
- Директива `#define` просто говорит компилятору заменить все вхождения заданного идентификатора на значение, заданное после пробела (здесь `9`), эти директивы помещают в начало кода. В конце данной директивы точка с запятой `;` не допустима

- Названия идентификаторов всегда нужно делать осмысленными, чтобы при возвращении к ранее написанному коду вам было ясно, зачем нужен каждый из них
- Также полезно снабжать код программы комментариями: в примерах мы видим однострочные комментарии, которые начинаются с двух прямых слэшей `//` и многострочные, заключённые между `/*` `*/`

```
// однострочный комментарий следует после двойного слеша до конца строки
/* многострочный комментарий
   помещается между парой слеш-звездочка и звездочка-слеш */
```

комментарии игнорируются компилятором, зато полезны людям при чтении давно написанного, а особенно чужого, кода

- Функция `analogWrite(pin, value)` не возвращает никакого значения и принимает два параметра:
  - `pin` — номер порта, на который мы отправляем сигнал
  - `value` — значение скважности ШИМ, которое мы отправляем на порт. Он может принимать целочисленное значение от 0 до 255, где 0 — это 0%, а 255 — это 100%

Вопросы для проверки себя

1. Какие из следующих идентификаторов корректны и не вызовут ошибку?
   - `13pin`
   - `MOTOR_1`
   - `контакт_светодиода`
   - `sensor value`
   - `leftServo`
   - `my-var`
   - `distance_eval2`
2. Что произойдет, если создать директиву `#define HIGH LOW`?
3. Почему мы не сможем регулировать яркость светодиода, подключенного к порту 7?
4. Какое усреднённое напряжение мы получим на пине 6, если вызовем функцию `analogWrite(6, 153)`?
5. Какое значение параметра `value` нужно передать функции `analogWrite`, чтобы получить усреднённое напряжение 2 В?

Задания для самостоятельного решения

1. Отключите питание, отключите светодиод от 9-го порта и подключите к 11-му. Измените программу так, чтобы схема снова заработала
2. Измените код программы так, чтобы в течение секунды на светодиод последовательно подавалось усреднённое напряжение 0, 1, 2, 3, 4, 5 В
3. Возьмите еще один светодиод, резистор на 220 Ом и соберите аналогичную схему на этой же макетке, подключив светодиод к пину номер 3 и другому входу GND, измените программу так, чтобы светодиоды мигали в противофазу: первый выключен, второй горит максимально ярко и до противоположного состояния



Гололобов В. Н.

С чего начинаются роботы?
О проекте Arduino для школьников (и не только)

— М.: Электронное издание, 2011. — 189 с.: ил.

Интернет-ресурс: http://vgololobov.narod.ru/content/arduino/intro.html

Есть такой открытый проект, который называется Arduino. Основа этого проекта - базовый аппаратный модуль и программа, в которой можно написать код для контроллера на специализированном языке, и которая позволяет этот модуль подключить и запрограммировать.
Модуль легко соединяется с разными исполняющими устройствами, позволяя создавать и роботов, и устройства автоматики, и приборы.

С момента появления проекта Arduino у него появилось множество почитателей - достаточно ввести в поисковую строку слово arduino, как вы обнаружите сотни сайтов, посвященных этой теме, сотни проектов, основанных на Arduino.

На английском языке издано несколько книг. И эта должна восполнить пробел в части книг на русском языке. Хотя книга рассчитана на школьников, она может быть интересна радиолюбителям, и, если не книга, то сам проект может быть интересен преподавателям, и не только работающим в школе, но и в других учебных заведениях, где изучают программирование и работу с микроконтроллерами.

СОДЕРЖАНИЕ

Глава 1. Паровозик из Ромашково, начало
Глава 2. Установка программы Arduino в ALTLinux
Глава 3. Введение в работу с программой Arduino
Глава 4. Введение в язык программирования Arduino
Глава 5. Arduino, визуальное программирование
Глава 6. Введение в язык программирования Scratch
Глава 7. Отладка программы на виртуальной плате
Глава 8. Немного больше о программе VirtualBreadboard
Глава 9. Паровозик из Ромашково, продолжение
Глава 10. С чего начинаются роботы?

Приложение А. О языке программирования Arduino
Перевод на русский язык:
Brian W. Evans, Arduino Programming Notebook
Arduino блокнот программиста

Приложение Б. Работа с модулем Arduino в других средах разработки
AVR Studio, WinAVR, FlowCode for AVR, AVR-GCC

Так что же эта книга?

Она в основном описывает ряд программ, которые предназначены для работы с модулем Arduino, как сама программа Arduino, как S4A, как VirtualBreadBoard... Если все эти программы почти обычным образом устанавливаются в Windows, то в Linux, а они работают и в этой операционной системе, есть особенности, которые описаны в этой книге. Повышенное внимание к Linux в последнее время делает актуальным подобное описание.

Помимо этого в книге рассказано о средах разработки AVR-контроллеров общего назначения, которые поддерживают работу с модулем Arduino - AVR Studio, WinAVR, FlowCode. В основном касательно настройки для работы с Arduino.

Но почему о программах, если речь идёт о роботах?

Суть любого робота - это аппаратные средства с процессорной базой и программа (или набор программ). Поэтому программирование неотъемлемый элемент процесса создания даже самого простого робота.

Научившись программировать модуль Arduino, а программа приходит с огромным набором примеров, касающихся всех областей применения модуля, вы будете готовы создавать интересные и полезные электронные устройства, к которым относятся и роботы. Начните с простых проектов, а остальное в ваших руках.

Возможно школьники, познакомившись с увлекательнейшим проектом Arduino, выберут роботостроение своей будущей профессией. Но даже если нет, то, уверен, через много лет, вспоминая свои эксперименты в этой области, они будут рады, что не прошли мимо, не пожалели времени на освоение основ - им будет, что вспомнить.

Радиолюбители давно и успешно осваивают работу с микроконтроллерами. Они с упоением спорят, какой язык программирования лучше. Возможно, проект Arduino позволит им сделать окончательный выбор? Тем более, что модуль Arduino может работать как программатор для программирования других микроконтроллеров.

Словом, всё интересное и полезное, что есть в проекте Arduino, можно узнать, только работая с ним в компании таких же увлечённых людей.

Глава 4. Введение в язык программирования Arduino

Основа языка программирования модуля Arduino — это язык Си (скорее Си++). Ещё точнее, этот диалект языка называется Processing/Wiring. Хорошее обозрение языка вы найдёте в приложении. А мне хочется больше рассказать не о языке, а о программировании.

Программа — это некий набор команд, которые понимает процессор, процессор вашего компьютера или процессор микроконтроллера модуля Arduino, не суть важно. Процессор читает команды и выполняет их. Любые команды, которые понимает процессор — это двоичные числа. Это только двоичные числа и ничто иное. Выполняя арифметические операции, для которых процессор некогда и предназначался, процессор оперирует с числами. Двоичными числами. И получается, что и команды, и то, к чему они относятся, это только двоичные числа. Вот так. Но как же процессор разбирается в этой «куче» двоичных чисел?

Во-первых, все эти двоичные числа записываются в последовательные ячейки оперативной памяти, имеющие адреса. Когда вы загружаете программу, и она начинает работать, процессор получает первый адрес программы, где обязательно должна быть записана команда. Те команды, которые требуют от процессора операций с числами, имеют «опознавательные знаки», например, что в следующих двух ячейках памяти два числа, которые нужно сложить. А счётчик, назовём его счётчиком команд, где записан адрес следующей команды, в данном случае увеличивает адрес так, что в программе по этому адресу будет следующая команда. При неправильной работе программы или сбоях процессор может ошибиться, и тогда, прочитав вместо команды число, процессор делает совсем не то, что должен делать, а программа «зависает».

Таким образом, любая программа — это последовательность двоичных чисел. А программирование — это умение правильно записывать правильные последовательности двоичных чисел. Достаточно давно для записи программ стали использовать специальные средства, которые называются языками программирования.

Однако любая программа в первую очередь требует от вас ясного понимания того, что должна делать программа, и для чего она нужна. Чем яснее вы это понимаете, тем легче создать программу. Небольшие программы, хотя трудно сказать, какие программы небольшие, а какие нет, можно рассматривать целиком. Более сложные программы лучше разбить на части, которые можно рассматривать как самостоятельные программы. Так их лучше создать, легче отладить и проверить.

Я не готов спорить, но считаю, что программу удобнее начинать с описания на обычном языке. И в этом смысле я считаю, что программирование не следует путать с написанием кода программы. Когда программа описана обычными словами, вам легче определить, например, какой язык программирования выбрать для создания кода программы.

Ближе всего к записи программы с помощью двоичных чисел, язык ассемблер. Для него характерно соответствие команд языка двоичным командам, понятным процессору. Но кодирование программ на ассемблере требует больших усилий и ближе к искусству, чем к формальным операциям. Более универсальны и легче в применении языки высокого уровня, как Бэйсик или Си. И давно для записи программ в общем виде используют графический язык, а в последнее время появились и «переводчики» с этого языка на язык процессоров.

Кроме языков программирования общего применения, всегда существовала некоторая специализация языков программирования, и существовали специализированные языки. К последним я бы отнёс и язык программирования модуля Arduino.

Всё, что нужно сказать модулю, чтобы он сделал что-то нужное нам, организовано в удобный набор команд. Но вначале о том, что нам нужно от Arduino?

Модуль можно использовать в разных качествах — это и сердце (или голова) робота, это и основа прибора, это и удобный конструктор для освоения работы с микроконтроллерами и т.д.

Выше мы уже использовали простые программы для проверки подключения модуля к компьютеру. Кому-то они могут показаться слишком простыми, а поэтому не интересными, но любые сложные программы состоят из более простых фрагментов, похожих на те, с которыми мы уже знакомились.

Давайте посмотрим, о чём нам может рассказать самая простая программа «Помигать светодиодом».

```
int ledPin = 13;

void setup()
{
  pinMode (ledPin, OUTPUT);
}

void loop()
{
  digitalWrite (ledPin, HIGH);
  delay (1000);
  digitalWrite (ledPin, LOW);
  delay (1000);
}
```

Вначале вспомним, что такое светодиод. В сущности это обычный диод, у которого, благодаря его конструкции, при протекании тока в прямом направлении начинает светиться переход. То есть, чтобы светодиод светился, нужно чтобы через него протекал ток, а, значит, к светодиоду следует приложить напряжение. А чтобы ток не превысил допустимого значения, последовательно со светодиодом следует включить резистор, который называют токоограничительным (см. Приложение А, цифровой выход). Напряжение к светодиоду прикладывает микроконтроллер, составляющий основу модуля Arduino. У микроконтроллера, кроме процессора, выполняющего наши команды, есть один или несколько портов ввода-вывода. Не вдаваясь в рассмотрение конкретного устройства порта, скажем так — когда вывод порта работает на выход, его можно представить как выход цифровой микросхемы с двумя состояниями, включено и выключено (есть напряжение на выходе, нет напряжения на выходе).

Но этот же вывод порта может работать и как вход. В этом случае его можно представить, например, как вход цифровой микросхемы – на вход подаётся логический уровень, высокий или низкий (см. Приложение А, цифровой ввод).

Как мы мигаем светодиодом:

```
Включить выходной вывод порта.
Выключить вывод порта.
```

Но процессор работает очень быстро. Мы не успеем заметить мигания. Чтобы заметить это мигание, нам нужно добавить паузы. То есть:

```
Включить выходной вывод порта.
Пауза 1 секунда.
Выключить вывод порта.
Пауза 1 секунда.
```

Это наша программа. Процессор прочитает первую команду и включит вывод, светодиод загорится. Затем процессор сделает паузу в работе и выключить вывод, светодиод погаснет. Но он только один раз мигнул.

Повторение какого-либо процесса или набора команд называется в программировании циклом. Используются разные виды циклов. Есть цикл, который выполняется заданное число раз. Это цикл for. Есть циклы, которые выполняются до тех пор, пока не будет выполнено некоторое условие, которое является частью языковой конструкции цикла. А если условие не будет выполнено никогда, то цикл выполняется бесконечное число раз. Это бесконечный цикл.

Я не думаю, что микроконтроллеры используются с программами того вида, который приведён выше. То есть, один раз выполнено несколько команд и больше контроллер не работает. Как правило, он работает постоянно, как только на него подаётся питающее напряжение. А, значит, микроконтроллер должен работать в бесконечном цикле.

Именно об этом говорит функция void loop(), loop — это петля, замкнутый цикл. Условия прекращения работы цикла нет, а, следовательно, нет условия его завершения.

Кроме того, мы должны сообщить модулю Arduino, какой вывод порта и как мы хотим использовать, для выхода (OUTPUT) или для входа (INPUT). Этой цели служит функция void setup(), которая для языка Arduino является обязательной, даже если она не используется, и команда pinMode(), для задания режима работы вывода.

```
void setup()
{
  pinMode (ledPin, OUTPUT);
}
```

И ещё, языковая конструкция использует переменные для определения номера вывода:

```
int ledPin = 13;
```

Использование переменных удобно. Решив, что вы будете использовать не вывод 13, а 12, вы внесёте изменение только в одной строке. Особенно сильно это сказывается в больших программах. Имя переменной можно выбирать по своему усмотрению, но, как правило, оно должно быть только символьным, и часто количество символов ограничивается. Если вы неверно зададите имя переменной, думаю, компилятор вас поправит.

Функция digitalWrite (ledPin, HIGH) устанавливает заданный вывод в состояние с высоким уровнем, то есть включает вывод.

A delay (1000), как вы уже поняли, означает паузу в 1000 миллисекунд или 1 секунду.

Осталось понять, что означают такие приставки, как int, void. Любые значения, любые переменные размещаются в памяти, как и команды программы. В ячейки памяти записываются числа зачастую из 8 битов. Это байт. Но байт — это числа от 0 до 255. Для записи больших чисел нужно два байта или больше, то есть, две или больше ячеек памяти. Чтобы процессору было ясно, как отыскать число, разные типы чисел имеют разные названия. Так число по имени byte, займёт одну ячейку, int (integer, целое) больше. Кроме того, функции, используемые в языках программирования, тоже возвращают числа. Чтобы определить, какой тип числа должна вернуть функция, перед функцией записывают этот тип возвращаемого числа. Но некоторые функции могут не возвращать числа, такие функции предваряют записью void (см. Приложение А, переменные).

Вот, сколько интересного может рассказать даже самая простая программа.

Обо всём этом вы, надеюсь, прочитаете в приложении. А сейчас проделаем простые эксперименты, используя только то, что мы уже знаем из возможностей языка. Первое, заменим переменную типа int, которая занимает много места в памяти, на byte — одно место, одна ячейка памяти. Посмотрим, что у нас получится.

```
byte ledPin = 13;

void setup()
{
  pinMode (ledPin, OUTPUT);
}

void loop()
{
  digitalWrite (ledPin, HIGH);
```

```
  delay (1000);
  digitalWrite (ledPin, LOW);
  delay (1000);
}
```

После компиляции и загрузки программы в модуль мы не заметим изменений в работе программы. Хорошо. Тогда изменим программу так, чтобы заметить изменения в её работе.

Для этого мы заменим число в функции delay (1000) переменной, назвав её my_del. Эта переменная должна быть целым числом, то есть, int.

```
int my_del = 5000;
delay(my_del);
```

Не забывайте заканчивать каждую команду точкой с запятой. Внесите изменения в программу, скомпилируйте её и загрузите в модуль. Затем поменяйте переменную и повторите компиляцию и загрузку:

```
byte my_del = 5000;
```

Разница, уверен, получится ощутимая.

Проделаем ещё один эксперимент с изменением длительности пауз. Уменьшение длительности пауз выполним, скажем, пять раз. Сделаем паузу в 2 секунды, а затем будем увеличивать тоже пять раз. И вновь сделаем паузу в 2 секунды. Цикл, выполняемый заданное количество раз, называется циклом for и записывается он так:

```
for (int i = 0; i<5; i++)
{
что-то, что выполняется в цикле for
}
```

Для выполнения цикла ему нужна переменная, у нас это i, переменной нужно задать начальное значение, которое мы ей и присвоили. Затем следует условие завершения работы цикла, у нас i меньше 5. А запись i++ — это характерная для языка Си запись увеличения переменной на единицу. Фигурные скобки ограничивают набор команд, подлежащих выполнению в цикле for. В других языках программирования могут быть другие ограничители для выделения блока кода функции.

Внутри цикла мы выполняем то же, что и раньше, с небольшими изменениями:

```
for (int i = 0; i<5; i++)
{
      digitalWrite (ledPin, HIGH);
      delay (my_del);
      digitalWrite (ledPin, LOW);
      delay (my_del);
      my_del = my_del – 100;
}
```

Об изменении записи паузы мы говорили выше, а изменение самой паузы достигается уменьшением переменной на 100.

Для второго цикла мы запишем этот же блок кода, но переменную длительности паузы будем увеличивать на 100.

```
for (int i = 0; i<5; i++)
{
      digitalWrite (ledPin, HIGH);
      delay (my_del);
      digitalWrite (ledPin, LOW);
      delay (my_del);
      my_del += 100;
}
```

Вы заметили, что запись уменьшения паузы и её увеличения выглядят по-разному. Это тоже особенность языка Си. Хотя для ясности следовало повторить эту запись, изменив только знак минус на плюс. Итак, мы получаем такую программу:

```
int ledPin = 13;
int my_del = 1000;


void setup()
      {
  pinMode (ledPin, OUTPUT);
            }

void loop()
     {
    for (int i = 0; i<5; i++)
    {
          digitalWrite (ledPin, HIGH);
          delay (my_del);
          digitalWrite (ledPin, LOW);
          delay (my_del);
          my_del -= 100;
    }
  delay (2000);
    for (int i = 0; i<5; i++)
    {
          digitalWrite (ledPin, HIGH);
          delay (my_del);
          digitalWrite (ledPin, LOW);
          delay (my_del);
          my_del += 100;
    }
  delay (2000);
      }
```

Скопируем код нашей программы в программу Arduino, скомпилируем её и загрузим в модуль. Изменение длительности пауз заметно. И будет ещё заметнее, попробуйте, если цикл for выполнить, скажем, раз 8.

То, что мы сейчас сделали, делают и профессиональные программисты — имея готовую программу, её легко можно модифицировать под свои нужды или желания. Поэтому все свои программы они хранят. Что я советую делать и вам.

Что мы упустили в своём эксперименте? Мы не прокомментировали нашу работу. Для добавления комментария используется либо двойная «прямая» косая черта, либо одиночная, но со звёздочками (см. Приложение А). Я советую вам это сделать самостоятельно, поскольку вернувшись к программе через некоторое время, вы легче в ней разберётесь, если будут пояснения, что вы делаете в том или ином месте программы. И ещё советую в папке с каждой программой хранить её описание на обычном языке, выполненное в любом текстовом редакторе.

Самая простая программа «помигать светодиодом» может послужить ещё для десятка экспериментов (даже с одним светодиодом). Мне кажется эта часть работы, придумывать, что ещё можно сделать интересного, самая интересная. Если вы обратитесь к приложению, где описан язык программирования, к разделу «управление программой», то можно заменить цикл for на другой вид цикла. И попробовать, как работают другие виды цикла.

Хотя процессор микроконтроллера, как любой другой, может производить вычисления (для того его и придумывали), и это используется, например, в приборах, всё-таки наиболее характерной операцией для микроконтроллера будет установка выхода порта в высокое или низкое состояние, то есть, «помигать светодиодом», как реакция на внешние события.

О внешних событиях микроконтроллер узнаёт, в основном, по состоянию входов. Настроив выводы порта на цифровой вход, мы можем следить за ним. Если исходное состояние входа — высокий уровень, а событие вызывает переход входа в низкое состояние, то мы можем что-то сделать, реагируя на это событие.

Самый простой пример — на входе кнопка. Когда кнопка не нажата, вход в высоком состоянии. Если нажать кнопку, то вход переходит в низкое состояние, а мы можем «зажечь» светодиод на выходе. При следующем нажатии на кнопку светодиод можно погасить.

Это опять пример простой программы. Даже начинающему она может показаться неинтересной. Однако и эта простая программа может найти вполне полезное применение. Приведу только один пример: мы будем после нажатия на кнопку не зажигать светодиод, а помигаем (определённым образом). И светодиод возьмём с инфракрасным излучением. В результате мы получим пульт управления. Вот такая простая программа.

В разных версиях программы есть различия в списке примеров. Но можно обратиться к руководству по языку в приложении, где есть пример и схема программы (в разделе примеров, названном «приложение») для работы с вводом. Я скопирую программу:

```
int ledPin = 13;
int inPin = 2;

void setup()
{
          pinMode (ledPin, OUTPUT);
          pinMode (inPin, INPUT);
    }

    void loop()
    {
          if (digitalRead(inPin) == HIGH)
          {
                digitalWrite(ledPin, HIGH);
                delay (1000);
                digitalWrite (ledPin, LOW);
                delay (1000);
          }
}
```

И, как вы видите, совершенно новую программу мы получаем, модифицируя старую. Теперь светодиод будет мигать только тогда, когда нажата кнопка, которая присоединена к выводу 2. Вывод 2 через резистор 10 кОм присоединён к общему проводу (земле, GND). Кнопка одним концом присоединена к питающему напряжению +5В, а другим концом к выводу 2.

В программе мы встречаем новую языковую конструкцию if из раздела управления программой. Читается она так: если выполняется условие (заключённое в скобках), то выполняется блок программы, заключённый в фигурные скобки. Обратите внимание, что в условии (digitalRead(inPin) == HIGH) равенство входа высокому состоянию выполнено с помощью двух знаков равенства! Очень часто в спешке об этом забывается, и условие получается неверным.

Программу можно скопировать и загрузить в модуль Arduino. Однако, чтобы проверить работу программы, понадобиться внести некоторые изменения в конструкцию модуля. Впрочем, это зависит от разновидности модуля. Оригинальный модуль имеет розетки для соединения с платами расширения. В этом случае можно вставить подходящие одножильные провода в нужные места разъёма. Мой модуль имеет ножевые контакты для соединения с платами расширения. Я могу либо поискать подходящий разъём, либо, что дешевле, использовать подходящую панельку для микросхемы в корпусе DIP.

Второй вопрос — как найти у модуля те выводы, которые используются в программе?

С этим вопросом поможет разобраться картинка, которую я взял с сайта: http://robocraft.ru/.

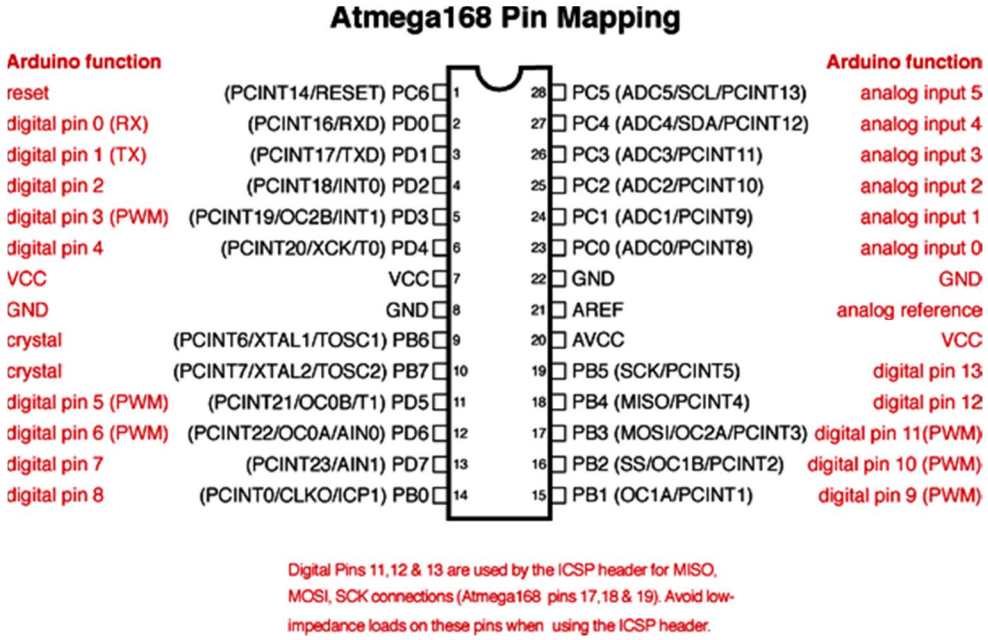


Рис. 4.1. Расположение и назначение выводов контроллера и модуля Arduino

Все выводы моего модуля CraftDuino промаркированы, так что найти нужный вывод не составит труда. Можно подключать кнопку и резистор и проверять работу программы. Кстати, на вышеупомянутом сайте RoboCraft весь процесс отображён на картинках (но программа использует не совсем такие выводы!). Советую посмотреть.

Многие микроконтроллеры в своём составе имеют дополнительные аппаратные устройства. Так Atmega168, на основе которого собран модуль Arduino имеет UART, встроенный блок для связи с другими устройствами с помощью последовательного обмена данными. Например, с компьютером через COM-порт. Или с другим микроконтроллером с помощью его встроенного блока UART. Есть ещё и аналого-цифровой преобразователь. И формирователь широтно-импульсной модуляции.

Использование последнего иллюстрирует программа, которую я тоже скопирую с сайта RoboCraft. Но программу можно взять и из приложения. И, возможно, она есть в примерах программы Arduino.

```
// Fading LED by BARRAGAN <http://people.interaction-ivrea.it/h.barragan>
int value = 0;    // переменная для хранения нужного значения
int ledpin = 9;   // светодиод подключен к digital pin 9

void setup()
{
 // Нет необходимости вызвать функцию pinMode
}
void loop()
{
      for(value = 0 ; value <= 255; value+=5) // постепенно зажигаем светодиод
      {
      analogWrite(ledpin, value); // значение вывода (от 0 до 255)
      delay(30);                  // ждём :)
      }
      for(value = 255; value >=0; value-=5)  // постепенно гасим светодиод
      {
      analogWrite(ledpin, value);
      delay(30);
      }
}
```

Если в предыдущей программе новой для нас была функция digitalRead(inPin), чтение цифрового ввода, то в этой программе новая для нас функция analogWrite(ledpin, value), хотя параметры этой функции — уже знакомые нам переменные. Об использовании аналогового входа, использовании АЦП (аналого-цифрового преобразователя), мы поговорим позже. А сейчас вернёмся к общим вопросам программирования.

Программирование это то, что доступно всем, но потребуется время, чтобы освоить и программирование, и какой-либо язык программирования. Сегодня есть ряд программ, помогающих освоить именно программирование. И одна из них имеет непосредственное отношение к модулю Arduino. Называется она Scratch for Arduino или сокращённо S4A. Найти и скачать эту программу можно по адресу: http://seaside.citilab.eu/scratch/arduino. Я не знаю, как точно переводится название программы, но «to begin from scratch» переводится, как «начать с нуля».

На сайте проекта S4A есть версии для Windows и Linux, но для последней операционной системы готовая к установке программа в версии дистрибутива Debian. Не хочу сказать, что её нельзя использовать с другими дистрибутивами Linux, но вначале посмотрим, как работать в программе с модулем Arduino в Windows.

После установки программы обычным образом можно настроить интерфейс на русский язык, используя переключатель языков.

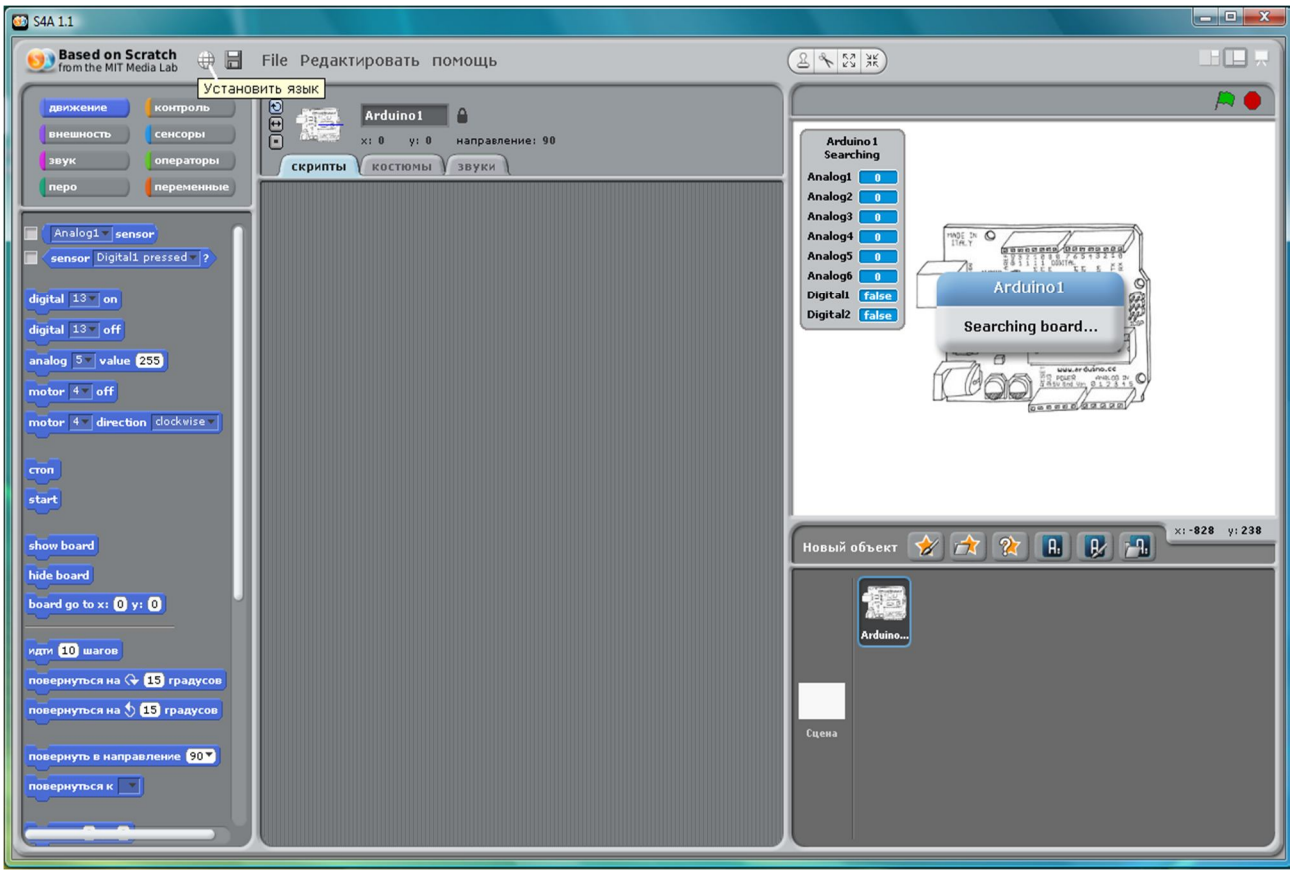


Рис. 4.2. Переключатель языков интерфейса программы

Первый значок инструментальной панели, если его нажать, отображает все возможные языки интерфейса программы. Русский язык можно найти в разделе…

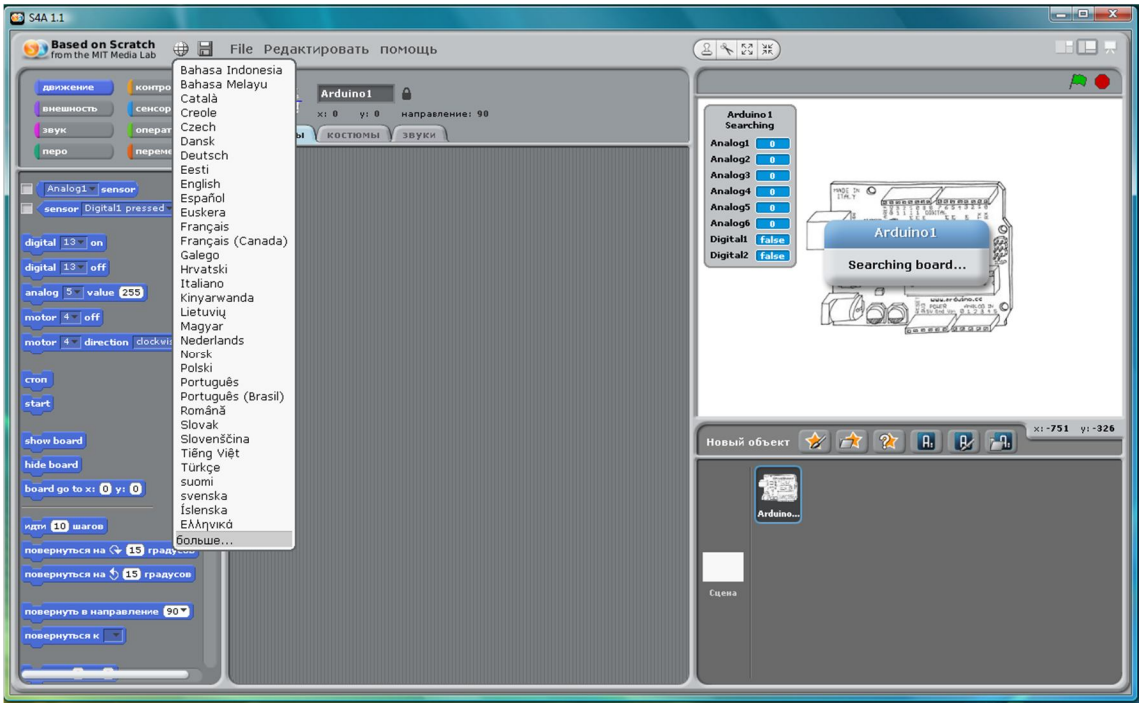


Рис. 4.3. Список языков для использования в интерфейсе программы

… отмеченном, как «больше…».

Если ничего не предпринимать, то надпись в правом окне «Searching board…» остаётся, но модуль не находится. Чтобы модуль Arduino подключить к программе S4A, следует загрузить с сайта проекта ещё кое-что.

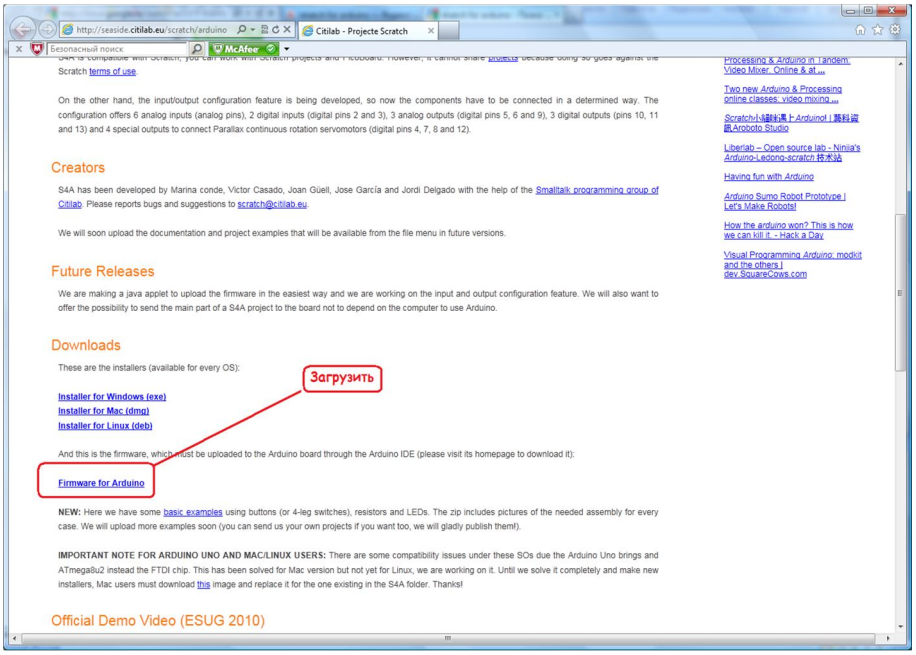


Рис. 4.4. Файл для загрузки в модуль Arduino для S4A

Этот файл не что иное, как программа для Arduino (Sketch). То есть, текстовый файл, который можно скопировать в редактор Arduino, откомпилировать и загрузить в модуль. После выхода из программы Arduino можно запустить программу S4A и теперь модуль находится.

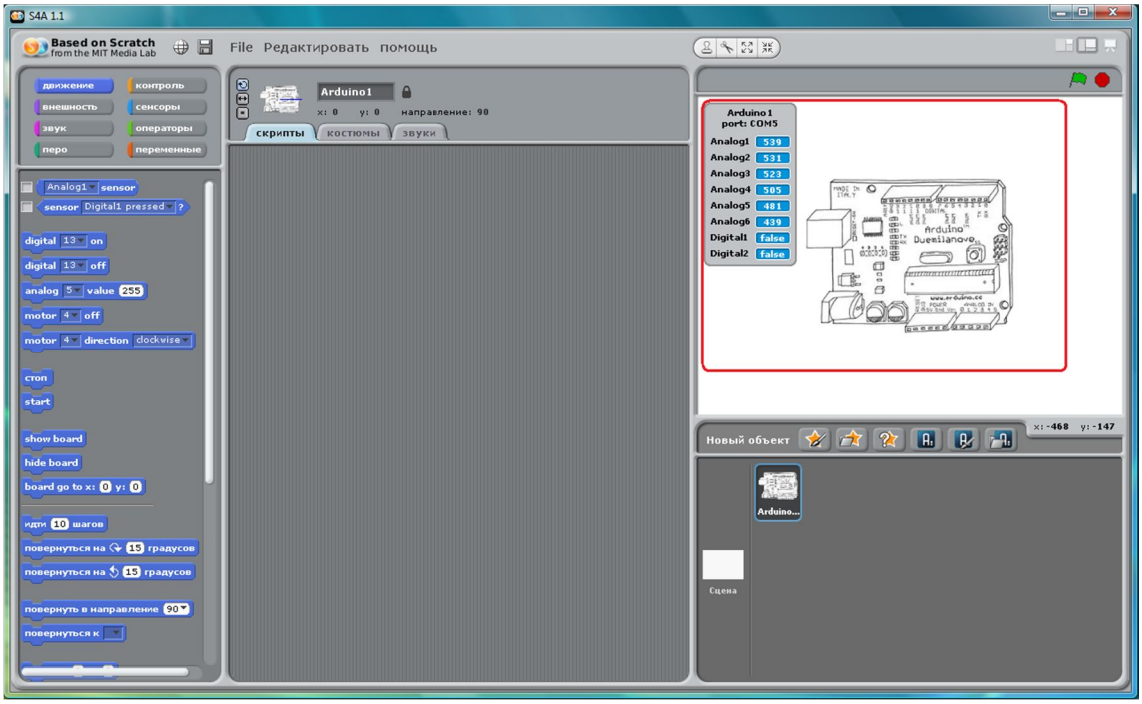


Рис. 4.5. Подключение модуля к программе

Аналоговые входы модуля не подключены, как и цифровые, поэтому значения, отображаемые для модуля, постоянно меняются произвольным образом.

Глава 5. Arduino, визуальное программирование

Возможно, правы разработчики операционных систем, считающие пользователя злейшим врагом и самым опасным вирусом. А, может быть, не правы, создают они свои творения не для себя, а для пользователей. Словом, не знаю. Но, что точно знаю, я хочу видеть работающую программу S4A не только в Windows, но в Linux, и не только в дистрибутиве Debian.

Начинаю я этот процесс с загрузки версии для Debian на сайте разработчиков: http://seaside.citilab.eu/scratch/arduino. Все загружаемые файлы располагаются по завершении загрузки в папке «Загрузка» или «Downloads». Архивированные файлы, предназначенные для Linux, распаковываются менеджером архивов. Скачанный мной в openSUSE файл имеет расширение deb, но, используя Ark, тот самый менеджер архивов, его можно распаковать. В openSUSE с графическим менеджером KDE 4 для этого достаточно щёлкнуть правой клавишей мышки по файлу и выбрать пункт выпадающего меню «Распаковать во вложенную папку». В итоге появляется папка с именем S4A.

Заглянем в неё.

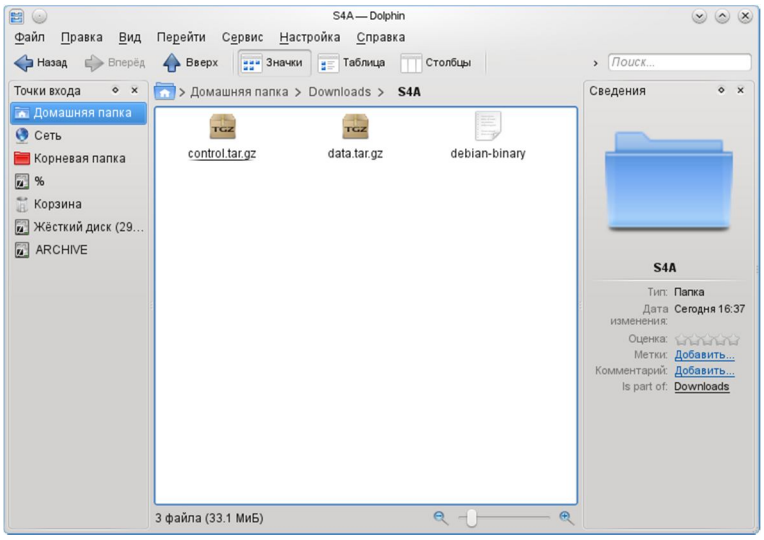


Рис. 5.1. Содержимое скачанной папки S4A

Два файла с расширением tar.gz подлежат дальнейшей разархивации.

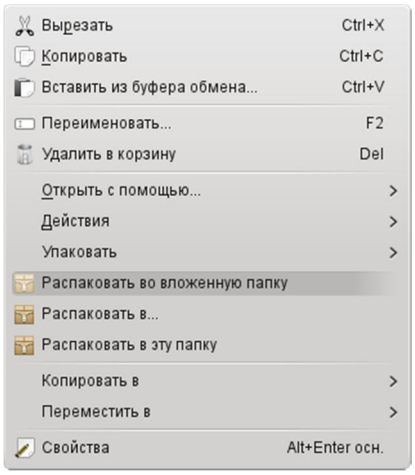


Рис. 5.2. Выпадающее меню работы с архивированными файлами

В результате рядом с архивами появляется ряд файлов и папка, озаглавленная «usr». Из опыта работы с Linux я знаю, что в этой папке могут находиться файлы, которые при установке размещаются по адресу /usr корневой файловой системы. Если открыть эту папку, то,

действительно, в ней можно увидеть еще три папки.

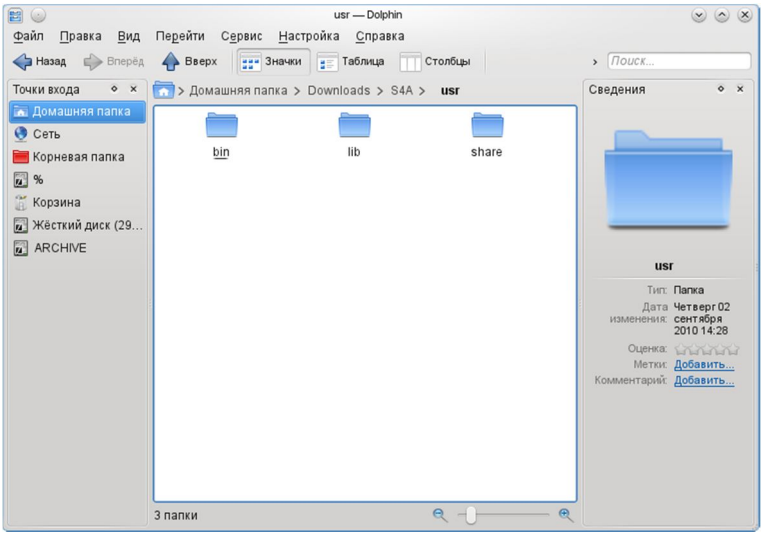


Рис. 5.3. Содержание распакованного файла

Эти три папки соответствуют разделам, которые можно увидеть, если открыть в файловом менеджере раздел «Корневая папка» в директории /usr.

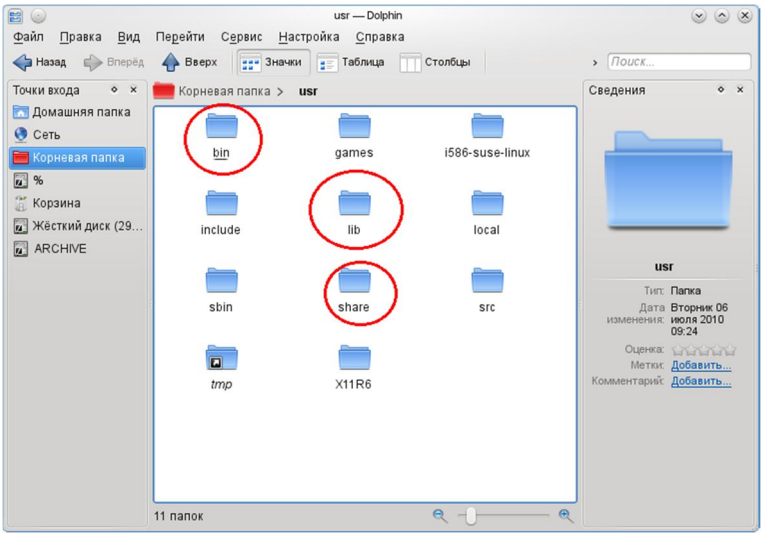


Рис. 5.4. Разделы директории usr файловой системы

Содержимое, скачанное ранее, папок bin, lib и share, как я полагаю, следует разместить в папки, отмеченные выше. Но, конечно, простому пользователю менять что-то в файловой системе никто не позволит. Поэтому в разделе основного меню «Система» находим пункт «Файловый менеджер», открывающий новое подменю, где есть «Менеджер файлов (с правами администратора)». Этот менеджер позволит перенести все нужные файлы в операционную систему. Ничего не выдумывая, открывая параллельно папки в двух проводниках, просто последовательно открывать нужные (они все названы) папки до появления файлов, а файлы копировать.

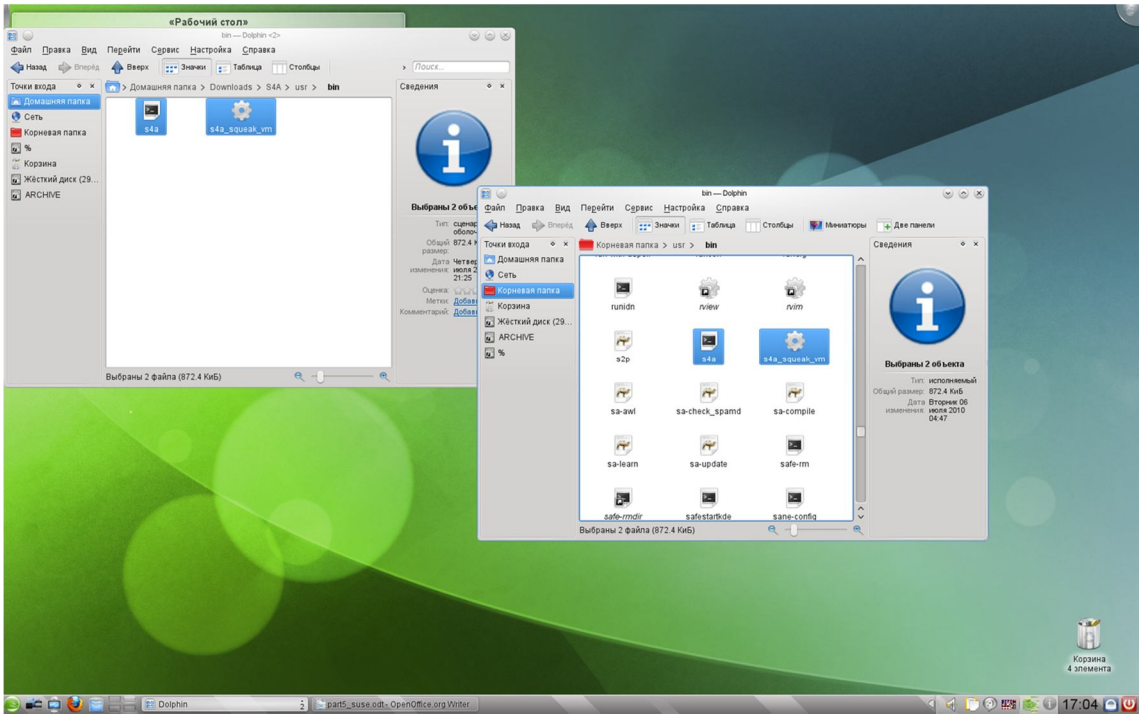


Рис. 5.5. Перенос файлов программы в openSUSE

Особенно внимательно следует отнестись к папке share, поскольку она имеет много вложенных папок, и соответствующие папки следует отыскивать в файловой системе.

Завершив копирование, можно попытаться отыскать программу в основном меню. И, впрямь, на закладке «Приложения» в разделе «Разработка» (у меня ещё один раздел «Другие программы») появляется программа S4A. И её даже можно запустить. Но она после нескольких движений мышкой начинает виснуть...

В терминале, а в openSUSE есть терминал с правами суперпользователя; от имени суперпользователя, предварительно подключив модуль Arduino, запускаем программу. И она работает. Теперь её можно запустить обычным образом.

В других дистрибутивах Linux операции схоже с теми, что описаны выше, отличия не столь значительны. Хотя в Fedora 14 я просто сменил пользователя, войдя в систему под root, что делать, конечно, не следует, но так было проще всё разместить в нужных местах.

Установив программу в Linux, посмотрим, а для чего мы её устанавливали?

Во-первых, программа работает с модулем, показывая, что происходит на аналоговых и цифровых входах модуля. Что уже неплохо. Но не это главное. Главное, во-вторых — программа позволяет собирать программу, а не кодировать на языке Arduino.

Когда программа начинает работать, в левом окне есть ряд элементов, которые можно, подцепив мышкой, перенести в среднее окно — рабочее «сборочное» поле. Перенесём так элемент, который называется Start.

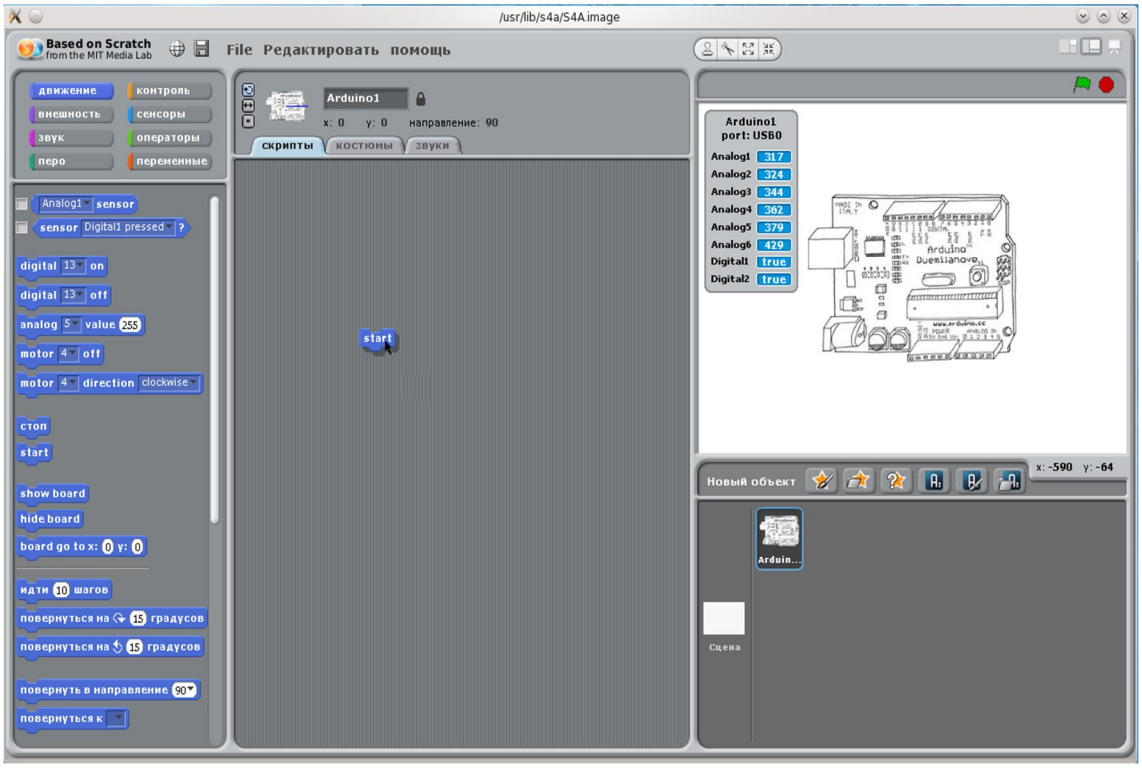


Рис. 5.6. Перенос нужных программных элементов

Теперь, нажав клавишу с надписью «контроль» в окошке чуть выше, получим ряд новых элементов.

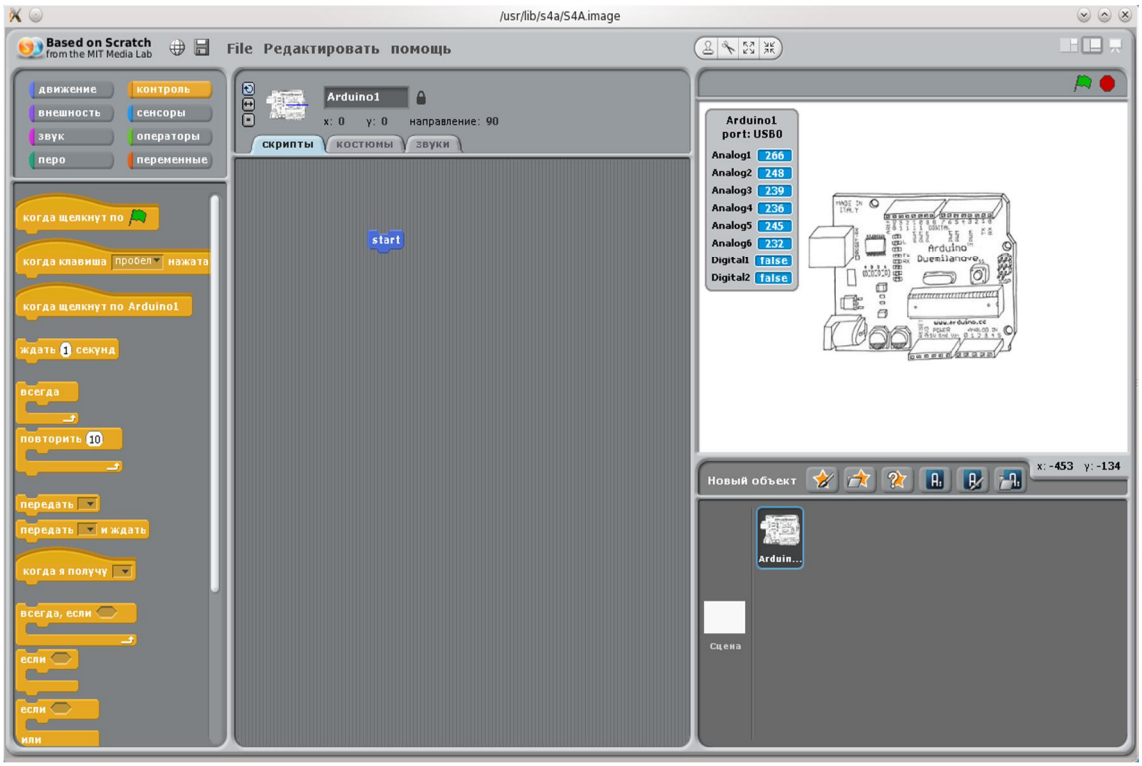


Рис. 5.7. Список элементов в группе «Контроль»

Среди этих элементов выберем элемент «всегда», который перенесём к уже имеющемуся элементу, и добавим так, чтобы верхний вырез вошёл в выступ.



Рис. 5.8. Добавление элементов в программу

Вернёмся к набору элементов, с которого начинали, нажав на клавишу «движение», и выберем элемент «digital 13 on», который перенесём и положим внутрь предыдущего.

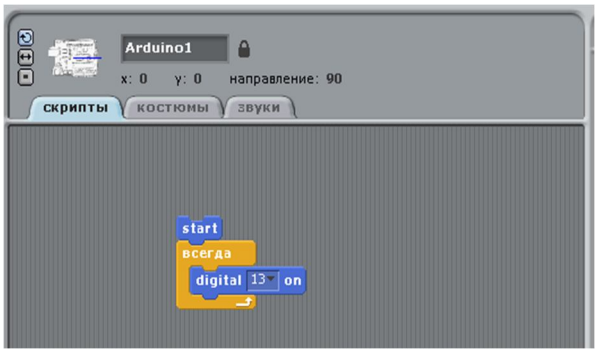


Рис. 5.9. Команда включения цифрового вывода

Из набора элементов «контроль» возьмём элемент «ждать 1 секунду», который вставим внутрь элемента «всегда» под элемент «digital 13 on». Чтобы ускорить этот процесс, вставим элемент ожидания ещё раз, вернёмся к элементам движения и добавим элемент «digital 13 off» между двумя элементами ожидания.

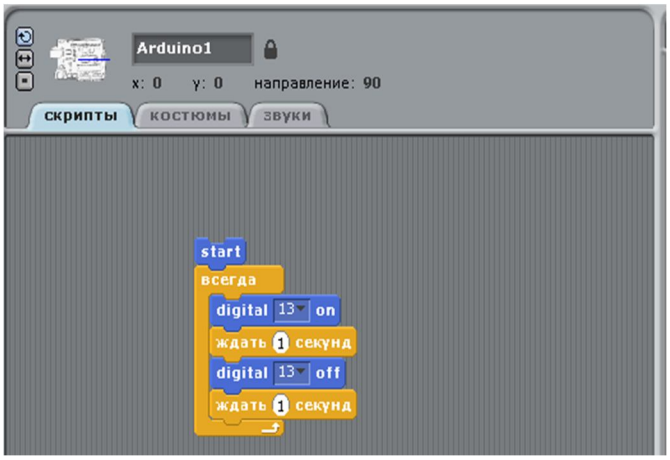


Рис. 5.10. Программа Blink в графическом виде

Вам эта конструкция ничего не напоминает? Когда мы начинали описывать первую программу обычным языком, мы так и записывали её.

Дважды щёлкните по элементу «start» левой клавишей мышки и посмотрите на модуль Arduino — молчавший до сих пор светодиод на выводе 13 исправно мигает раз в секунду.

Мы собрали программу, запустили её и заставили работать модуль согласно этой программе. И мы не написали ни строчки кода. Именно по этой причине я предпочитаю различать программирование и написание программного кода.

Но, может быть, это работает ранее загруженная программа, а не нами собранная?

Остановим работу программы, вновь дважды щёлкнув мышкой по элементу «start». Щёлкнем левой клавишей по единичке элемента «ждать 1 секунду».

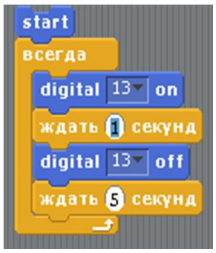


Рис. 5.11. Изменение параметров программных элементов

Впечатаем цифру 5 (как на нижнем элементе). Запустим программу... и убедимся, ничего мы не перепутали, светодиод мигает с интервалом раз в 5 секунд!

Мы не проверяли работу цифрового входа в «живом» виде. Не пора ли это сделать?

Соберём программу в S4A. Первые «кирпичики» те же, что и в предыдущей программе. Далее... нам понадобится выполнить условие: если кнопка нажата, включить светодиод, иначе выключить. Такой элемент есть — это «если... или...». В его верхней части есть «гнездо», куда можно вставить нужное нам условие «цифровой вход...».

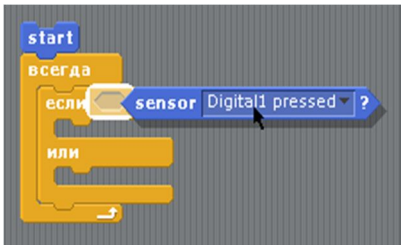


Рис. 5.12. Добавление условия в элемент if ветвления программы

Осталось добавить действия, чтобы получить нужный вид программы.

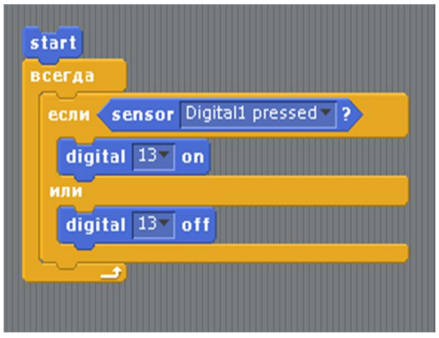
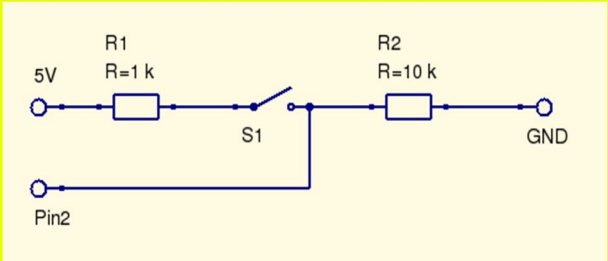


Рис. 5.13. Окончательное формирование программы

Если сравнивать её с программой, написанной на языке Arduino, то можно сказать, что отличия только те, что были внесены сознательно: когда кнопка отпущена, светодиод не горит, когда нажата, светодиод загорается.

Пора перейти к проверке. Но прежде небольшое предупреждение.

На схеме, приведённой в примерах, кнопка соединяется с выводом +5 В. Я бы советовал включить её несколько иначе.



Особенно, если вы проверяете все «на весу». При случайной ошибке может получиться так, как было у меня, из модуля пойдёт дымок, который очень подпортит настроение. А самый правильный путь — использовать макетную плату с переходными разъёмами (для Arduino Nano, думаю, найдётся подходящая панелька под микросхему).

Проверив правильность соединений на макетной плате, подключив к ней модуль Arduino, можно включить его в разъём USB компьютера и запустить программу S4A. Обратите внимание — когда вы между цифровым входом и землёй включили резистор 10 кОм, показания (в правом окошке программы) перестали случайным образом меняться между «false, ложно» и «true, истинно». Запускаем нашу программу двойным щелчком по элементу «start», добавим, зайдя в раздел основного меню «Редактировать», пошаговое выполнение.

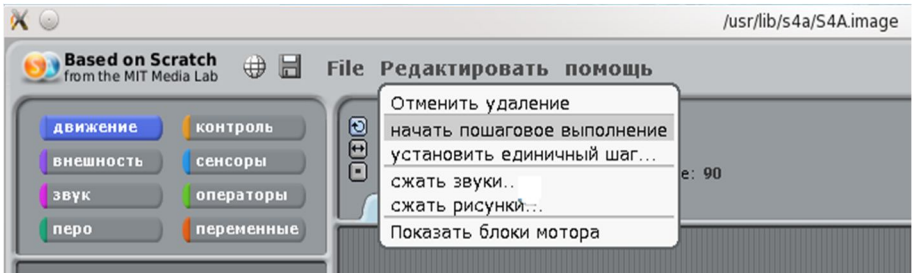


Рис. 5.14. Добавление пошагового выполнения в отладочную процедуру

Можно ещё в пункте «установить единичный шаг...» выбрать скорость выполнения. И теперь, пока кнопка не нажата, мы видим, что светодиод не горит, а программа выполняется только в той части, где это задано.

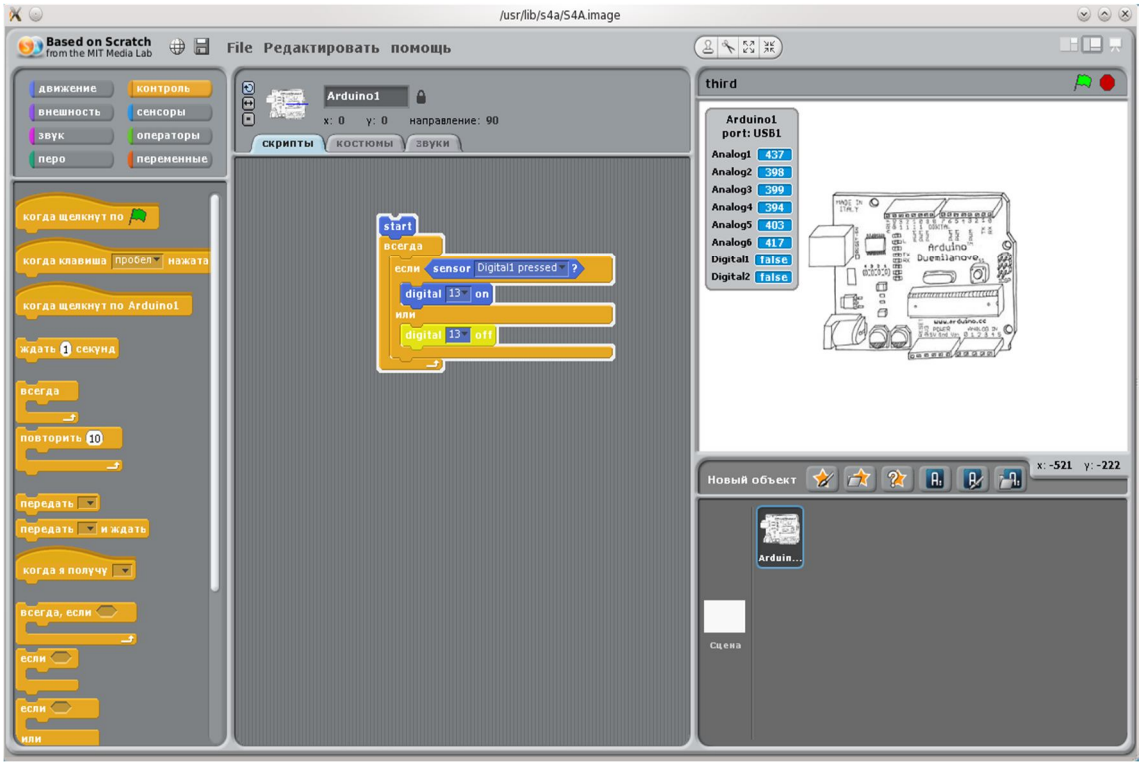


Рис. 5.15. Выполнение программы в режиме отладки

В правом верхнем окошке можно видеть состояние входа Digital1 — false. Вход на земле, на входе низкий логический уровень, а это, с точки зрения программы, состояние «ложно». Нажмём кнопку.

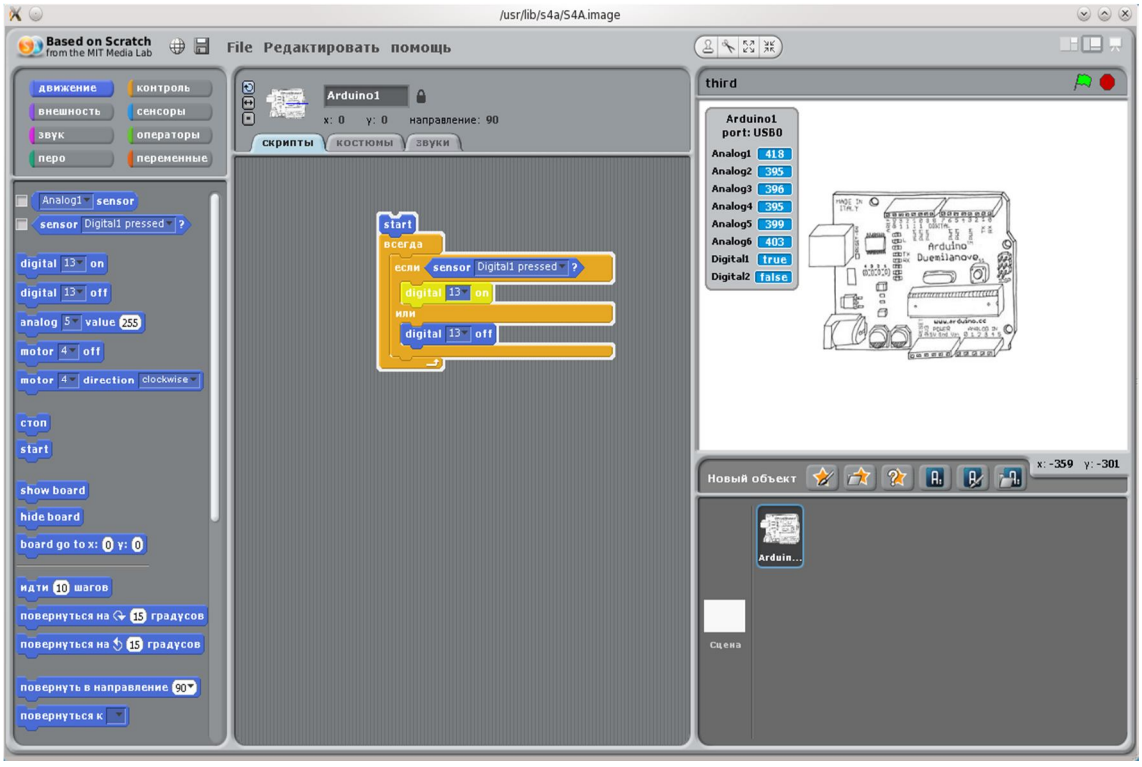
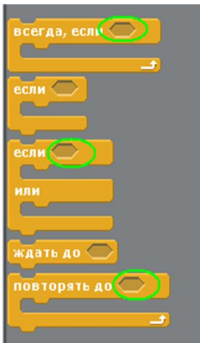


Рис. 5.16. Работа программы при нажатой кнопке

Изменилось состояние входа «true», горит светодиод, и программа входит в ту часть, где условие выполнено.

Если обратить внимание на оранжевые элементы в разделе «контроль», то видно, многие из

них имеют «гнёзда» для вставки условий.



Условия могут быть разными. Выше мы использовали в качестве условия изменение состояния цифрового входа. Но это могут быть и другие условия.

Рис. 5.17. Гнёзда для добавления условий в элементах контроля

И ещё — обратите внимание на чёрные стрелочки «вниз» рядом со многими элементами. Нажимаем эту стрелочку с помощью мышки...

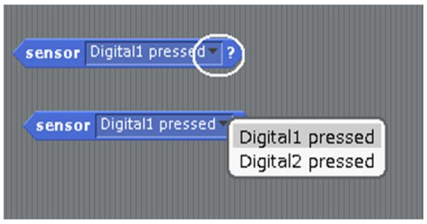


Рис. 5.18. Стрелка, открывающая список возможных сенсоров

... и получаем возможность менять, например, как в этом случае используемый вход. В других случаях меняется, скажем, выходной вывод или аналоговый вход. У нас большой выбор возможностей для экспериментов с модулем Arduino. Впрочем, отчего с модулем? Мы вправе использовать несколько модулей. Достаточно, не скажу, что это единственный способ, перейти на закладку «костюмы», щёлкнуть правой клавишей мышки по существующему «костюму» и выбрать раздел «переключиться к новому объекту».

Появится ещё один модуль Arduino. Если у вас он есть, если вы его подключили к USB порту, то можно, думаю, с ним тоже работать.

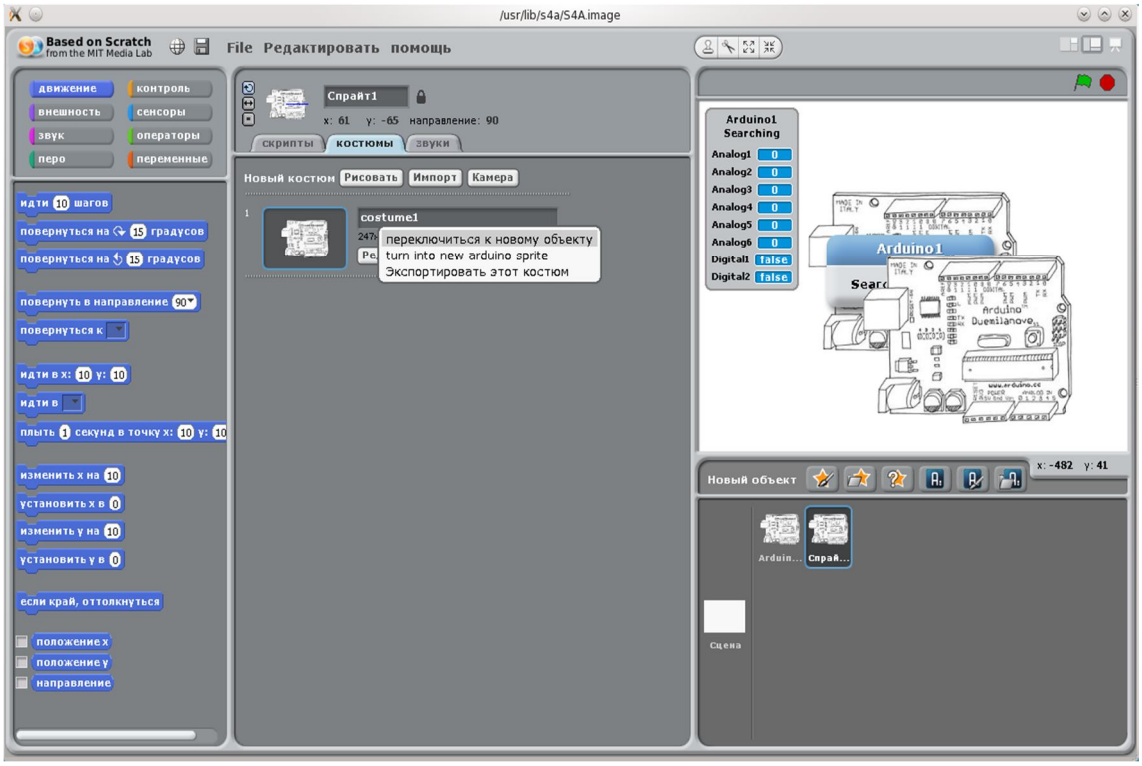


Рис. 5.19. Добавление второго модуля Arduino

И последнее замечание. Всё, что мы делаем в программе S4A, мы делаем, используя язык программирования Scratch. Как вам это?

Глава 6. Введение в язык программирования Scratch

Мы использовали программу S4A для работы с модулем Arduino. Но программа позволяет сделать много больше. Она помогает научиться программировать (не путать с написанием кода программы на традиционных языках программирования).

Для Linux, если заглянуть в раздел /usr/lib/s4a, то можно увидеть папку:

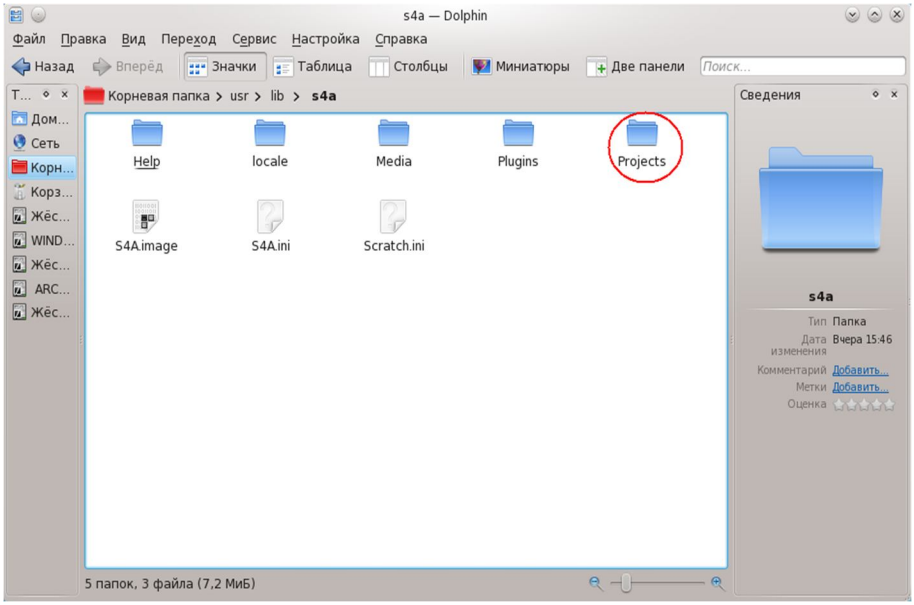


Рис. 6.1. Место расположения учебных проектов S4A

Запустим программу, выберем в основном меню раздел «File» и подраздел «Открыть...». Появится диалоговое окно выбора проекта. Используем клавишу «Компьютер», чтобы начать перемещение по файловой системе. Используем движок прокрутки справа (или колёсико мышки), чтобы перемещаться вниз, и двойным щелчком левой клавиши мышки будем открывать нужные нам папки.

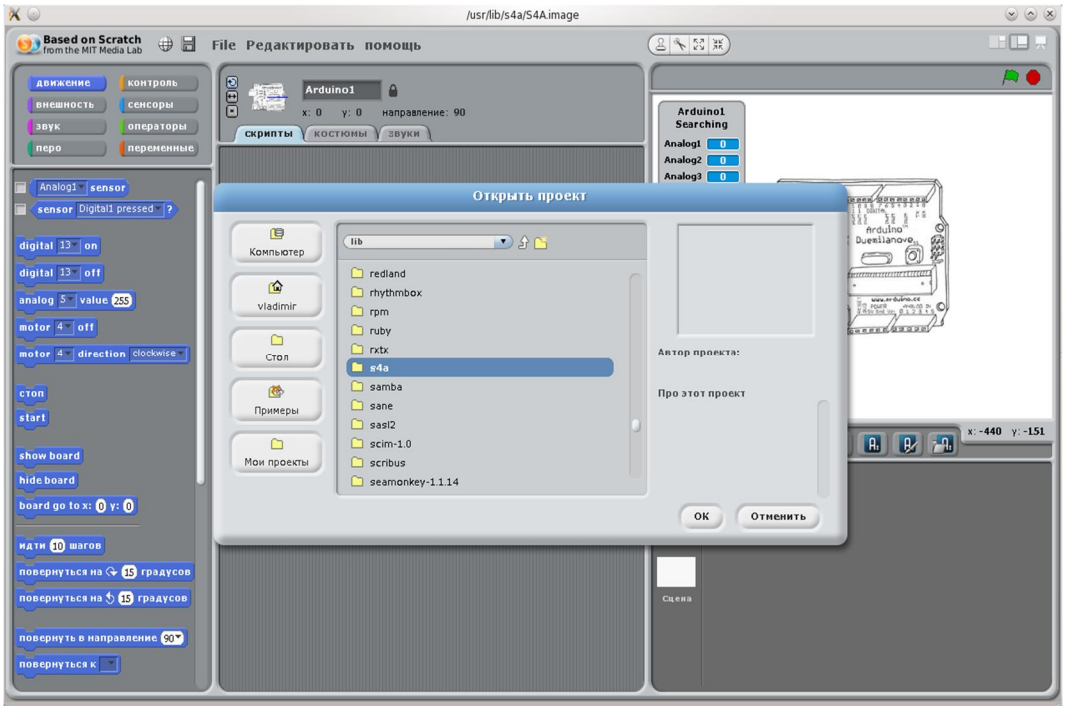


Рис. 6.2. Менеджер файлов в программе S4A

Дойдя до папки s4a, откроем её, откроем папку «Projects», в которое тоже много папок, но мы используем сейчас первую «Animation», где выберем проект Playground.

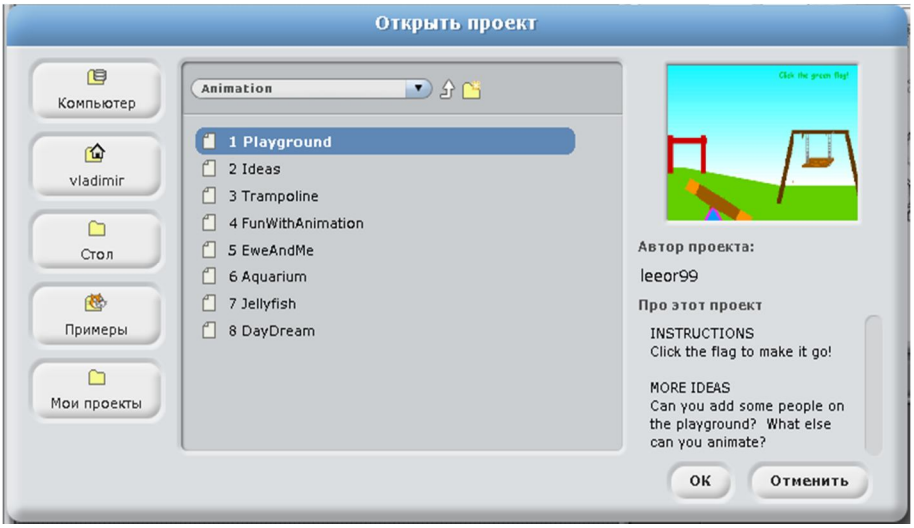


Рис. 6.3. Содержание папки примеров

Можно выбрать другой проект. И я думаю, вам интересно будет посмотреть все предложенные проекты, но для начала разберёмся с первым.

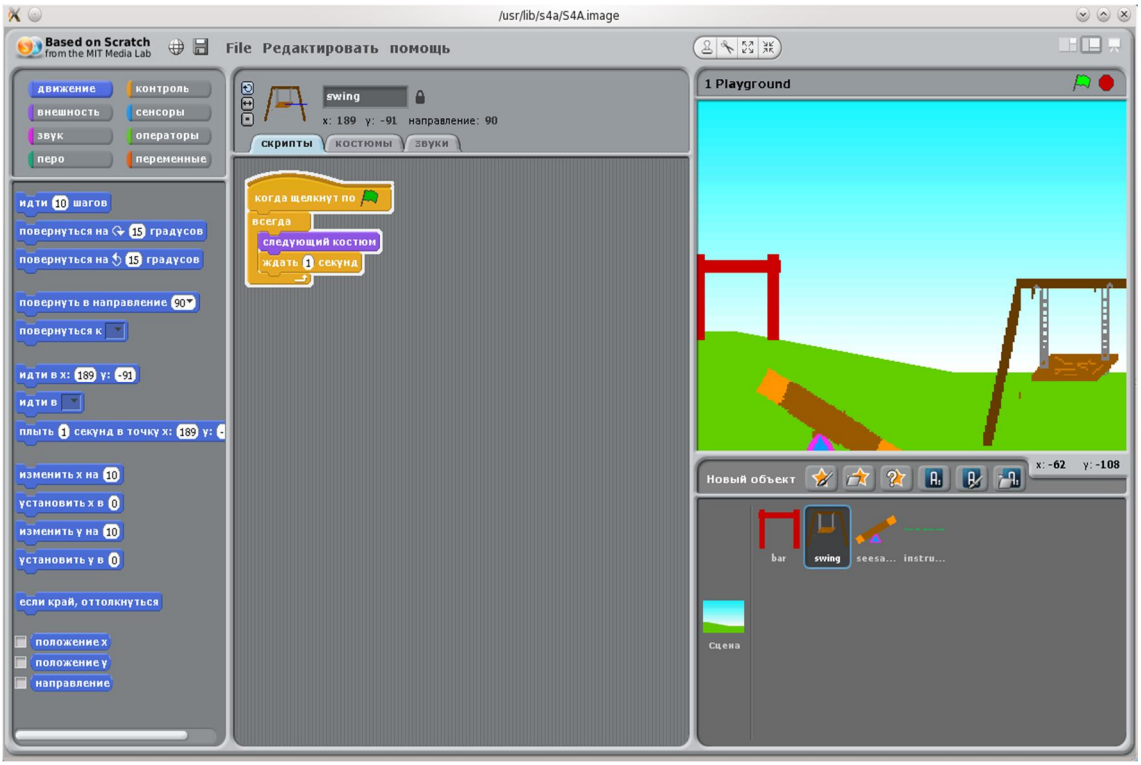


Рис. 6.4. Один из проектов примеров

Щёлкнув по флажку над правым верхним окошком, вы увидите, как качели начинают качаться. Это работы программы, которая собрана в центральном окошке, и которая начинается с условия «когда щёлкнут по флажку».

Пользователи Windows найдут аналогичные примеры по адресу: C:\Program Files\S4A\Projects.

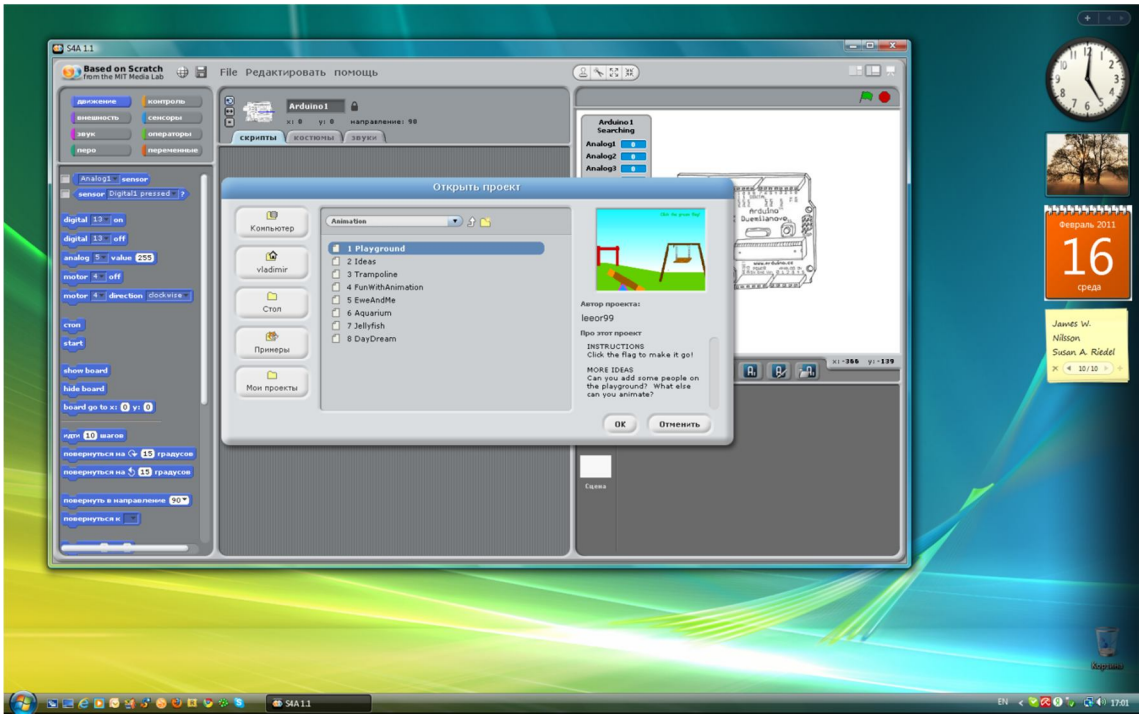


Рис. 6.5. Расположение примеров в Windows

Сразу скажу, что есть руководства по использованию языка Scratch на русском языке, которые можно найти на сайте: http://info.scratch.mit.edu/ru/Languages.

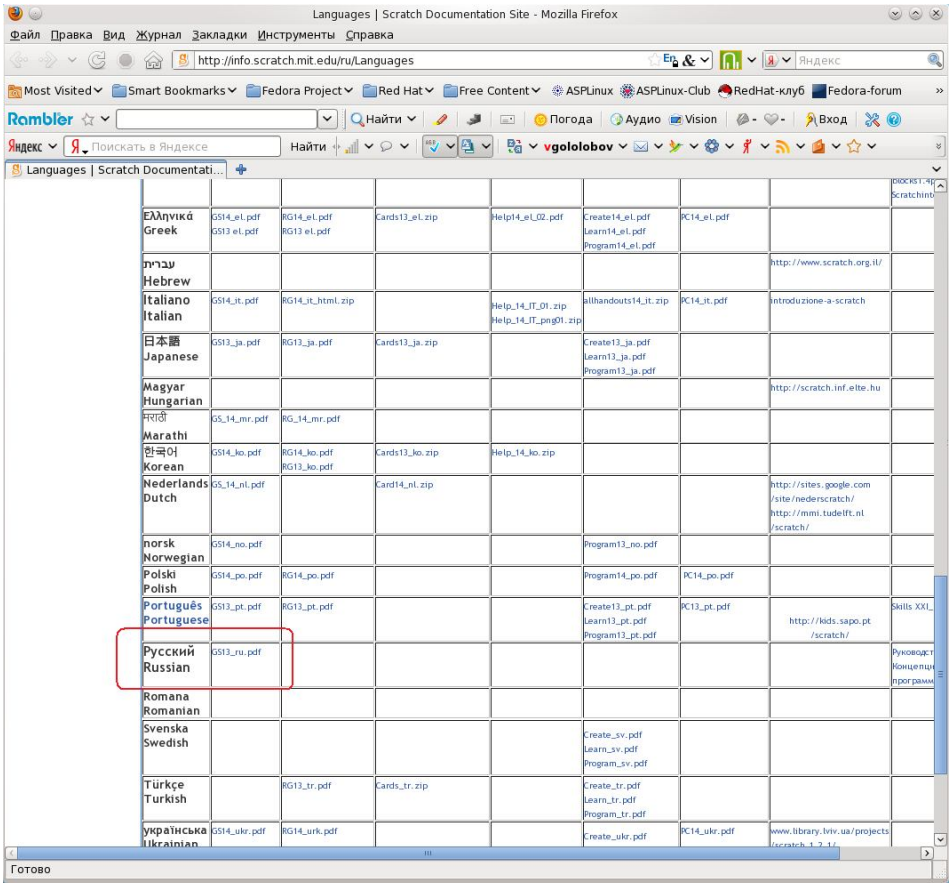


Рис. 6.6. Русскоязычные руководства к программе S4A

Я не вижу смысла пересказывать их в этой книге, но для тех, кто не найдёт эти руководства, я приведу пример, взятый из первой главы «Начнём сначала», благо есть некоторые отличия в

программах руководства и нашей S4A.

Для соответствия описания, сделанного для программы Scratch, откроем проект в папке «Greetings» с длинным названием «HellowInManyLanguages». Удалим скрипт — щелчок правой клавишей мышки по блоку «когда щёлкнут по флажку» программы и «Удалить» из выпадающего меню. Первое, что предлагается сделать в руководстве — перетащить блок «идти».

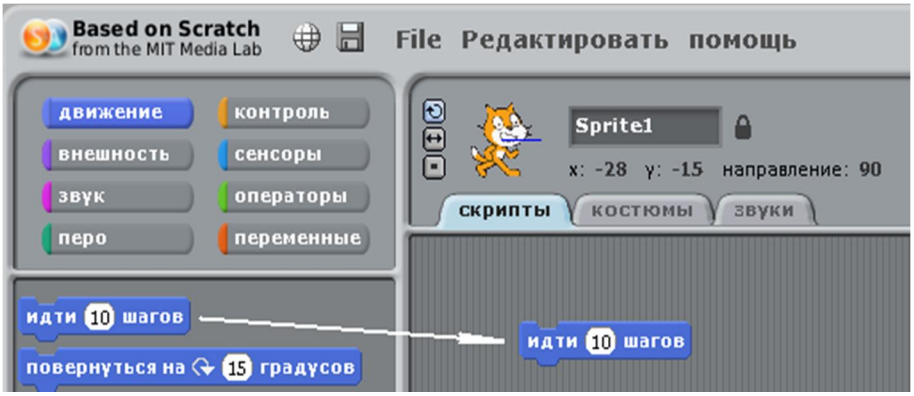


Рис. 6.7. Начало работы с программой

Если сделать двойной щелчок мышкой по этому блоку, котик сдвинется на картинке вправо. Затем из раздела «звук» перетаскиваем новый блок, который соединяем с предыдущим.

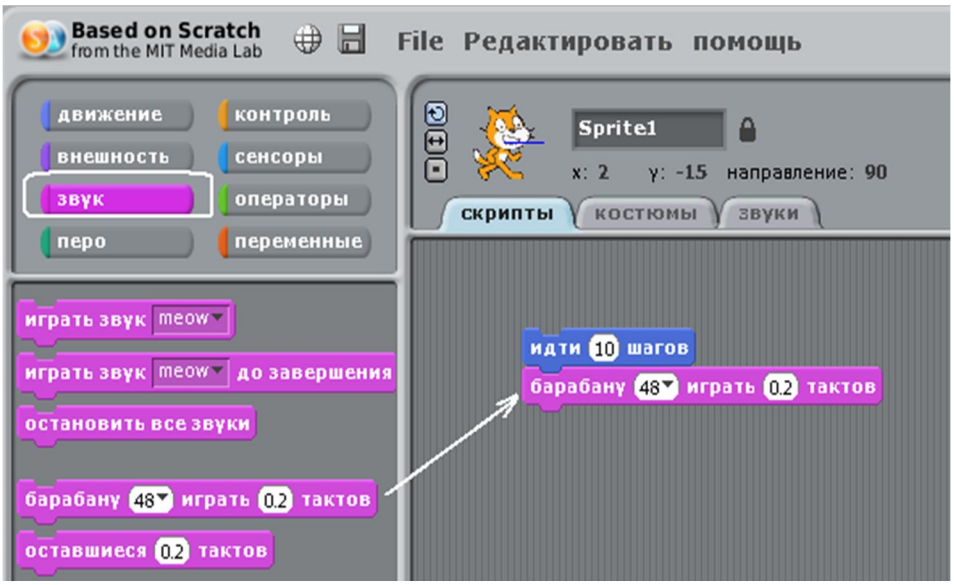


Рис. 6.8. Добавление нужных действий

Повторив двойной щелчок по любому блоку, мы услышим звук барабана. Из меню, которое можно найти, нажав на стрелочку рядом с цифрой 48, можно выбрать другой звук, например, под номером 49.

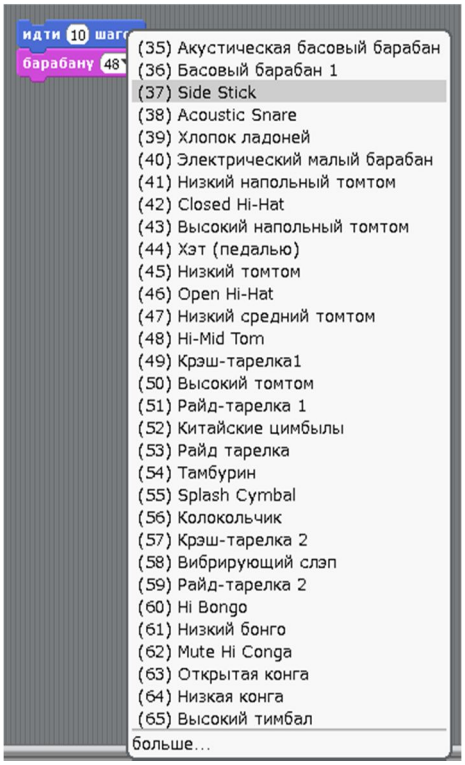


Рис. 6.9. Выбор звука барабана

Добавим к программе ещё один элемент «идти». Теперь, выделив цифру 10 внутри блока, заменим её на -10. И добавим второй блок «играть барабану», заменив барабан. В итоге получим следующую программу.

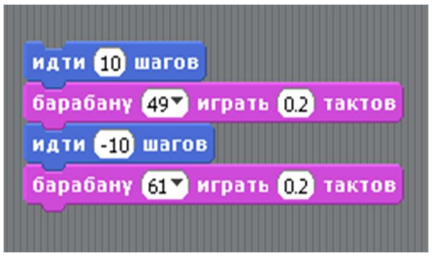


Рис. 6.10. Продолжение построения программы

Двойной щелчок по любому блоку выполнит собранные команды. Осталось обернуть это в цикл.

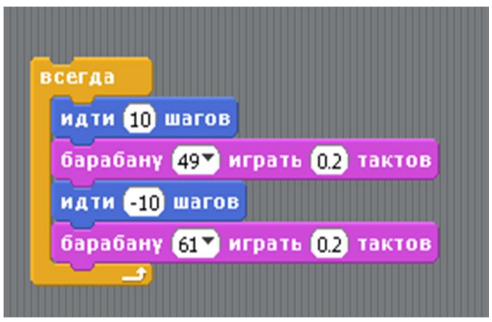


Рис. 6.11. «Зацикливание» программы

Можно добавить смену костюмов (вида) из раздела «внешность».

Запустите программу двойным щелчком по любому объекту и получите некоторое удовольствие от проделок этого котика.

О, я и не заметил, увлёкся, не заметил, как почти всё пересказал. Но почти, ещё не всё. Остальное вы сами...

Используя язык Scratch можно научиться программировать. И учиться с удовольствием, просматривая результаты работы сразу после создания программы. Однако вернёмся к модулю Arduino. Создавая свою программу, можно ошибиться. Я сейчас приведу такой пример. Вернёмся к схеме с кнопкой и светодиодом.

Мы можем рассуждать так: светодиод включается, если кнопка нажата, и выключен без нажатой кнопки. И собрать схему так, как показано ниже.

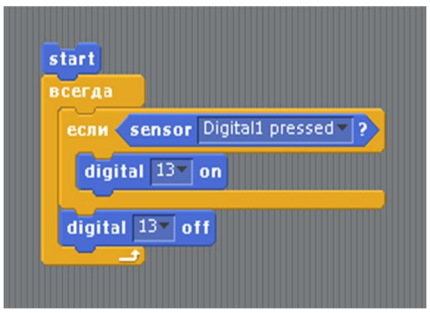


Рис. 6.12. Один из вариантов программы

Запустите программу, нажмите кнопку и посмотрите на модуль. Светодиод не горит. В чём проблема? Что мы сделали не так, как следует? Программа S4A поможет разобраться в этом. Во-первых, запустив пошаговое выполнение, мы можем убедиться, что при нажатой кнопке мы выполняем команду включения (напомню, раздел «Редактировать» пункт «Начать пошаговое выполнение»).

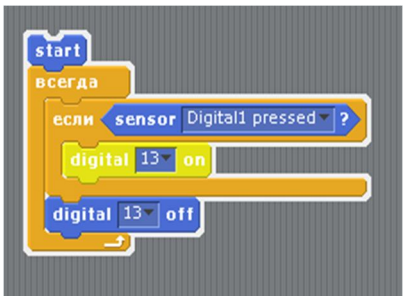


Рис. 6.13. Пошаговая проверка программы

Добавим паузы, это поможет разобраться, отчего мы не видим включения.

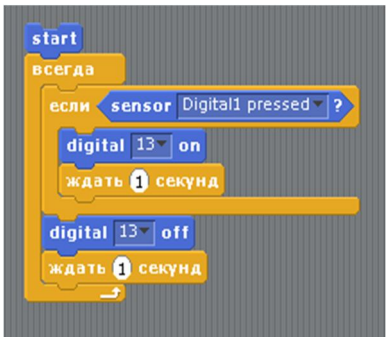


Рис. 6.14. Изменение программы для отладочных целей

Вот теперь мы видим светодиод включённым. Когда кнопка нажата, светодиод мигает с интервалом раз в секунду. А, значит, в предыдущей версии он мигает так быстро, что мы не видим этого.

То, что мы сейчас проделали, называется отладкой программы. И S4A помогает это сделать

быстро, легко и понятно.

Посмотрите, как изменится поведение светодиода, если изменить программу следующим образом.

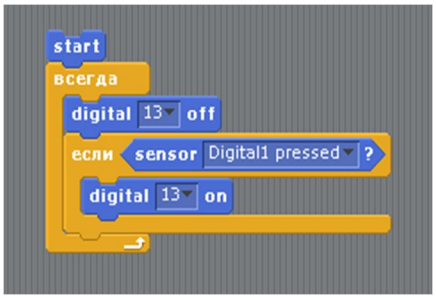


Рис. 6.15. Модификация программы

Мы подчас забываем, что процессор модуля Arduino работает очень и очень быстро. И забываем, что одни команды выполняются быстро, а другие команды, работаем ли мы с языком Scratch или Си, для процессора разбиваются на множество команд, требующих на своё выполнение времени. Отладка программы позволяет разобраться с возникающими проблемами.

Прежде, чем продолжить рассказ об отладке программ, ещё одно маленькое замечание. Хотя о состоянии цифровых входов можно узнать из сводной таблицы (рядом с изображением модуля), можно, если поставить галочку рядом с названием программного элемента (как отмечено на рисунке ниже), появится отдельное информационное окно, относящееся к цифровому входу.

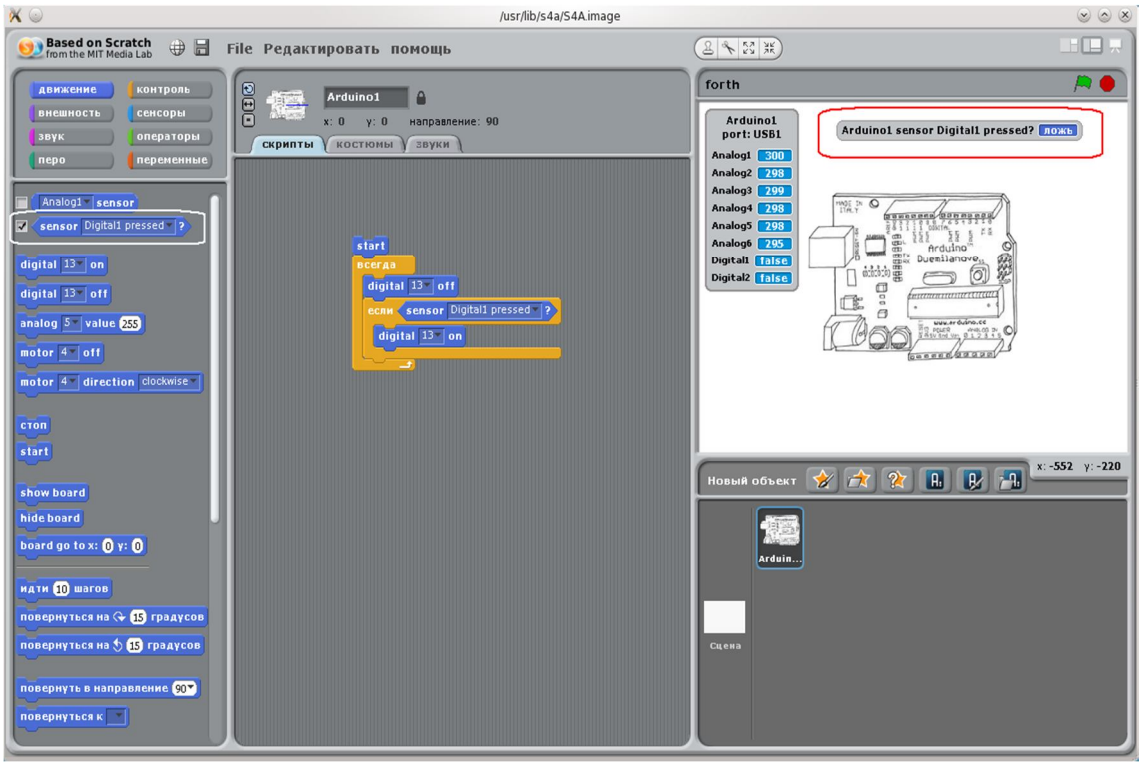


Рис. 6.16. Вынос состояния цифрового входа

Кроме программы S4A есть ещё одна, помогающая отлаживать программу, написанную для модуля Arduino. К сожалению, её версия есть только для Windows.

Глава 7. Отладка программы на виртуальной плате

Любую программу следует отлаживать. Поэтому удобство работы со средой разработки программ во многом обусловлено встроенным в неё отладчиком (debugger). Чем мощнее отладчик, тем легче проверить работу программы, и особенно это относится к программированию микроконтроллеров. О том, как можно отлаживать программу в среде разработки Arduino мы поговорим в следующих главах. О том, как научиться программировать и проверить работу программы в среде Scratch for Arduino, мы обсудили в предыдущей главе. А сейчас обратимся к ещё одной программе, которая называется VirtualBreadboard. Её можно найти на сайте проекта: http://visualbreadboard.com.

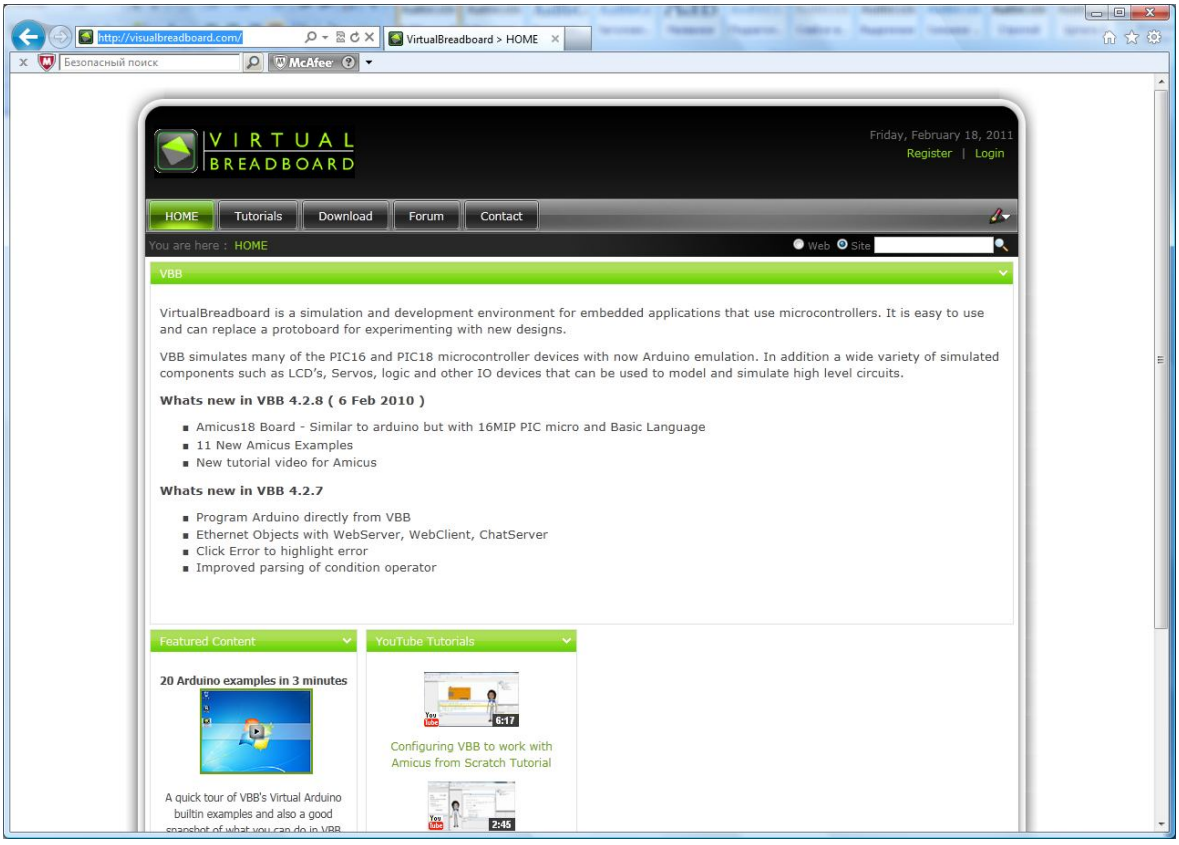


Рис. 7.1. Сайт программы виртуальной макетной платы

На первой странице вы найдёте много примеров работы программы в виде видеоуроков. Перейдя на страницу Download, вы можете загрузить программу. К сожалению, на момент написания этой главы версии для Linux нет. И попытка запустить программу в Linux под Wine закончилась неудачей.

Чуть позже мы поговорим о том, как установить программу, а сейчас несколько примеров из набора, предлагаемого автором. После запуска программы появляется диалоговое окно выбора с несколькими закладками: использовать примеры (открыта), использовать существующий проект или обратиться к тем, с которыми работали недавно.

Рядом с разделом Arduino есть плюсик, щёлкнув по которому, вы можете открыть примеры, созданные автором проекта.

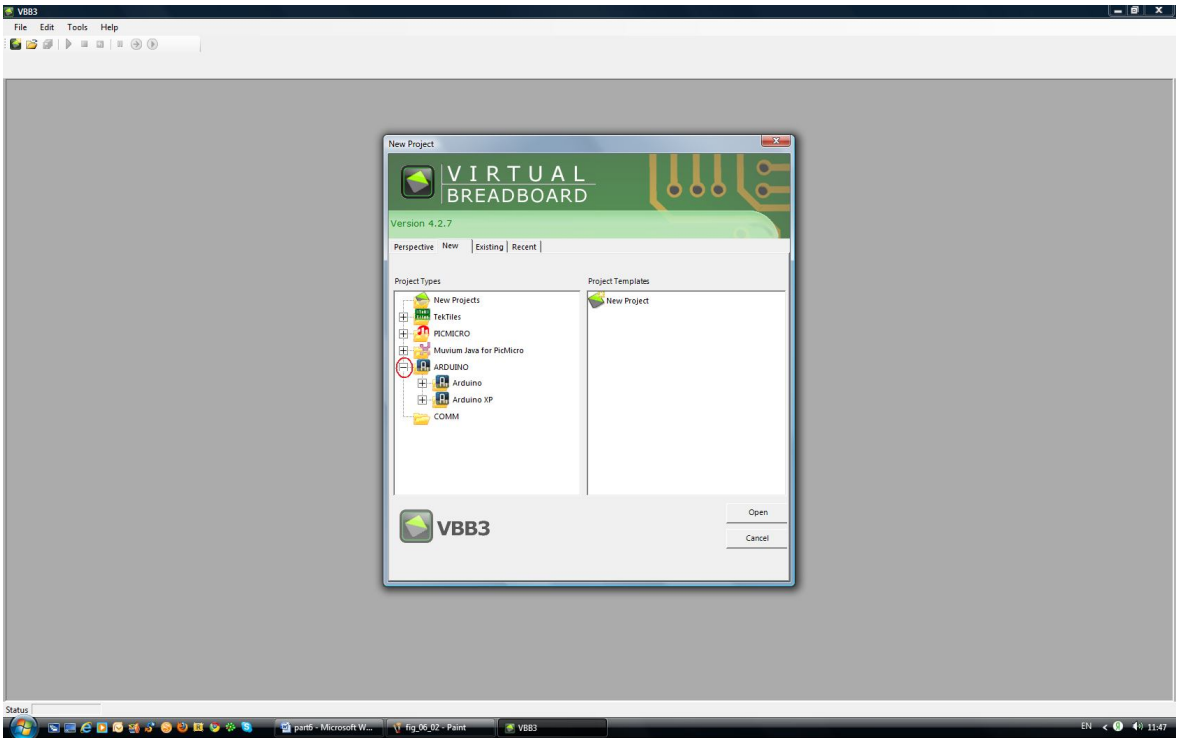


Рис. 7.2. Начальный диалог программы

Перемещаясь по дереву примеров, можно выбрать разные проекты, как, например, такой интересный проект: после запуска программы, если перемещать движок потенциометра, мигание светодиода меняет свой характер.

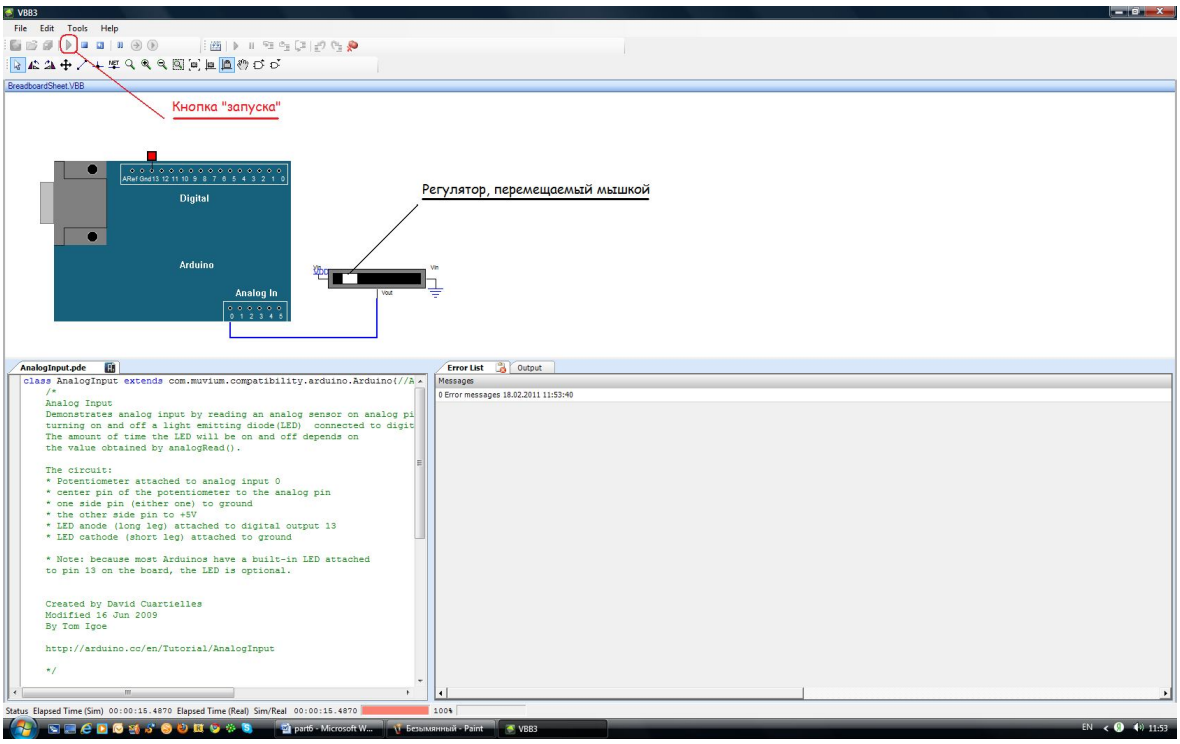


Рис. 7.3. Один из примеров работы программы

Или такой проект.

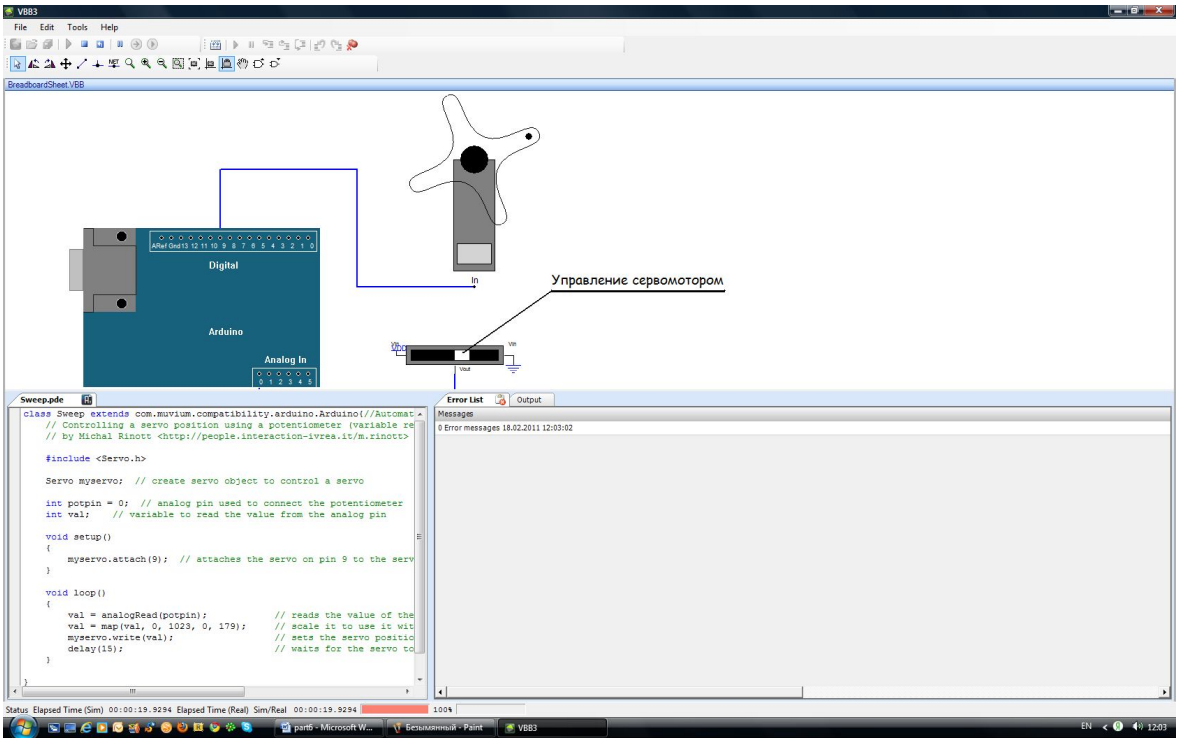


Рис. 7.4. Ещё один проект из набора примеров

Перемещая движок потенциометра, вы управляете поворотом сервопривода. Словом, первое, что я советую сделать после запуска программы – посмотреть приведённые для модуля Arduino примеры.

Теперь об установке программы. Собственно, сама программа не требует установки – после скачивания её можно разместить в удобном месте и запускать. На сегодняшний день есть версия VBB 4.2.8, но я советую сегодня использовать версию VBB 4.2.7.

Для использования программы следует дополнительно загрузить некоторые вспомогательные программы, перечень которых есть на странице загрузки. Вот, что сказано об этом:

…программа может располагаться на рабочем столе или в другом удобном месте.
Однако VirtualBreadboard имеет следующие зависимости, которые нужно дополнить установкой на вашем компьютере. Они уже могут быть установлены на вашем компьютере, но могут потребовать установки до VBB.

- Latest version of DirectX. Click here to download from Microsoft
- .Net 2.0 Redistributable. Click here to download from Microsoft
- J# 2.0 Redistributable. Click here to download from Microsoft
- Java JRE 1.6 (version 6). Click here to download from Oracle

Выделенное мною указание «Click here» – это ссылка на загрузку. Щёлкнув по ней, вы попадёте в нужное место, где можно найти всё необходимое для загрузки.

Посмотрев видеоролик на сайте проекта, я создал папку с именем arduino в своей директории, куда и отправил программу. Забегая вперёд, в ту же папку я распаковал и программу Arduino. И, опять таки на сегодняшний день, я советую, хотя использовал версию программы Arduino 0022, для совместной работы применить более раннюю версию Arduino 0018.

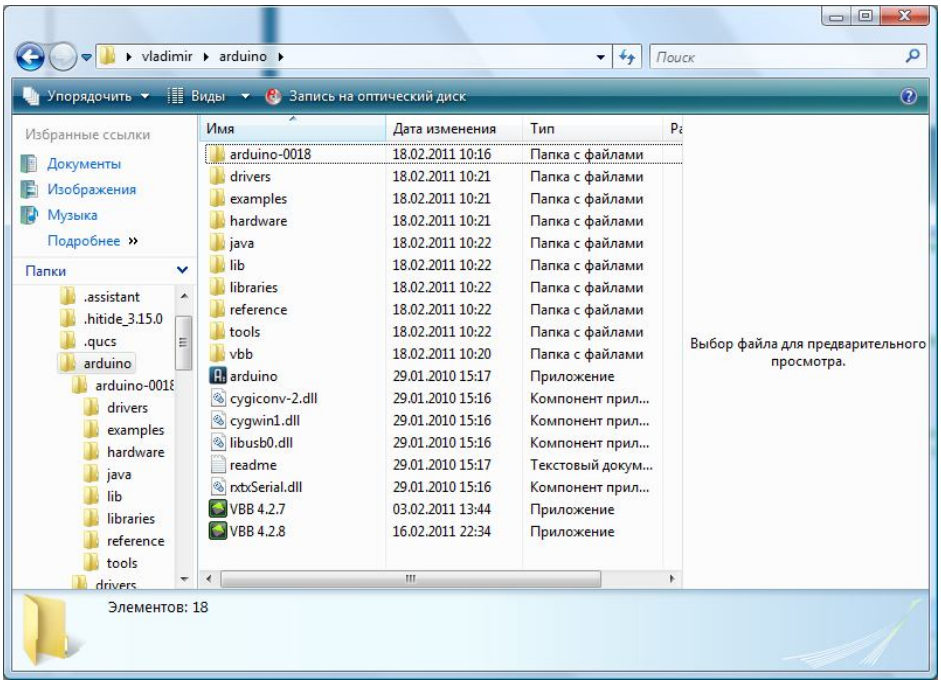


Рис. 7.5. Выбор версии Arduino для работы с программой

Сочетание, например, vbb-4.2.8 и arduino-0022 мне в операционной системе Vista SP2 заставить работать должным образом не удалось. Это относится к использованию возможности программы VirtualBreadboard программировать модуль Arduino непосредственно, без перехода в программу Arduino. В остальном же, видимо, не возникнет проблем и при использовании последней на сегодня версии.

Проделав все загрузки программ, собрав всё нужное в удобном вам месте, вы можете столкнуться с ещё одной проблемой при попытке загрузить программу в модуль Arduino.

Кстати, ещё одна функция программы весьма полезная. К полезным функциям я отнёс бы и возможность работать с разными модулями, и возможность работать с PIC-контроллерами. Я не советую начинающим «прыгать» от модуля к модулю, от контроллера одного производителя к контроллеру другого. Но это будет полезно, когда вы освоитесь с первыми (и последующими) шагами и захотите создавать свои конструкции, разрабатывать свои программы.

Вернёмся к установке и работе с программой VirtualBreadboard, которую я считаю очень полезной, особенно для начинающих. Потратив некоторое время на её установку, вы не пожалеете. Итак.

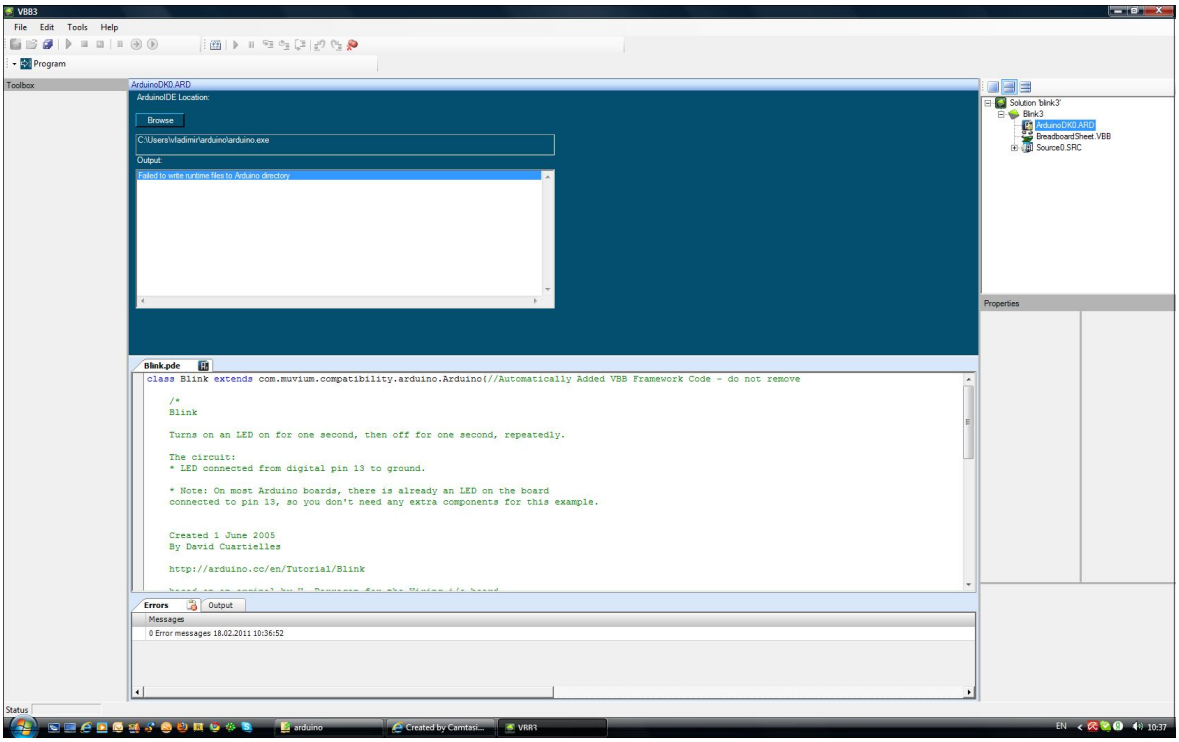


Рис. 7.6. Сообщение о проблемах в работе программы

Программа сообщает, что не может записать нужные ей файлы в директорию Arduino. Не могу сказать, столкнётесь ли вы с этой проблемой в Windows XP, некоторые считают, что можете столкнуться, но в Windows Vista я столкнулся (видимо, и в Windows 7 это будет иметь место). Операционная система защищена от несанкционированного изменения файлов. В данном случае работает эта защита. Её можно отключить, я пробовал, но потом долго и не без труда включал эту защиту – весь процесс получился столь долгим, что у меня, просто, не хватало терпения дождаться, когда оживёт операционная система, которая на мою просьбу перейти в диалог включения защиты задумалась…

Есть ещё одна причина, по которой я не советую снимать защиту – ослаблять защиту Windows, это, знаете, себе дороже. Я сейчас использую бесплатную версию антивирусной программы Avira. В последнее время её усовершенствовали, она проверяет компьютер до подключения его к сети, доступ к Интернету закрыт, и длится это долго. Я, конечно, злюсь, но вспоминаю, как много лет назад, переустанавливая Windows, я не поставил антивирусную программу. Решив, что вначале я поменяю разбивку жесткого диска, я запустил программу… В то время был такой вирус, который самопроизвольно перезагружал компьютер. В результате жёсткий диск оказался испорчен, и я с большим трудом восстановил только некоторые, самые необходимые для работы, файлы. Злись, не злись, но лучше не снимать защиту с компьютера. Может быть, хотя и не факт, по этой причине я в основном работаю в Linux, обращаясь к Windows в случаях, подобных сегодняшней необходимости описать работу с VirtualBreadboard.

Но, что же делать? Перед запуском (или получив такое сообщение, когда вы нажали клавишу «Program» в верней части окна) следует проделать несложную операцию: в проводнике перейти к папке, которую вы создали и назвали arduino; щёлкнуть по ней правой клавишей мышки…

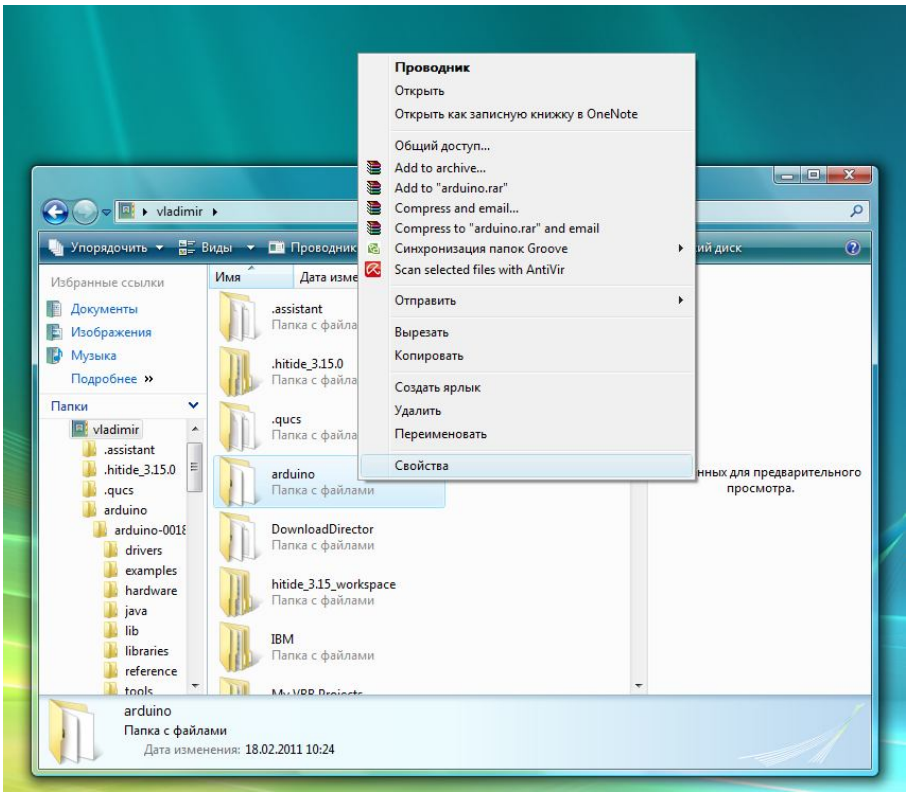


Рис. 7.7. Доступ к свойствам папки

В открывающемся меню выберите раздел «Свойства». Открывается диалоговое окно свойств папки.

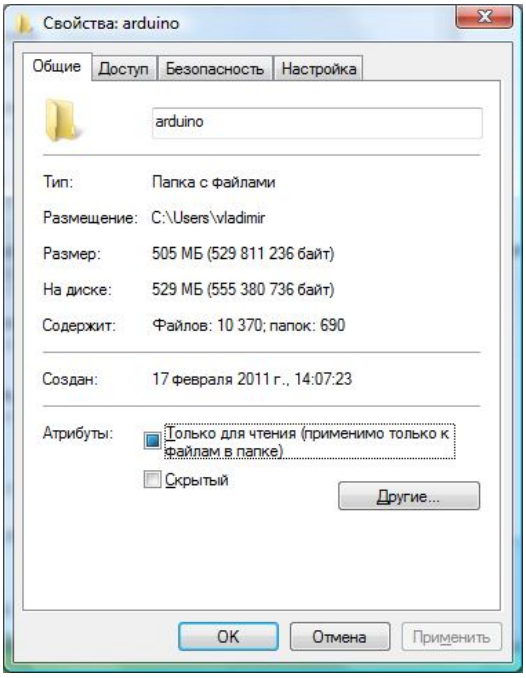


Рис. 7.8. Изменение свойств папки

Как вы видите, папка «Только для чтения». Такое свойство помогает защитить программы и файлы от несанкционированного изменения. Но нам нужно снять эту опцию – щёлкнуть левой клавишей мышки по синему квадратику рядом с надписью. Откроется следующее окно диалога.

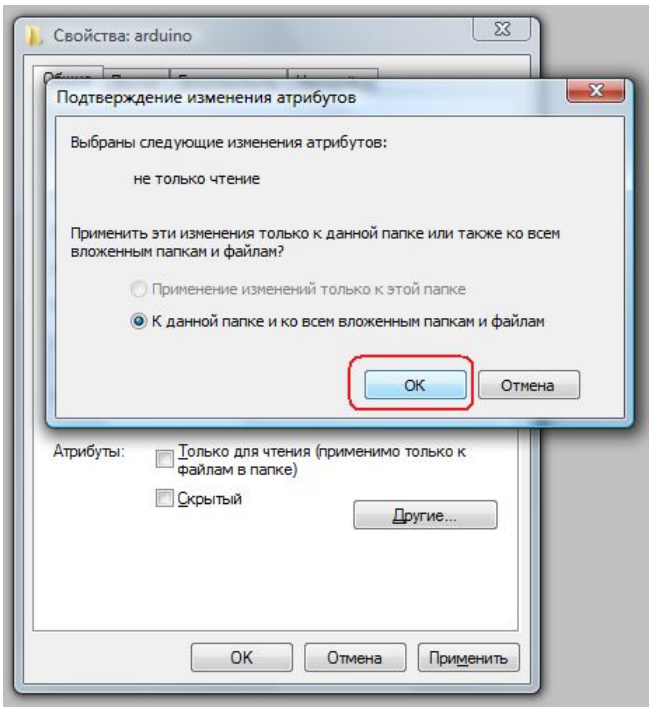


Рис. 7.9. Подтверждение изменения свойств

Достаточно нажать выделенную кнопку «ОК», что закрывает диалог, нажать кнопку «Применить» в следующем окне, дождаться пока завершится процесс разблокирования файлов, и нажать кнопку «ОК», когда кнопка «Применить» перестанет быть активна.

Теперь, в программе VirtualBreadboard клавиша «Program» запускает загрузку модуля, о чём можно судить и по активному миганию светодиодов, работающих с COM-портом (виртуальным) компьютера.

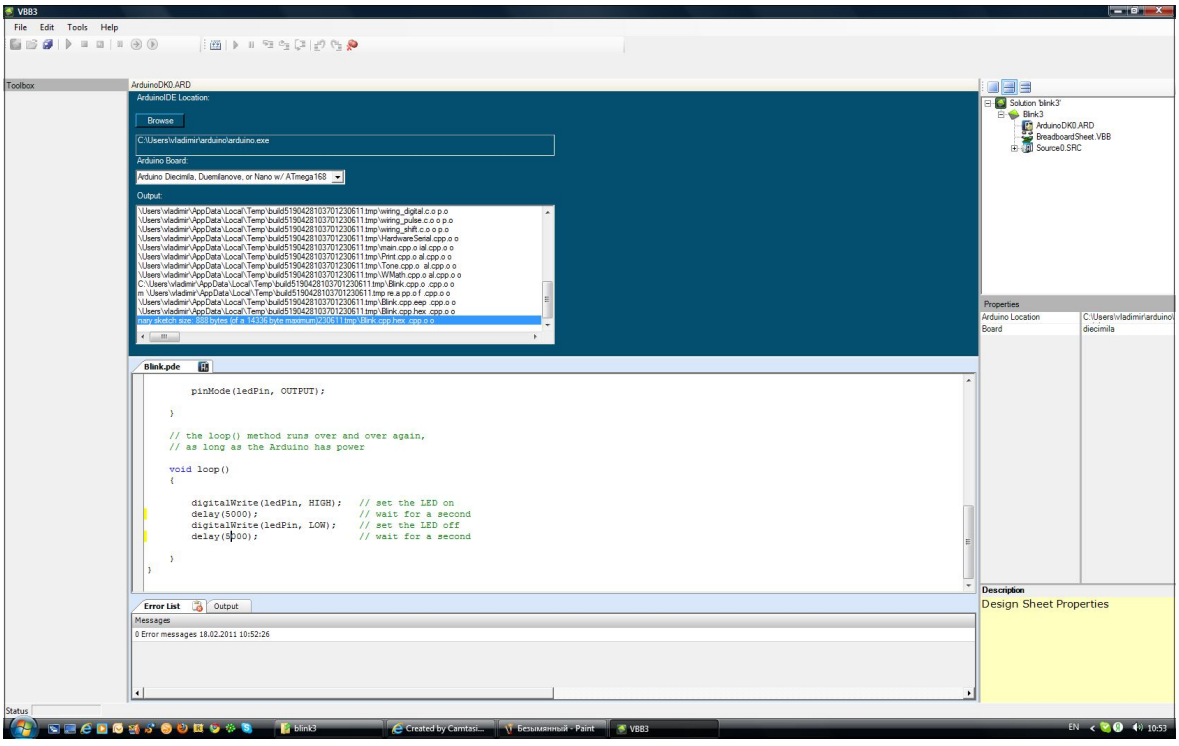


Рис. 7.10. Правильная работа программы

Если вы последовали моему совету, то, как проверить, что всё у вас получилось?

Я не буду ничего придумывать, я перескажу только то, что увидел на видеоролике, размещённом на сайте проекта.

Как советует автор, запускаем программу и выбираем нужный пример.



Рис. 7.11. Выбор одного из примеров в начальном диалоге

Открываем его, нажав на кнопку «Open». Мы можем проверить работу программы, запустив моделирование кнопкой на инструментальной панели (отмечена на рисунке ниже).



Рис. 7.12. Основное меню и инструментальные панели программы

Убедившись, что светодиод мигает с частотой раз в секунду, мы должны сохранить проект. Для этого используем в разделе основного меню «File» пункт «Save as…, сохранить как».

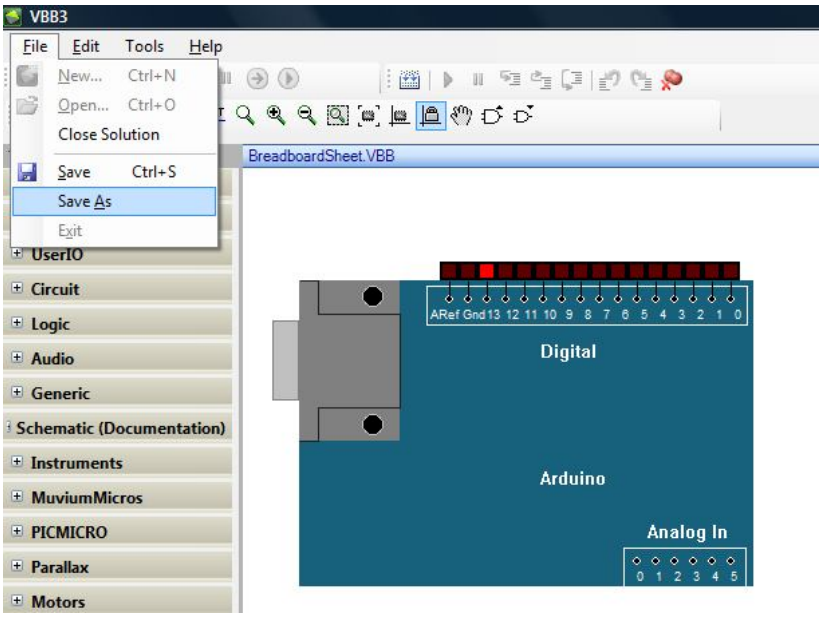


Рис. 7.13. Содержание раздела File основного меню

При этом открывается диалоговое окно, в котором обратите внимание на выделенный мною фрагмент.

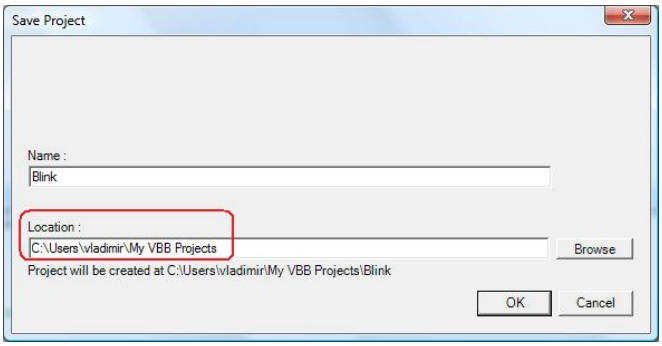


Рис. 7.14. Создание папки проектов при первом сохранении

Этой папки пока у вас нет. Она будет создана, когда вы нажмёте кнопку «ОК».

Теперь обратимся к правой части рабочего окна программы. Туда, где отображаются все компоненты проекта.

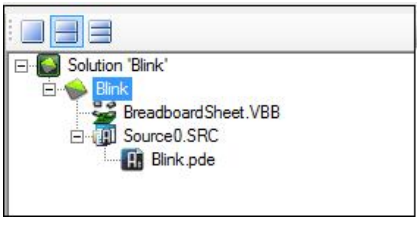


Рис. 7.15. Окно менеджера проекта

Щёлкнув по выделенной папке проекта правой клавишей мышки, выбираем из выпадающего меню вначале пункт «Code Generators», что открывает подменю, затем в нём пункт «Add New Arduino Code Generator, добавить новый Arduino генератор кода».

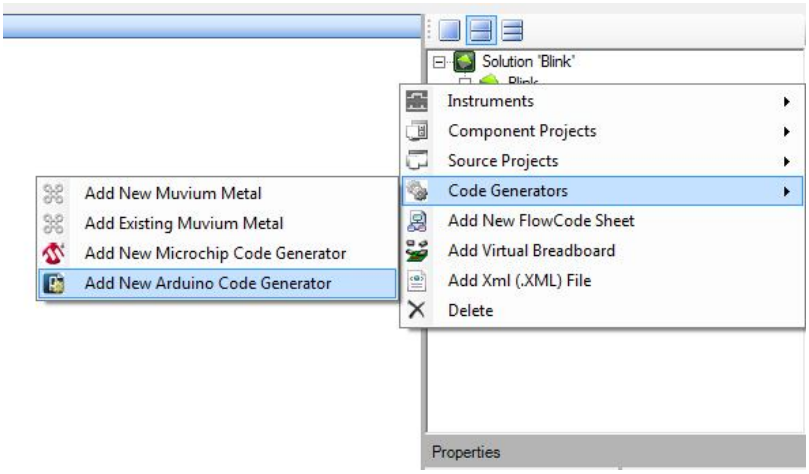


Рис. 7.16. Выбор генератора кода

После этого «дерево» проекта несколько изменится.



Рис. 7.17. Изменение вида дерева проекта

Подцепите этот появившийся элемент мышкой и перенесите его к верхней кромке окна с рисунком модуля. Там его и «сбросьте».

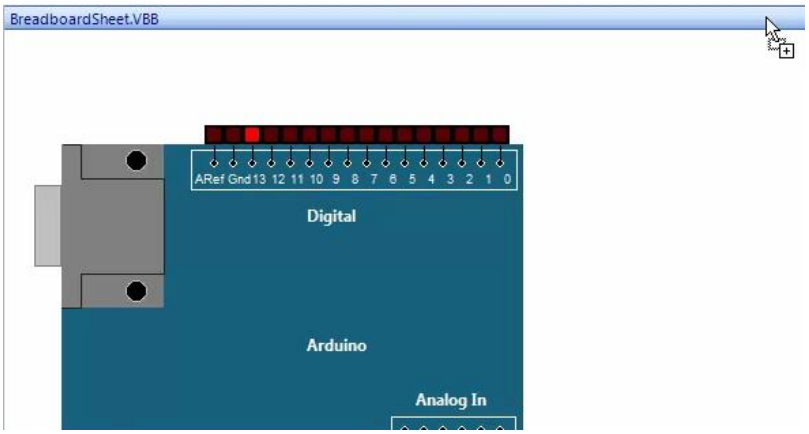


Рис. 7.18. Перенос «генератора кода» в рабочее окно

Вид рабочего окна изменится. А, если щёлкнуть по этому элементу в дереве проекта дважды левой клавишей мышки, то ниже появится окно свойств. Следующие шаги, которые нам предстоит выполнить, это указать место расположения программы Arduino и выбрать свою модель модуля Arduino. Для указания места расположения служит кнопка «Browse», а модуль выбирается из выпадающего списка, когда вы нажмёте кнопку со стрелкой рядом с окошком модели, озаглавленным «Arduino Board:». Как видно из списка, программа работает со многими модулями.

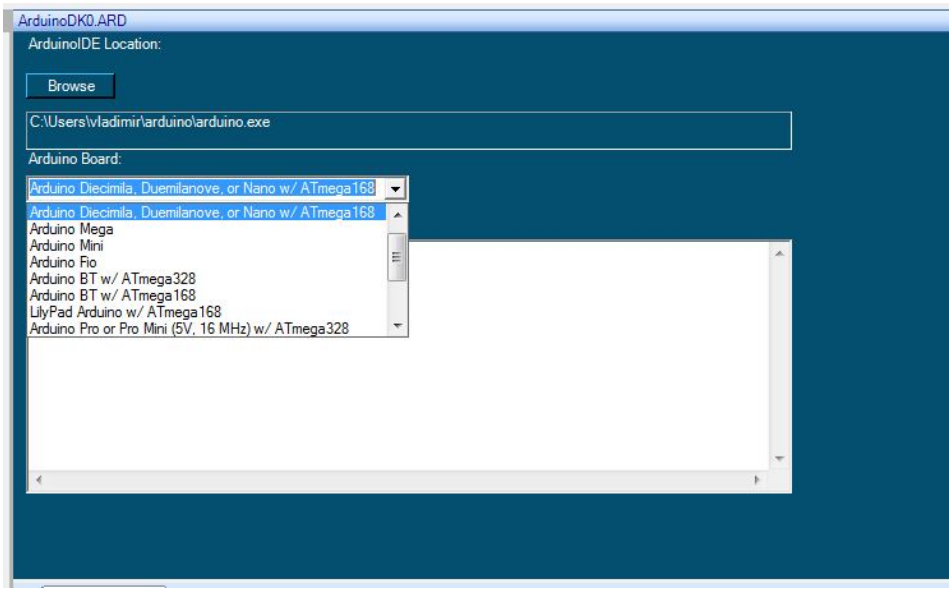


Рис. 7.19. Выбор модели модуля Arduino

Далее двойным щелчком в дереве проекта по компоненту Source0.SRC мы откроем его свойства (в окошке ниже), где должны выбрать генератор кода (отмечены на рисунке).

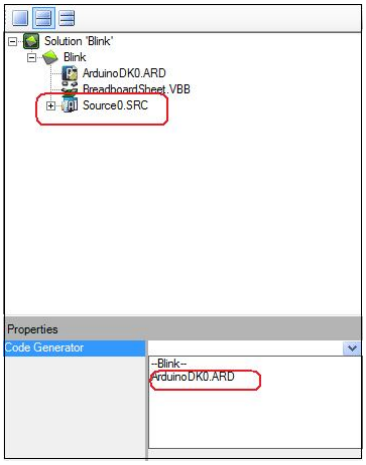


Рис. 7.20. Настройка исходного кода

После выбора в окне свойств рядом с «Code Generator» появится нужный нам вариант.

Следующая процедура – трансляция исходного кода. Проще всего это сделать с помощью отмеченной на рисунке кнопки на инструментальной панели.

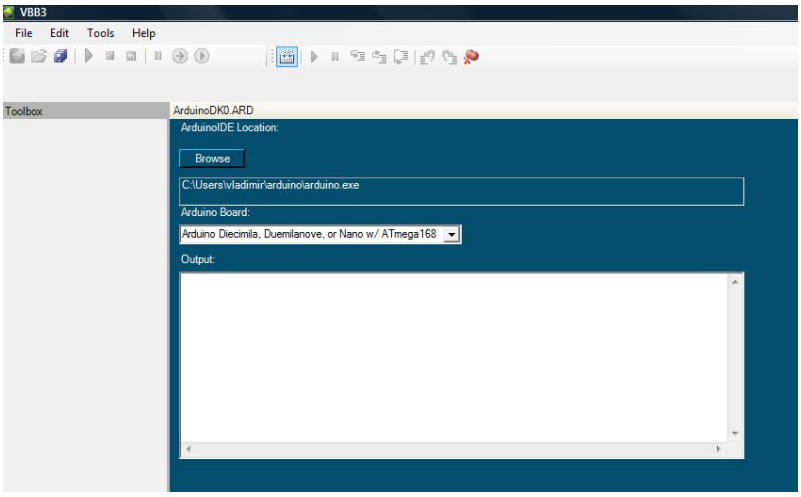


Рис. 7.21. Окончание настроек генератора кода

Остаётся указать порт для связи с модулем…

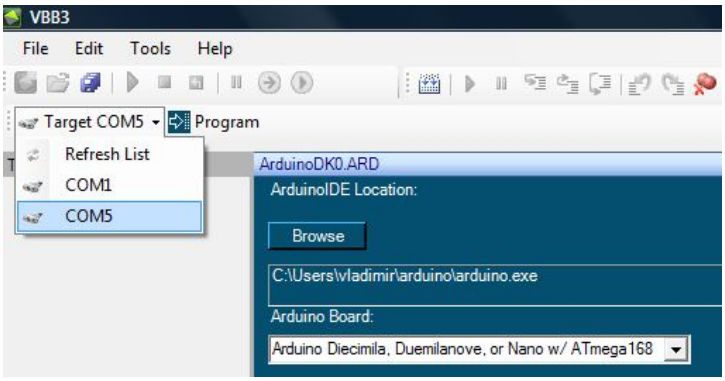


Рис. 7.22. Порт связи компьютера с модулем Arduino

…и можно нажать, наконец, кнопку «Program».

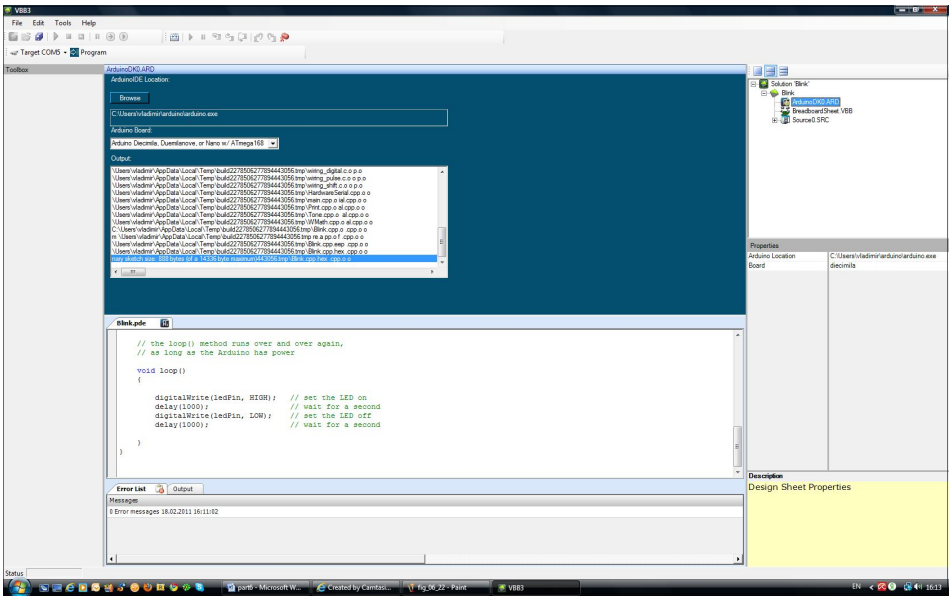
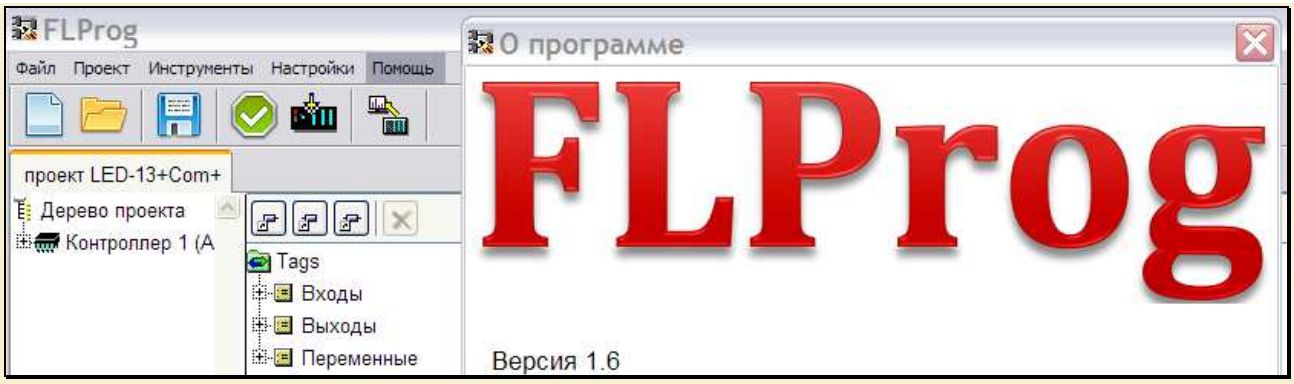


Рис. 7.23. Загрузка программы в модуль из VBB

Мой модуль Arduino, надеюсь, что и ваш, начинает весело мигать всеми своими светодиодами.

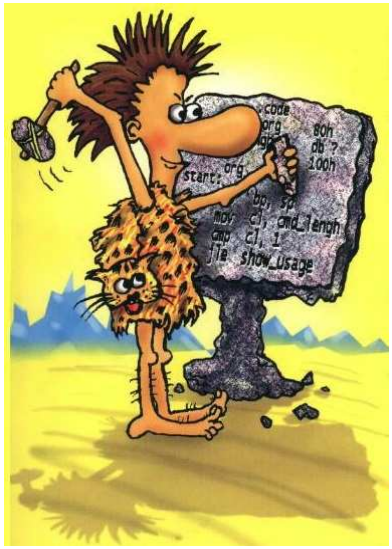


FLProg — система визуального программирования плат Arduino

Сергей Глушенко, http://flprog.ru/ support@flprog.ru

В настоящее время в мире начался бум по использованию микроконтроллеров в различных самоделках и стартапах. Действительно, цены на микроконтроллеры упали, а возможности их постоянно растут. Да и наши друзья, китайцы, научились изготавливать периферию к ним, и продают её к тому же по смешным ценам. Но вот с программированием микроконтроллеров всё не так радужно...

С чего всё началось и как развивалось





С самого момента появления микропроцессоров развитие принципов работы с ними идет по пути роста абстракции. Первый этап представлял программирование непосредственно в машинных кодах. Программирование было сложным, долгим и требовало очень специфичного склада ума. Поэтому программистов было очень мало.

Но человек существо ленивое, а лень, как известно, двигатель прогресса. Придумали первый уровень абстракции — ассемблер. Писать программы стало проще и веселее. Количество программистов возросло. Но все равно ассемблер не очень сильно отличался от машинных кодов.

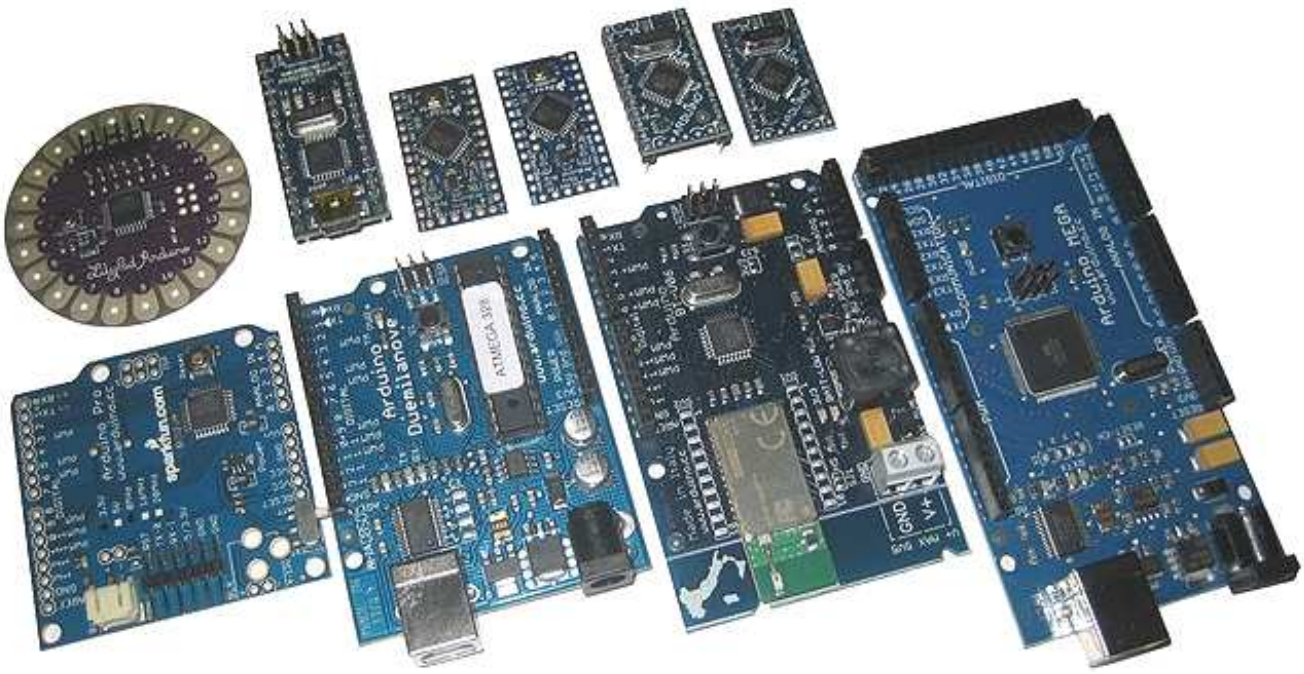
Поэтому появился следующий уровень абстракции. Языки высокого уровня. Основной целью этих языков была возможность объяснить машине, что от нее хотят, на языке максимально приближенном к человеческому. Это позволяло заниматься программированием людям с менее специфичным складом ума. Поэтому с развитием языков высокого уровня количество программистов росло, и соответственно росло количество полезных программ, которые они создавали.



Как дела обстоят сейчас

Конечно, для начала работы непосредственно с контроллером требуется определенная подготовка. То есть, необходимы программатор, настроенная среда для программирования на компьютере, ну и, естественно, знание языка программирования. Кроме того, требуется умение в работе с паяльником, разработке печатных плат, знания в электротехнике и электронике. Так что порог вхождения в область создания собственных устройств на микроконтроллерах остается высоким.

Кроме того, для такой работы требуется сочетание навыков, которые достаточно редко встречаются вместе. Программисты редко дружат с паяльником, а электронщики не часто являются программистами. Для программистов проблему решили созданием платы Arduino, которая позволяет собирать устройства без использования инструментов.

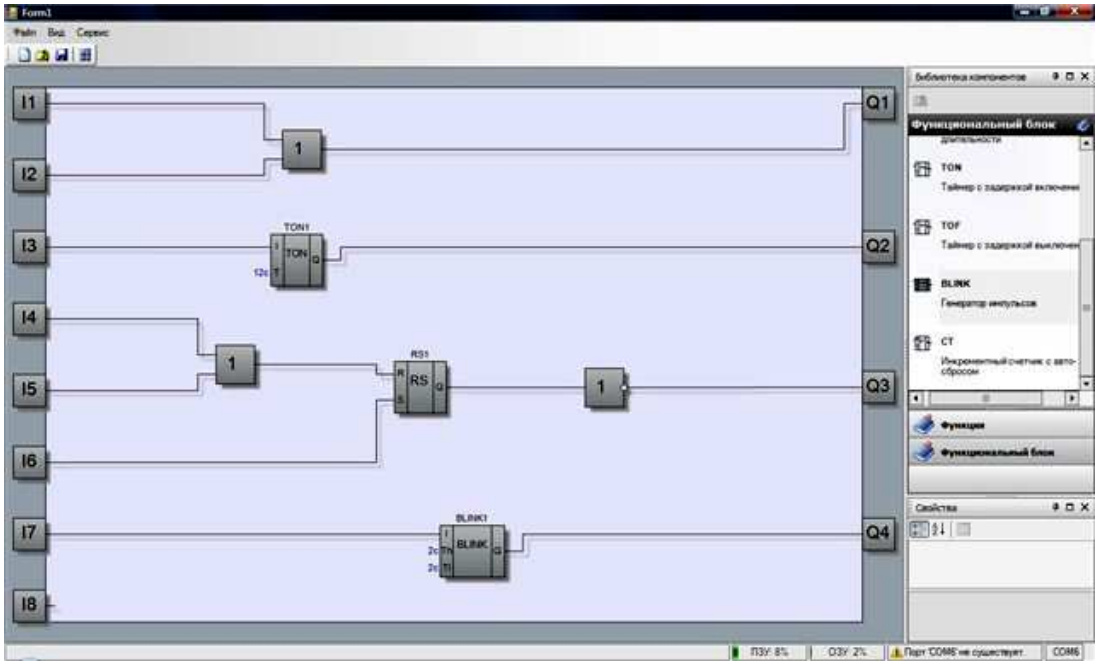


Для электронщиков и электриков все хуже. До последнего времени для того, чтобы создать свое устройство с применением микроконтроллера, у них было два пути. Либо самим изучать язык программирования "С", либо просить помощи у программиста. Оба пути не самые лучшие. Для того что бы стать программистом, необходим определенный склад ума, не всегда совместимый с опытом чтения электрических схем. А знакомого программиста может не оказаться под рукой.

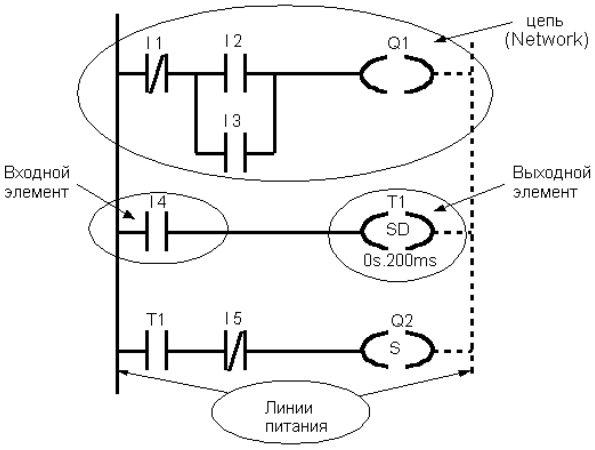
В то же время давно существуют среды программирования адаптированные под обычного инженера – электронщика, ну или просто электрика. Я имею в виду среды программирования промышленных контроллеров – ПЛК. Они позволяют создавать программное обеспечение для контроллеров на языках ***FBD*** и ***LAD***. Собственно говоря, как таковыми языками они не являются. Это, скорее, графические среды для рисования принципиальных или логических схем.

FBD (Function Block Diagram)

— графический язык программирования стандарта МЭК 61131-3. Программа образуется из списка цепей, выполняемых последовательно сверху вниз. При программировании используются наборы библиотечных блоков. Блок (элемент) — это подпрограмма, функция или функциональный блок (И, ИЛИ, НЕ, триггеры, таймеры, счётчики, блоки обработки аналогового сигнала, математические операции и др.). Каждая отдельная цепь представляет собой выражение, составленное графически из отдельных элементов. К выходу блока подключается следующий блок, образуя цепь. Внутри цепи блоки выполняются строго в порядке их соединения. Результат вычисления цепи записывается во внутреннюю переменную либо подается на выход контроллера.



Ladder Diagram (LD, LAD, РКС)



— язык релейной (лестничной) логики. Синтаксис языка удобен для замены логических схем, выполненных на релейной технике. Язык ориентирован на инженеров по автоматизации, работающих на промышленных предприятиях. Обеспечивает наглядный интерфейс логики работы контроллера, облегчающий не только задачи собственно программирования и ввода в эксплуатацию, но и быстрый поиск неполадок в подключаемом к контроллеру оборудовании. Программа на языке релейной логики имеет наглядный и интуитивно понятный инженерам-электрикам графический интерфейс, представляющий логические операции, как электрическую цепь с замкнутыми и разомкнутыми контактами. Протекание или отсутствие тока в этой цепи соответствует результату логической операции (истина — если ток течет; ложь — если ток не течет). Основными элементами языка являются контакты, которые можно образно уподобить паре контактов реле или кнопки. Пара контактов отождествляется с логической переменной, а состояние этой пары — со значением переменной. Различаются нормально замкнутые и нормально разомкнутые контактные элементы, которые можно сопоставить с нормально замкнутыми и нормально разомкнутыми кнопками в электрических цепях.

Такой подход оказался очень удобным для легкого вхождения в разработку систем АСУ инженеров-электриков и электронщиков. Разрабатывая проекты установок, они могли легко привязать работу этих установок к алгоритмам работы контроллера. В обслуживании этих установок на объекте также лучше, когда существующий обслуживающий персонал может легко проверить работу системы АСУ, найти проблему. И при этом нет необходимости вызывать по каждому пустяку программиста из «Центра». И это подход себя оправдал. На сегодняшний день почти все системы промышленной автоматики созданы с помощью таких средств разработки.



Такая среда разработки есть у Siemens, ABB, Schneider Electric... да и практически у всех производителей ПЛК. Казалось бы, идеальное решение для любителей самоделок. Но, как всегда есть «но». Все эти среды программирования привязаны к промышленным контроллерам определённого производителя. И цены на эти контроллера мало вдохновляют. Очень редко какой семейный бюджет позволит приобрести контроллер ценой в несколько десятков тысяч рублей.



Центральный процессор Siemens S7-1200: 6AG12121BD305XB0

SIPLUS S7-1200 CPU 1212C AC/DC/RLY -25... +55 градусов цельсия, с комформным покрытием, возможно использование сигнальных плат, на основе 6ES7212-1BD30-0XB0, компактное ЦПУ, AC/DC/RLY, встроенные вход...

19321,52 руб.

Добавить в корзину

Зато платы Arduino идеально подходят для самодельщиков и кулибиных, на которых наша страна всегда была, есть и будет богата. Но, опять «но». Программируются эти платы на языке С. Для большинства этих умнейших людей, с очень прямыми руками, растущими из положенного места, язык С – это китайская азбука. Они могут придумать, нарисовать, собрать, отладить и запустить сложнейшие схемы, но `If`, `For`, `Case`, `Void` и т.п. - это не для них. Конечно, можно почитать инструкции в интернете, поиграться какое-то время, помигать светодиодом с помощью примера. Но для более серьезного применения необходимо детальное изучение языка. А зачем им это?

Они не собираются быть профессиональными программистами. У них другой путь. Они что-то придумали. Да, это проще и красивее собрать с помощью микроконтроллера, но становится для этого программистом, потратив месяцы на изучение языка? Нет, конечно. Собирают по старинке, попроще, конечно, но в своей области.

Где выход?



```
if (photoState == HIGH){
      delay(1);   // ждём 1 мС
      photoState = digitalRead(photoPin);
      if (photoState == LOW){
            delay(1);   // ждём 1 мС
            photoState = digitalRead(photoPin);
            if (photoState == HIGH){
                  delay(1);   // ждём 1 мС
                  photoState = digitalRead(photoPin);
                  if (photoState == LOW){
                        delayMicroseconds(3500);      // ждём 3.5 мС
                        photoState = digitalRead(photoPin);
                        if (photoState == HIGH){
                              // выключаем красный
                        digitalWrite(ledPinRed, LOW);
                              //включаем зелёный
                        digitalWrite(ledPinGreen, HIGH);
                        delay(2000); // зелёный 2 секунды
                              //выключаем зелёный
                        digitalWrite(ledPinGreen, LOW);
                        }
                  }
            }
      }

}
}// иначе ничего не делаем
}
```

На основании всех этих выкладок и был создан проект **FLProg**. Основная идея проекта – совместить принципы промышленного программирования с дешевизной и удобством Arduino. Проект предлагает новый уровень абстракции с довольно смелым заявлением —

Чтобы программировать микроконтроллеры

не обязательно знать языки программирования!

В результате получился инструмент, позволяющий создавать свои проекты на Arduino любому человеку, знакомому с электротехникой и электроникой, позволяющий создать свое изделие с использованием данных плат.

Проект состоит из двух частей.

Первая часть - это десктоп-приложение **FLProg**, представляющее собой графическую среду программирования плат Arduino.

Во-вторых, это сайт **FLProg.ru**, с помощью которого члены сообщества пользователей программы могут пообщаться между собой, узнать последние новости проекта, скачать последнюю версию программы, ну и найти необходимую информацию по работе с приложением.

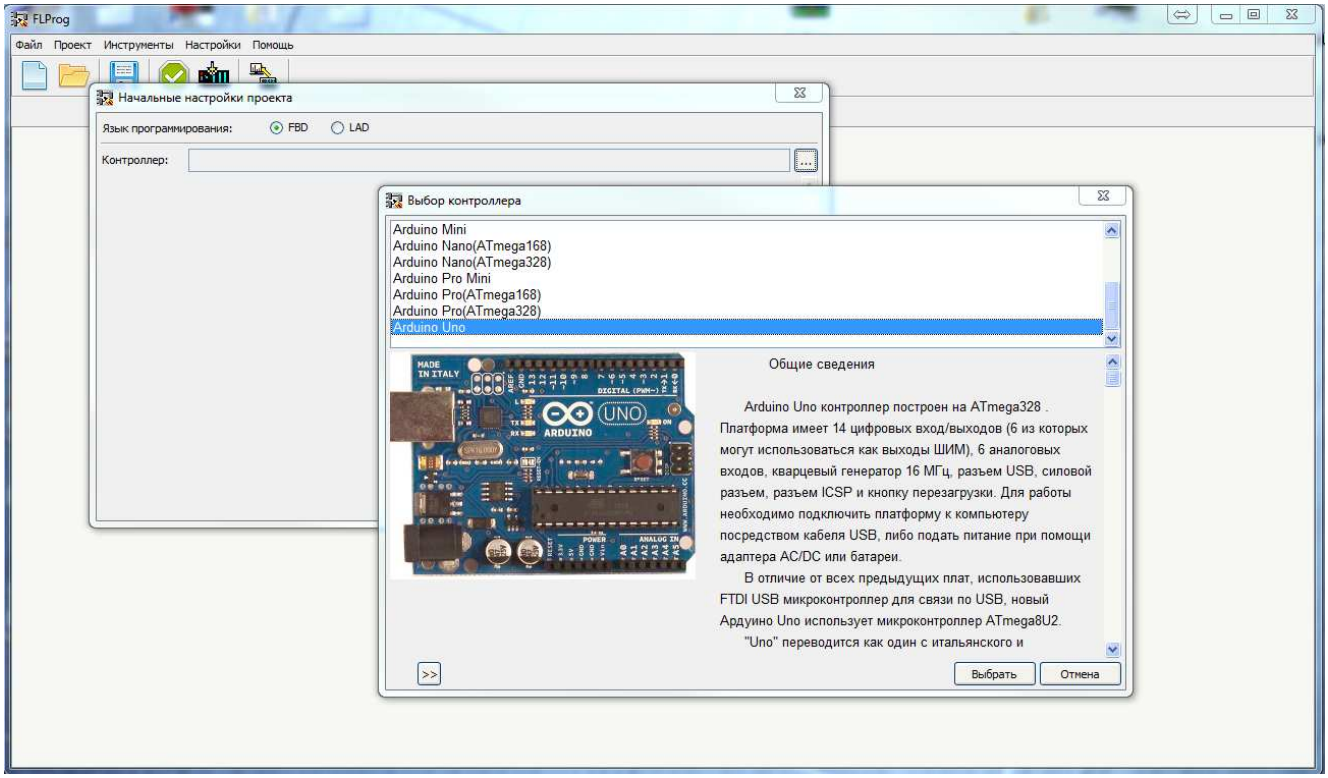
Начнем по порядку

Программа FLProg позволяет создавать прошивки для плат Arduino с помощью графических языков **FBD** и **LAD**, которые являются стандартом в области программирования промышленных контроллеров. При создании программы я постарался максимально использовать наработки программистов Siemens, ABB, Schneider Electric в их средах программирования.

Я немного расширил классический функционал этих языков, добавив функциональные блоки, отвечающие за работу с внешними устройствами. Они являются «обертками» над библиотеками, предназначенными для работы с ними. Программа работает на компьютере под управлением OS Windows. Но энтузиастами из сообщества пользователей программы была опробована возможность запуска программы по ОС Linux. Этому посвящена статья на сайте проекта, которую можно прочитать по ссылке:

http://flprog.ru/FLProg/pid144781676/vpi19538946

При создании нового проекта вам предложат выбрать язык программирования, на котором вы будете создавать проект, и контроллер, на котором этот проект будет реализован.

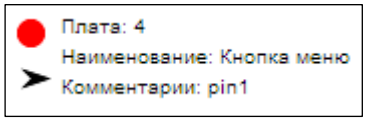


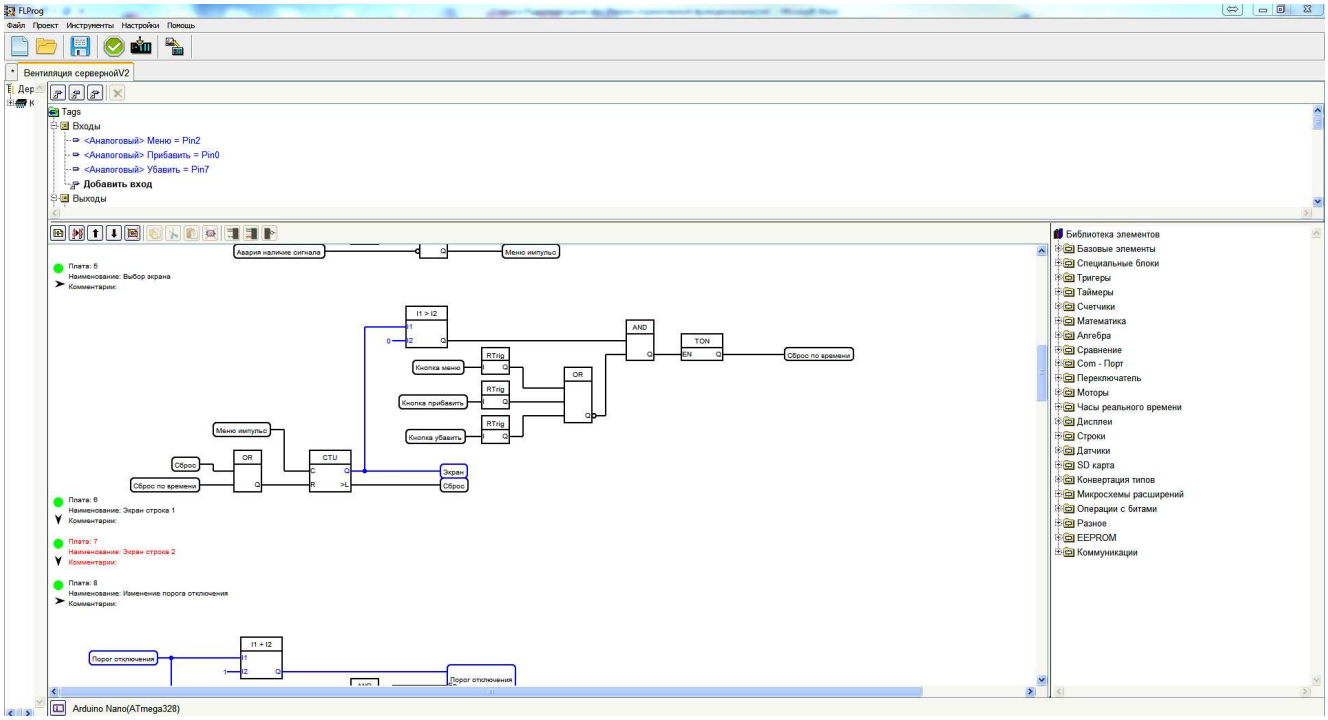
Вот список плат Arduino, поддерживаемых программой на сегодняшний день:

Arduino Diecimila
Arduino Duemilanove
Arduino Leonardo
Arduino Lilypad
Arduino Mega 2560
Arduino Micro
Arduino Mini
Arduino Nano (ATmega168)
Arduino Nano (ATmega328)
Arduino Pro Mini
Arduino Pro (ATmega168)
Arduino Pro (ATmega328)
Arduino Uno

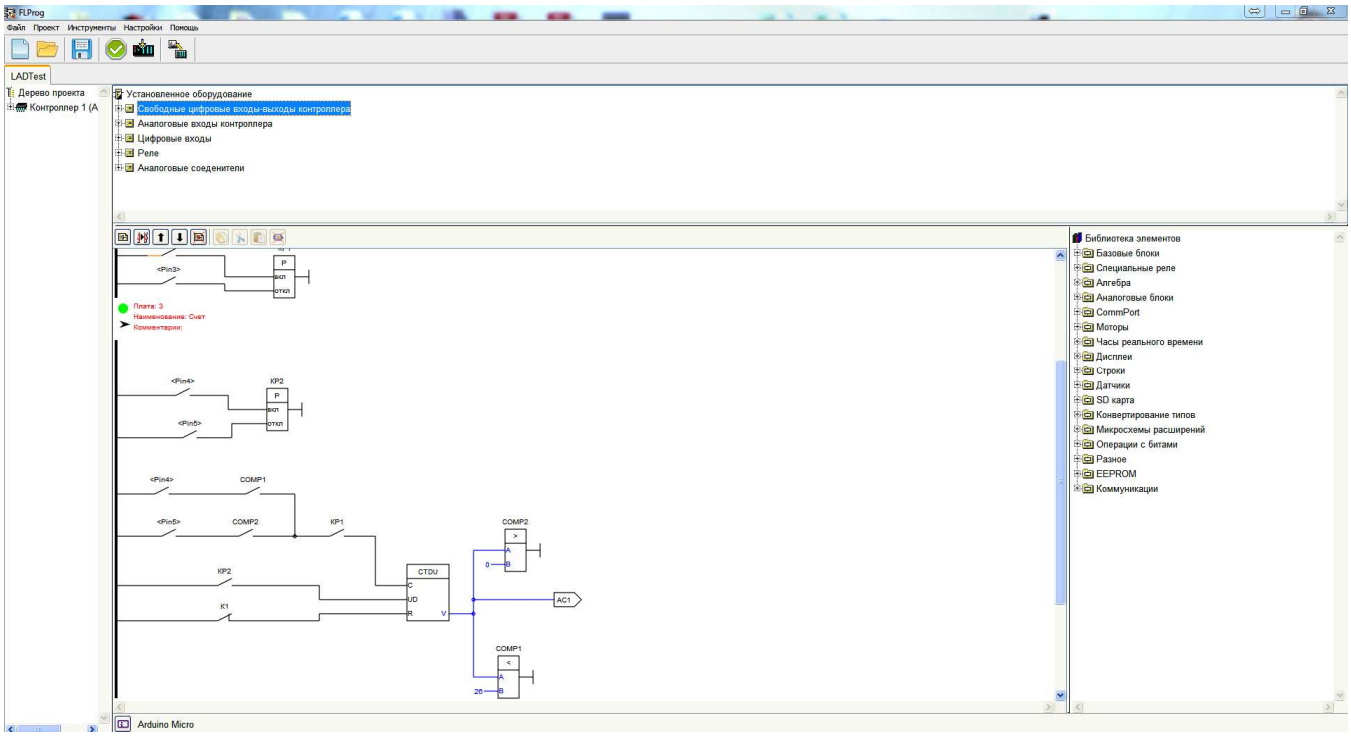
В скором времени ожидается пополнение в семействе поддерживаемых плат. *Arduino Due* уже в пути, а плату *Intel Galileo* (gen.2) обещал предоставить руководитель лаборатории интернета вещей при Санкт-Петербургском Государственном университете телекоммуникаций им. проф. М.А. Бонч-Бруевича. Со временем, по мере приобретения, планируется поддержка плат основанных на контроллерах STM.

Проект в FLProg представляет собой набор плат, на каждой из которых собран законченный модуль общей схемы. Для удобства работы каждая плата имеет наименование и комментарии. Так же каждую плату можно свернуть (для экономии места в рабочей зоне, когда работа над ней закончена) и развернуть. Красный индикатор в наименовании платы указывает на то, что в схеме платы есть ошибки.





Вид окна программы в режиме языка FBD



Вид окна программы в режиме языка LAD

В правой части рабочей зоны расположена библиотека элементов. В схему элементы переносятся простым перетаскиванием. При двойном клике по элементу будет показана информация о нём.

Вот список блоков доступных на сегодняшний день.

FBD:

Базовые элементы
- *[XOR]*
- *[AND]*
- *[OR]*
- *[Bounce]*

Специальные блоки
- *[Scale]*

Тригеры
- *[SR]*
- *[TT]*
- *[RS]*
- *[Rtrig]*

Таймеры
- *[Generator]*
- *[Timer]*

Счетчики
- *[Counter]*
- *[SpeedCounter]*

Математика
- *[SUM(+)]*
- *[MUL(*)]*
- *[SUB(-)]*
- *[DIV(/)]*

Алгебра
- *[SIN]*
- *[COS]*
- *[TAN]*
- *[ABS]*
- *[SQ]*
- *[SQRT]*
- *[MIN]*
- *[MAX]*
- *[POW]*
- *[RANDOM]*

Сравнение
- *[Comparator]*

Com - порт
- *Send*
- *SendVariable*
- *ReceiveVariable*

Переключатель
- *[SWITCH]*
- *[MUX]*
- *[DMS]*

Моторы
- *ServoMotor*
- *StepMotor*

Часы реального времени
- *[Alarm]*
- *[GetTime]*
- *[SetTime]*

Дисплеи
- *Дисплей на чипе HD44780*
- *Подсветка дисплея на чипе HD44780*
- *I2C*
- *Блок декодирования семисегментного индикатора*

Строки
- *Сложение строк*

Датчики
- *[Ultrasonic HC-SR04]*
- *[DHT11, DHT21, DHT22]*
- *[DS18x2x]*
- *[IR Ressive]*
- *[BMP-085]*

SD карта
- *Запись переменной на SD карту*
- *Выгрузка файла с SD карты*

Конвертация типов
- *Преобразование строк*
- *Преобразование Float в Integer*

Микросхемы расширений
- *Расширитель выводов 74HC595*

Операции с битами
- *Шифратор*
- *Дешифратор*
- *Чтение бита*
- *Запись бита*

Разное
- *Матричная клавиатура*
- *Пьезодинамик*

EEPROM
- *Запись в EEPROM*
- *Чтение из EEPROM*

Коммуникации
- *SendVariableFromCommunication*
- *RessiveVariableFromCommunication*
- *WebServerPage*
- *WebClient*

LAD:

Базовые блоки
- *Контакт*
- *Катушка*
- *Защита от дребезга*
- *Выделение переднего фронта*

Специальные реле
- *Двустабильное реле*
- *Реле времени*
- *Генератор*
- *Реле сравнения*

Алгебра
- *SIN*
- *COS*
- *TAN*
- *ABS*
- *MAX*
- *MIN*
- *SQ*
- *SQRT*
- *POW*
- *RANDOM*

Аналоговые блоки
- *Масштабирование*
- *Математика*
- *Счетчик*
- *Аналоговый переключатель*
- *Переключатель много к одному*
- *Переключатель один ко многим*
- *Аналоговый вход контроллера*
- *Аналоговый выход контроллера*
- *Вход аналогового соединителя*
- *Выход аналогового соединителя*
- *Скоростной счетчик*

CommPort
- *Передача в ComPort*
- *Передача переменной через Comm port*
- *Прием переменной через Comm port*

Моторы
- *Сервомотор*
- *Шаговый двигатель*

Часы реального времени
- *Получить данные*
- *Будильник*
- *Установка времени*

Дисплеи
- *Дисплей на чипе HD44780*
- *Блок управления подсветкой дисплея на чипе HD4480 I2C*
- *Блок декодирования семисегментного индикатора*

Строки
- *Сложение строк*

Датчики
- *Ульразвуковой дальномер HC-SR04*
- *Датчик температуры и влажности DHT11 (DHT21, DHT22)*
- *Датчик температуры DS18x2x*
- *IR Ressive*
- *BMP-085*

SD карта
- *Запись переменной на SD карту*
- *Выгрузка файла с SD карты*

Конвертирование типов
- *Конвертация строк*
- *Преобразование Float в Integer*

Микросхемы расширений
- *Расширитель выводов 74HC595*

Операции с битами
- *Шифратор*
- *Дешифратор*
- *Чтение бита*
- *Запись бита*

Разное
- *Матричная клавиатура*
- *Пьезодинамик*

EEPROM
- *Запись в EEPROM*
- *Чтение из EEPROM*

Коммуникации
- *Блок отправки переменной через коммуникации*
- *Прием переменной через коммуникации*
- *Страница Web сервера*
- *Web клиент*

В настоящее время ведется разработка функциональных блоков для работы с трех-осевым гироскопом, люксометром, и другими датчиками и сенсорами. Также ведется работа над организацией обмена данными через блютуз, радиоканал, и интерфейс RS-485. В дальнейших планах – разработка SCADA-системы для организации интерфейса систем, разработанных в программе FLProg на персональном компьютере или графических дисплеях.

Список периферийного оборудования, поддерживаемого программой, доступен на сайте проекта по ссылке:

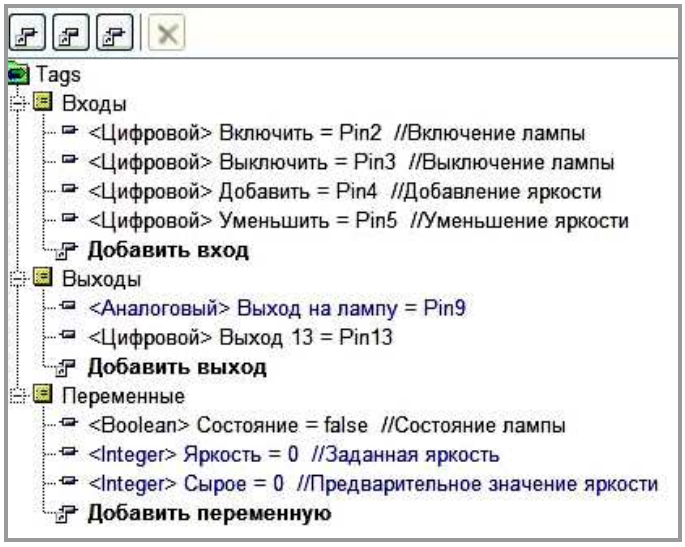
http://flprog.ru/FLProg/pid218088913/vdi194000369

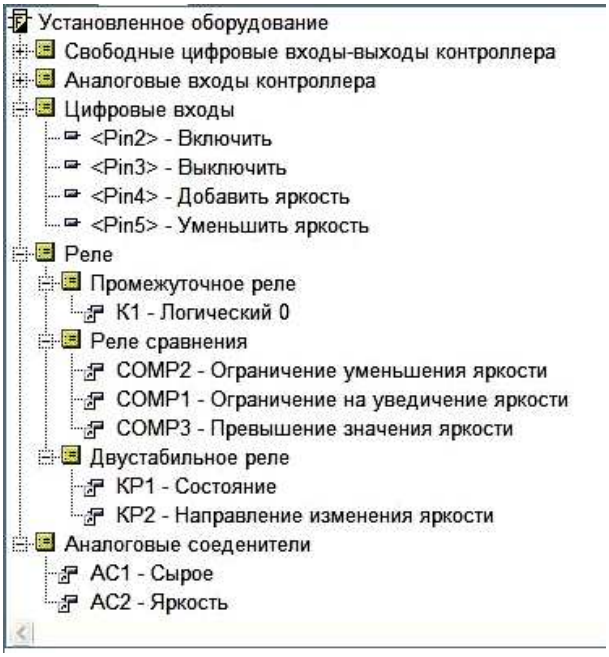


Датчик температуры и влажности DHT11, DHT21, DHT22. Подробнее в статье

Для части оборудования в разделе на сайте присутствуют обзорные статьи, облегчающие понимание применения его в программе.

В верхней части рабочей зоны расположен список тэгов (переменных и входов выходов) (FBD) или установленного оборудования (LAD). Тэги или оборудование переносятся на схему простым перетаскиванием.





После завершения работы над проектом производится его компиляция. После компиляции автоматически откроется программа "Arduino 1.5.7" с загруженным скетчем вашего проекта. В программе "Arduino IDE 1.5.7" вам необходимо будет указать номер COM-порта, к которому подключен ваш контроллер, выбрать его тип, и произвести заливку скетча в контроллер. Подробнее о программе "Arduino IDE 1.5.7" можно почитать на сайте Arduino.ru.



Где скачать FLProg?

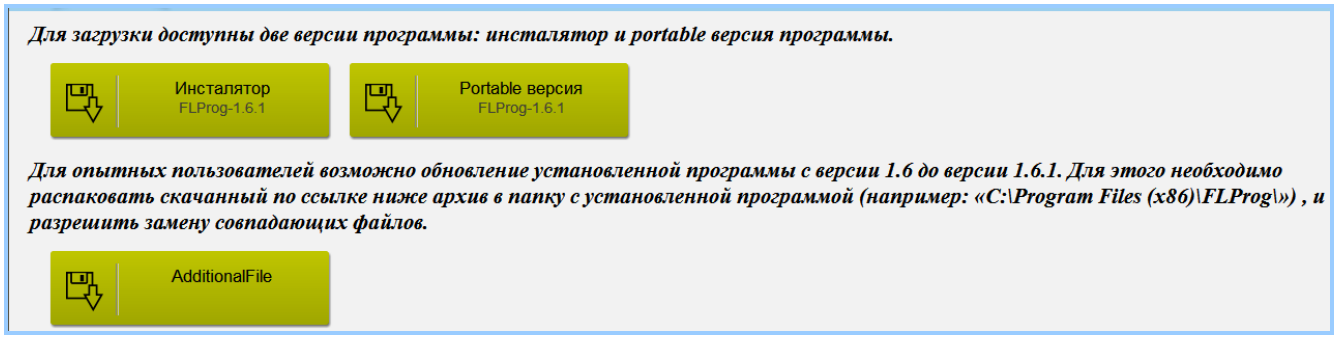
В рамках проекта существует сайт http://flprog.ru
Основная задача сайта – дать возможность пользователям скачать последнюю версию программы, узнать о нововведениях и изменениях.

Скачать программу можно без регистрации на сайте, но для зарегистрированных пользователей функционал сайта заметно расширяется. Регистрация очень проста и требует только подтверждения электронной почты. Никаких других данных при этом вводить не требуется.



На странице загрузки программы всегда доступны две версии: инсталлятор и портативная версия, не требующая установки. Если возможно, то я также выкладываю файл обновления значительно меньшего размера, позволяющий обновить предыдущую версию.

Также на странице загрузки можно посмотреть список нововведений и исправленных ошибок для данной версии и перейти в архив предыдущих версий.



Общение



Вторая задача сайта – обратная связь между пользователями и разработчиком. Для этого на сайте создан баг – трекер. Незарегистрированные пользователи могут просматривать список обнаруженных ошибок, их состояние. А зарегистрированные могут добавлять в список свои обнаруженные ошибки. Я стараюсь, чтобы к выходу очередной версии баг-трекер был чистым, и, практически, это всегда удается.

Также цели обратной связи от пользователя к разработчику служит страница нового функционала. Организация этой страницы немного другая. Незарегистрированные пользователи также могут просматривать пожелания к новым функциям программы, а зарегистрированные могут добавлять свои «хотелки», и голосовать за уже существующие. Таким образом, формируется список наиболее востребованных функций, требующих реализации, которого я стараюсь придерживаться.

167	Голосов: 171.1	Категория: Интерфейс	Статус: Принята
Автор: Администрация			
форматирование преобразования в строку			
142	Голосов: 160.063	Категория: Язык FBD	Статус: Принята
Автор: ValeRICH			
Работа с модулем NRF24L01			



Самое интересное с моей точки зрения место на сайте – форум. Здесь пользователи могут задать вопросы по работе с программами, спросить совета и обсудить проблемы. Форум живет очень интенсивно и требует от меня все меньше времени. Уже появились свои гуру и старожилы, которые помогают начинающим. Я очень им благодарен, поскольку это снижает нагрузку на меня. Здесь, как и на других страницах сайта, незарегистрированные пользователи могут просматривать сообщения, а зарегистрированные имеют возможность полного общения. Никаких рейтингов и уровней не ведется. Не вижу в этом смысла. Вообще, на форуме собрались очень серьезные и отзывчивые люди. Пока за год работы не отмечено никаких попыток хулиганства, рекламы всякой всячины, чем обычно болеют другие форумы. Я, конечно, оставил за собой право модерации и «бана», но пока ни разу воспользоваться этим не пришлось.

Общение с Администрацией	Последнее сообщение 7 января 2015 г. 18:42:44	Тем: 72 Сообщений: 505
В этом разделе Вы можете узнать о новостях проекта, задать вопросы администрации сайта, почитать о правилах поведении на форуме		
О сайте и пограмме	Последнее сообщение 8 января 2015 г. 15:38:02	Тем: 72 Сообщений: 296
В этом разделе Вы можете обсудить сайт и программу		
Разное	Последнее сообщение 3 января 2015 г. 13:05:58	Тем: 34 Сообщений: 195
Раздел для разговоров на разные около Ардуинные темы		
Документация	Последнее сообщение 9 ноября 2014 г. 20:06:28	Тем: 2 Сообщений: 4
В этом разделе можно обменяться ссылками или посто рассказать об информыции относительно Ардуино, или языках FBD и LAD		

Для удобства пользователей сайта создан файловый хостинг для хранения изображений и файлов, вставляемых в сообщения. Поскольку место у меня на диске не бесконечное, я регулярно провожу удаление неиспользуемых на сайте файлов со сроком хранения больше месяца.

Обучение

Для обучения начинающих пользователей программы существует страница документации. На ней можно посмотреть:
[описание программы](),
[описание сайта](),
[видео-уроки и примеры работы с программой](),
[список оборудования, применяемого в программе]().



Видео-уроки описывают большую часть функционала программы. Первые вводные уроки получились, конечно, слишком длинными – более часа каждый, но мне не удалось порезать их на части без потери логической завершённости.



Творчество

В виде эксперимента на сайте создан раздел [публикаций]. За основу взята идеология Хабра. Зарегистрированные пользователи могут писать свои статьи и комментировать написанные другими пользователями публикации. В этом разделе присутствуют статьи о разработках пользователей программы, описание большинства из используемого в проекте оборудования.



7 января 2015 г. 17:48:29	Автор: alecs.83 Тема: Проекты пользователей
Небольшой проект. Кухня- датчик движения- свет	
27 декабря 2014 г. 15:51:51	Автор: Former Naval Person Тема: Разное
FLProg и пингвины	
27 декабря 2014 г. 9:47:26	Автор: ALEX Тема: Проекты пользователей
Новогодняя мигалка	
23 декабря 2014 г. 2:33:54	Автор: aleksys77 Тема: Проекты пользователей
Коммутатор ёлочных гирлянд	
18 декабря 2014 г. 4:33:35	Автор: Администрация Тема: Железо
LCD Keypad Shield для Arduino	

Сообщество пользователей

На сегодняшний день количество зарегистрированных пользователей проекта более четырехсот человек. Создана группа на сайте "В контакте". Адрес группы:

http://vk.com/flprogusers

Вот некоторые отклики о программе на форуме проекта:

Vint
15 июня 2014 г. 5:19:53

Доброго здравия Всем!
Я не программер, а вот FBD и LAD это мне доступно и понятно, это именно то что мне не хватало для полного счастья. Спасибо, Восхищаюсь Вашим энтузиазмом и проделанной работой

вова
17 июня 2014 г. 5:19:53

Сам киповец, по своей работе приходится работать с контроллерами и со скадой, программирование в мэк 63131 знаком, с ардуино познакомился в конце2012г., скажу честно это игрушка для взрослых т.к. микроконтроллеры должны программировать в С++, но ваш проект по созданию альтернативной программной среды для более простого чми. меня воодушевил, в связи с этим презентовал проекту пока 200р, с надеждой на дальнейшее развитие. Было-бы здорово если линейка микроконтроллеров в дальнейшем расширялась

Автор: Любитель

Здравствуйте!
Хотел поблагодарить Вас за труды. Скачал обновленную версию программы и был просто поражен функциональным блоком "Дисплей", насколько упростили вывод сообщений на дисплей.
Тут же накидал простенький вольтметр на 5 вольт(MAX) и вроде как показывает, но только целые числа. Возможно ли вывести и десятичные? Если да, то подскажите пожалуйста как?
Заранее спасибо.

Автор: vns

Большое спасибо за программу! Хотел через ардуинку приобщиться к большому и загадочному Си-плюсу но с вашей помощью эта идея провалилась. За пару часов решил поставленную задачу. Для абсолютного счастья хотелось бы увидеть поддержку Ethernet-шилда. Заранее благодарен.

Автор: Kroshka.nikki

Уважаемый автор! Огромное спасибо за проделанный труд! Ардуинка пылилась давно, но сегодня найдя ваш чудесный софт, чувствую заживет своей жизнью. Дело в том что из меня программист не очень, да и долго осваивать(ленивый ж),а тут все просто и понятно,с простым интерфейсом и описанием) Вот раз уж ожило детище, наткнулся на ОРС сервер, проверил на стандартном примере-работает, Trace mode успешно увидел ее. Вот бы добавить функционал орс, ну или хотябы modbus, было бы уж совсем Здорово.

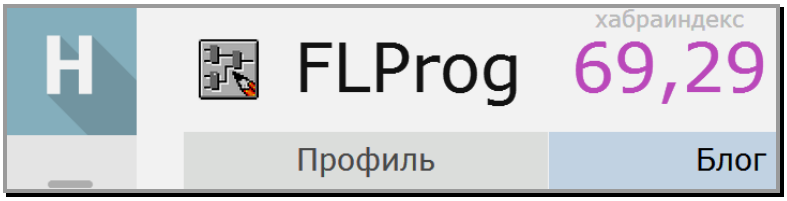
Автор: djon1

Доброго дня!!! Сразу бы хотелось поблагодарить администрацию сайта, молодцы так держать. Мне не очень дается программирование, но вот Ваш вариант FBD позволяет схватывать и решать необходимые мне задачи на лету. Спасибо ВАМ! За свою благодарность "Помог проекту"!

Автор: СашаЖ

Здравствуйте, случайно наткнулся на Ваш как оказалось очень нужный сайт. Сам никогда не занимался программированием, только готовые HEX заливал и тут такая возможность попробовать самому. Собрал ардуино попробовал КЛАСС. Простые проекты получается в протэусе симулировать. Благодаря видео урокам дисплей показывает температуру и влажность , шаговик крутит, и тут пришла в голову мысль сделать родителям инкубатор, и тут засада как заставить поддерживать температуру в заданных пределах. Хотелось бы увидеть в видео уроках пример простого терморегулятора

На известном IT портале Habrahabr.ru проект представлен корпоративным блогом:



http://habrahabr.ru/company/flprog/

Спасибо администрации Хабра, за предоставление корпоративного блога по программе “Стартап”. Аудитория портала встретила проект неоднозначно. Вот пример переписки в комментариях к посту о проекте:


segrus 2 ноября 2014 в 15:14 #

Не знаю, где вы увидали, что STM32 дороже AVR… Отладочные платы (некоторые) возможно. Но сами кристаллы в разы дешевле, а возможностей в STM больше. Если Вам надо просто поучиться программированию, то Arduino возможно неплохой вариант. Но на производстве всё должно быть всерьёз и надолго.

Я лично себе на завод не поставлю промышленный контроллер на AVR и языке, созданном для обучения.

Надо смотреть в будущее. А будущее за ARM.

Вот такое ИМХО (:


gbg 2 ноября 2014 в 15:31 #

Сами AVR в ответственной технике прекрасно работают — в автомобилях, в автоматических медицинских анализаторах (от результатов анализов крови прямо зависит дальнейшее лечение) и так далее, так что с надежностью у AVR проблем нет — обычный industrial grade, если по-ихнему.

По поводу стоимости — ST активно демпингует, так что их отладочные платы нынче даже в Чипе и Дипе в целом дешевле Ардуин.

При наличии приятного фарша вроде распаянного ST-Link (можно отлаживать прошивку в любимой IDE как будто это десктопное приложение), эти платы так и просятся в руки.


segrus 2 ноября 2014 в 15:56 #

Я в курсе про надёжность атмеловских ядер. в своё время мы их гоняли на температурах до +150 градусов, где они (некоторые отобранные экземпляры) вполне успешно работали, в отличие от тогдашних STM.

Просто автор пишет программу для учебных слабомощных плат. Пройдёт немного времени и ARM вытеснит AVR и вместе с ним Arduino. И что потом опять всё менять и переходить на ARM?

Знаете сколько стоит час простоя конвейера?

Если для себя побаловаться или детишек поучить, то пожалуйста.

А в промышленность должны идти только профессиональные, проверенные и долгосрочные продукты.

Вот об этом моя мысль.

А вообще, бесит уже, что каждый начинающий программист вместо того, чтобы нормально выучить существующий принятый всеми язык, старается придумать свой новый. А если не хватает мозгов придумать язык, придумывает кучу библиотек и втирает всем, что с ними вам уж точно будет легче, чем раньше…


totuin 2 ноября 2014 в 16:13 #
Я не придумал язык, я взял готовый, всеми принятый именно в этой области. И почему всякий "настоящий" программист считает только свой язык настоящим и принятым. Ну и во вторых пока никто не собирается применять ардуинку в промышленности, а вот собрать автоматический сварочник, или систему управления освещением для своего старенького жигуленка, или ту же погодную станцию — это самое то. И что для этого изучать С? Почему бы не перенять опыт того же Сименса и сделать упрощенный вариант его TiaPortal.


segrus 2 ноября 2014 в 16:24 #
А почему не изучить С? Для простых контроллерных применений вам понадобится несколько простых команд. И кстати, мне как энергетику, проще было бы программировать промышленные контроллеры на С, а не строить бесконечные «схемы» на языках типа Сименса.
Но вы поймите меня правильно. Я не пытаюсь вас поругать или остановить. Я хочу донести и другой взгляд на данный вопрос. Дискуссия и ничего личного ;)


totuin 2 ноября 2014 в 16:35 #
Конечно — же ничего личного. Но с моей точки зрения, кто захочет изучить тот же С например — возьмет и изучит. Даже после моей программы. У меня в пользователях есть один программист. Он мне пишет что использует программу для создания заготовки, а потом то чего нет в функционале дописывает уже в готовом скетче. То же вариант. Хотя я конечно им восхищаюсь, поскольку после компилятора код выходит не очень то удобочитаемый. Но заставлять всех обязательно учить именно С — это неправильно. Я просто хочу дать людям выбор.


gbg 2 ноября 2014 в 16:43 #
Этот программист делает именно то, чего я боялся — берет продукцию вашего генератора, а потом модифицирует вручную. А вы еще удивляетесь, зачем я простыню об этой продукции сочинил.


totuin 2 ноября 2014 в 16:46 #
А чем это плохо. Я не понимаю. Ну пишет он для себя, Понимает что делает. Не понимал бы не брался, а подождал пока нужный функционал появится. В чем проблема — то?


Alexeyslav 3 ноября 2014 в 12:54 #
На С легко сделать ошибку которую трудно будет найти, схемы они как-то нагляднее и ошибку сделать будет достаточно трудно, поэтому они и применяются.


gbg 2 ноября 2014 в 16:36 #
Ардуино уже оказалась в профессиональной деятельности:
1. Arduino управляет промышленным грузовым лифтом — ни один здравомыслящий человек не допустит самоделку без кучи сертификатов и тестов к управлению потенциально опасным объектом.
2. Вскрытие показало: виноват пробел — в падении сервиса виновен любимый метод написания программ — копипаст+опечатка = Эффект последней строки
А разработка автоэлектроники или сварочного автомата таки требует знаний, более широких нежели «здравый смысл». Иначе из-за непредусмотренных эффектов можно получить материальный ущерб, или чего похуже. О том, как совсем неглупые люди имели проблемы в огромных количествах, разрабатывая сложный технический объект (а все аппаратно-программное, оно уже «сложное»), можно почитать у того же Чертока, в четырехтомнике «Ракеты и люди»


totuin 2 ноября 2014 в 16:44 #

Тот же опытный электрик лучше поймет схему, и сделает её более безопасной, чем будет собирать код из кусочков надерганных из интернета, ничего не понимая в них. Или же вы предлагаете ему вообще запретить что то делать — поскольку нефик. Не умееш программировать — нефик изобретать. В одном из предыдущих постов в комментариях один деятель предлагал уволить за профнепригодность всех асушников не знающих языков программирования. Я думаю он не прав. Кстати копипаст — это как раз способ написания кода в C. В FBD как раз пробелы не пишутся поскольку код там вообще не пишется


gbg 2 ноября 2014 в 16:54 #

Возьмите, например, схему пассажирского лифта с автоматическим приводом дверей. Сколько времени у вас уйдет на то, чтобы выяснить, как схема определяет, в какую сторону запускать кабину, не читая отдельное описание логики ее работы?
Хороший код позволяет восстановить логику своей работы просто путем прочтения, сразу. Не зря изобрели люди именно язык программирования — мы мыслим и общаемся словами и фразами, а не схемами.
Не зря отмечают, что зачастую филологи выдают более внятные программы, нежели математики — они рассказывать умеют.


segrus 2 ноября 2014 в 17:03 #

Ну тут вы уже перегибаете) нормальный АСУТПшник прекрасно поймёт куда поедет лифт, когда посмотрит схему. Другое дело — удобство. В схеме чтобы сделать сложную функцию, мне надо разложить её на элементарную логику. Потом обязательно в правильном порядке эти кусочки соединить, чтобы получить верную функцию на выходе.
С одной стороны, электрик видит знакомые ему выключатели, реле, кнопки…
Но, когда это всё завязано в огромную схему и всё друг от друга зависит, образуя логические функции, у него мозг закипает.
Другое дело, когда он читает:
«Если Датчик1=1 то Выход3=0...»
Вот над этой золотой серединой между схемой и программой и бьются умы человечества много лет уже)


gbg 2 ноября 2014 в 17:19 #

Так я говорю именно об этом, на более-менее сложной схеме мозг «закипает», а если схема на многих листах — разбор превращается в мрак из прослеживания кучи соединений и взаимозавязок.


totuin 2 ноября 2014 в 17:35 #

Примерно то же самое можно сказать и про язык. Если методов штук пять, и классов десяток — то все хорошо. А если классов 5000 и у каждого по 100 методов? Тут то же мозг закипает. Да еще взаимодействия между ними. А если еще система change от вложенных классов наверх, с обработкой параметров. Тут то же мозги закипают. А не дай боже логическое кольцо и программа зацикливается, попробуй разорви. Конечно stack overflow помогает, но все равно трудно. Тут сложно определить что лучше. Просто кто то думает как программист, кто то как электрик. У меня бывает мозги взрываются и от того и того. На работе турбина от сименса на несколько десятков тысяч чартов. На досуге проект на пару тысяч классов наверное уже (не считал, но если еще добавить движок сайта то точно получится). Но это же интересно.

Кстати программы на лифты написаны именно на FBD, ну или на LAD — е. Именно потому что там стоит промышленный сертифицированный контроллер. А они все пишутся именно на этих языках. И как вы с прекрасно представляете как поведет себя программа читая код, так и асушник прекрасно понимает как поведет себя схема читая её. У электрика портянка схем, у Вас портянка кода в миллион строк. Так же как у Вас законченные логические кусочки в методах, так и в схемотехнике логически законченные куски схем. В С — осмысленные названия переменных, в FBD — логически осмысленные названия тэгов. В принципе разницы не так много — разное представление. Я не призываю отказаться от С и других языков. Я просто хочу дать альтернативу. Этот проект и представляет собой альтернативную среду программирования а не замену.

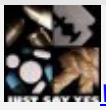

segrus 2 ноября 2014 в 17:22 #
ну вот и славно! если вы так отстаиваете свою идею, уверен вас ждёт успех. когда прославитесь, не забудьте дать нам с gbg по маленькому пакету акций за то, что мы в вас поверили! ;)


totuin 2 ноября 2014 в 17:26
Обязательно, только проект не коммерческий, и не будет таким никогда. Так что акции будут дешёвыми))))
Мне просто интересно этим заниматься. Вахты такие длинные, и не хочется на них отупеть. Мозги должны работать.

Эта переписка хорошо показывает первую реакцию сообщества читателей Хабра на проект. После выхода еще нескольких постов реакция на программу сменилась:

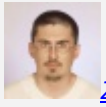

btd 5 ноября 2014 в 14:22
Планируются ли версии не только для Windows?


totuin 5 ноября 2014 в 14:25 #
Планируются, но не скоро. Я пока не разобрался с некоторыми тонкостями работы в Mac и Linux. Как разберусь, постараюсь перенести.


Wedmer 5 ноября 2014 в 15:55 #
Если будет нужна помощь в портировании, пишите лс.


totuin 5 ноября 2014 в 16:03 #
Спасибо, сейчас немного разберусь с делами и у меня будет к вам несколько технических вопросов


nemilya 6 ноября 2014 в 23:32 #
Интересный проект! Кстати Scratch так же сделан на SmallTalk.


ZigFisher 22 ноября 2014 в 21:38 #
Автору — большое спасибо и успехов в развитии программы.
Ждем Linux версию… ;)

Поэтому я надеюсь, что проект найдет своих поклонников и среди аудитории Хабра.

Last but not least

В свете текущей политики импортозамещения проект представляет собой особый интерес. На сегодняшний день в России я знаю очень мало проектов, в которых для программирования контроллеров используется отечественное программное обеспечение. Программу FLProg можно адаптировать для работы с другими контроллерами, отличными от плат Arduino, что позволит применять российское ПО в программировании ПЛК.



Портал научно-практических публикаций
Portal of scientific & practical publications
ISSN 2313-0342
18+

Портал научно-практических публикаций
http://portalnp.ru/

Приглашаем всех желающих публиковать статьи, обзоры, книги, учебники на Портале научно-практических публикаций.
Издатель портала – Международный научно-инновационный центр (ООО).
Публикации на портале индексируются в системе цитирования Google Scholar.
Публикация на портале – бесплатная.

Публикации автора **Dr. Bob Davidov** в рубрике «50.00.00 Автоматика. Вычислительная техника»

Самонастраиваемая высокоскоростная система прецизионного перемещения на базе контроллера EPOS

Excel OPC клиент

Модуль дискретного ввода-вывода ОВЕН МК110 (RS-485, ModBus-RTU/ ASCII, DCON, ОВЕН)

TCP/IP обмен данными

Создание отчетов в МатЛАБ

Управление COM портом в LabView

Преобразователь интерфейса порта последовательной передачи данных

Технология моделирования электронно-механических устройств в МатЛАБ: SimPowerSystems, Simscape

Интеграция VHDL и Simulink моделей

Вычисления в реальном времени с использованием таймера MatLAB

Система термостатирования на базе промышленного модуля дискретного ввода-вывода ОВЕН МК110 (RS-485, ModBus)

LabView OPC клиент

Генератор отчетов LabVIEW с использованием macros

Освоение MasterSCADA

Построение графического интерфейса пользователя (GUI) в среде МатЛАБ для целей управления и наблюдением за состоянием объекта

Фильтрация сигналов. Средства разработки фильтров в МатЛАБ

Обмен данными через COM port

Интерфейс пользователя LabVIEW для моделей Simulink: Simulation Interface Toolkit

Знакомство с компилятором GNAT языка Ада для программирования систем управления автоматизированными комплексами

Использование памяти виртуальным прибором LabView

Симулятор устройств ModBUS RTU-TCP/IP

Simulink OPC DA клиент. OPC Toolbox

Влияние методов интегрирования на результаты моделирования

Система термостатирования на базе USB интерфейса Lcard E14-440 (S-function)

Интернет-клиент MasterSCADA

DLL Lcard USB интерфейс для CPP процесса

Программный интерфейс MEX. Расширение MATLAB на языке АДА

Многоканальная измерительная система ввода, вывода и обработки аналоговой и цифровой информации USB L-card

OPC DA m-file клиент

Протокол ModBus-RTU / ASCII

Узел скриптов приложения MATLAB в LabVIEW

UDP обмен данными

Построение интерфейса пользователя многоканальной измерительной системы ввода, вывода и обработки аналоговой и цифровой информации LabView USB L-card

Система термостатирования на базе PCI интерфейса NI6014 (Sensoray 626)

COM/DCOM клиент и сервер автоматизации Microsoft ActiveX control

Программные имитаторы сигналов OPC сервера

Автоматическая передача e-mail сообщений через SMTP сервер

Система термостатирования на базе интерфейса LabView

Среда разработки интерфейсов и программных модулей систем управления LabVIEW

МатЛАБ OPC HDA клиент

Улучшение характеристик квантования с использованием псевдослучайного сигнала. Dithering – как метод сжатия данных

Simulink. Работа моделей в режиме реального времени

Автоматическая оптимизация Simulink моделей и регуляторов

Управление LPT портом в среде МатЛАБ

Обеспечение адекватности математической модели ее физическому прототипу

Управление из Excel через COM порт

Установка Modbus OPC DA/HDA сервера компании ИнСАТ

Создание интерактивных объектов и сред на базе платформы Arduino

Математическое моделирование систем управления в среде Simulink

Идентификация объектов систем управления. System Identification Toolbox

Средства проектирования линейных стационарных систем управления по известной структуре и параметрах системы. МатЛАБ Control Systems Toolbox

Построение локальной сети с выходом в Интернет

Генерация кода Программируемых Логических Контроллеров (ПЛК) в среде МатЛАБ, PLC Coder

Стенд для разработки алгоритмов высокоскоростного управления шаговым приводом

Компьютерные средства систем управления. Raspberry Pi

Многоканальное устройство ввода и накопления аналоговых данных на базе MS Excel

Использование общей памяти накопителя: текстовый файл, электронные таблицы

Подключение периферии к среде разработки систем управления МатЛАБ

MEXW32 обмен данными модуля USB L-card

Импорт и экспорт МатЛАБ данных через Raspberry Pi

Построение RT системы управления на базе компьютера Raspberry Pi

Удаленное управление через Веб-сервер

Dr. Bob Davidov

Кафедра Автоматики и процессов управления Санкт-Петербургского государственного электротехнического университета “ЛЭТИ”

Создание интерактивных объектов и сред на базе платформы Arduino

Цель работы: рассмотрение вариантов сопряжения сред проектирования МатЛАБ, Simulink, LabVIEW с внешней средой через электронную платформу (контроллер) Arduino.

Задача работы: подключение аппаратного интерфейса к среде проектирования MatLAB.

Приборы и принадлежности: Персональный компьютер, контроллер Arduino UNO или DFRduino UNO R3, USB кабель, датчик температуры LM35, Интегрированная среда разработки Arduino -1.0.4., COM Port Toolkit, MATLAB R2008a или новее, Simulink - MATLAB R2010a или новее, LabView.

ОБЩИЕ СВЕДЕНИЯ

Arduino — аппаратная вычислительная платформа (универсальный контроллер), основными компонентам которой являются плата ввода/вывода и среда разработки на языке Processing/Wiring. Через последовательный порт Arduino можно подключить и к другим программным средам, например, Matlab, Simulink, LabVIEW и они, таким образом, получают инструмент для взаимодействия с реальными объектами систем управления, модулями визуализации или робототехники.

MATLAB m-файлы или Simulink поддерживают совместную работу с контроллером Arduino Mega 2560 или Arduino Uno по USB каналу в режиме RS-232 последовательного соединения. Это взаимодействие основано на выполнении серверной программы контроллера, который принимает команды через последовательный порт, выполняет их и, при необходимости, возвращает результат. Такой подход помогает

- запускать программы сразу при помощи загрузчика контроллера, без дополнительных средств (программаторов),
- работать в среде MATLAB или Simulink для интерактивной разработки и отладки,
- разрабатывать программы ввода аналоговых и цифровых данных,
- управлять двигателями постоянного тока, серводвигателями, и шаговыми двигателями,
- выполнять контурное управление с частотой до 25 Гц (не в реальном времени).
- создавать реальные электронно-механические устройства.

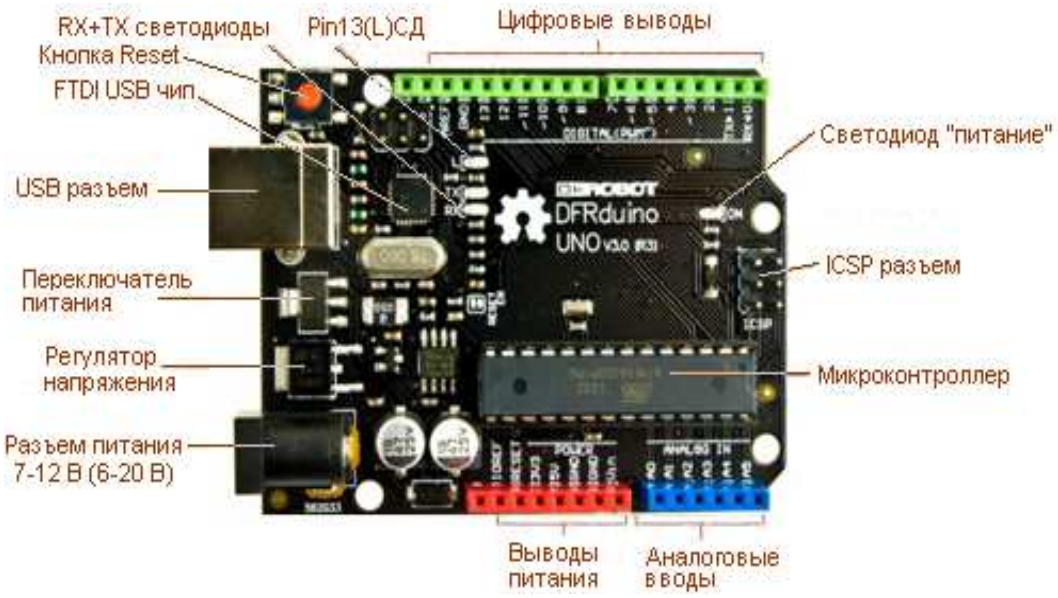


Рис. 1. Периферия платформы ARDUINO (DFRduino) UNO R3. Цифровые выходы, поддерживающие работу в режиме ШИМ обозначены на плате волнистой линией ~. AREF вывод опорного напряжение АЦП (используется с функцией analogReference()).

В этой работе используется аналог Arduino UNO: DFRduino UNO R3. Аналог, как и прототип, имеет следующую спецификацию:

- Микроконтроллер ATmega 328, частота 16 MHz,
- Напряжение питания: 7-12 В (6-20 В - предел). Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB).
- Вывод питания 3.3В/50 mA
- 32 Кб флэш память (2Кб занято загрузчиком),
- 2Кб ОЗУ
- 1Кб EEPROM
- 6 аналоговых (0-5В, 10бит, 0.1мс) вводов и 14 цифровых вводов / выводов (до 40 мА) с 6 PWM (ШИМ) выходами (~490 Гц, 0 .. 255).
- Встроенные USB-COM (300, .., 115200 бод), SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), и I2C: 4 (SDA) и 5 (SCL)каналы связи;
- 2 внешних прерывания

В микроконтроллер предварительно прошит загрузчик, поэтому внешний программатор не нужен. На каждой платформе присутствует линейный стабилизатор напряжения 5В.

На рынке доступны платы расширения для Arduino, известные как «shields».

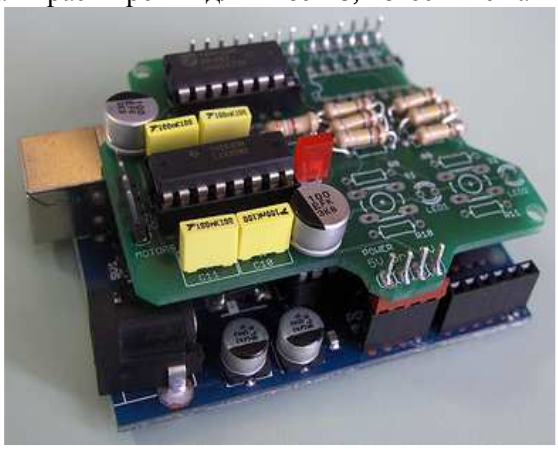


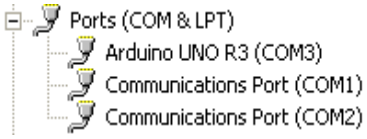
Рис. 2. Пример платы расширения «shields» для Arduino: модуль управления двигателем.

ПОДГОТОВКА К РАБОТЕ С ARDUINO ЧЕРЕЗ ИНТЕГРИРОВАННУЮ СРЕДУ РАЗРАБОТКИ (http://robocraft.ru/blog/arduino/98.html)

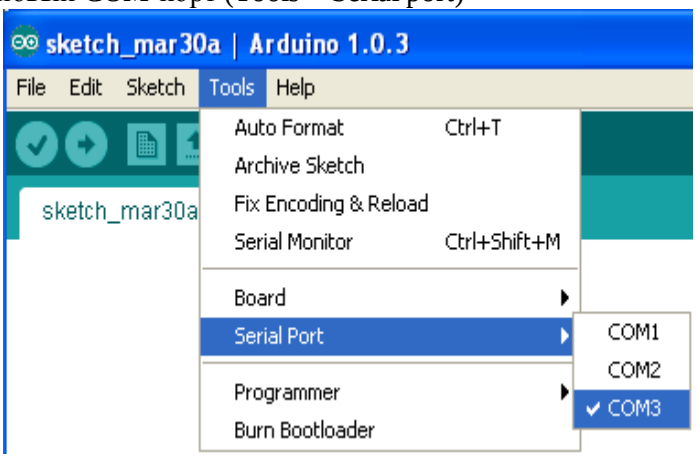
Для того, чтобы иметь возможность писать свои программы и записывать их на Arduino необходимо подключиться к Интегрированной среде разработки Arduino — это кроссплатформенное приложение на Java, включающее в себя редактор кода, компилятор и модуль передачи прошивки в плату. Среда разработки основана на языке программирования Processing. Строго говоря, это C/C++, дополненный некоторыми библиотеками. Программы обрабатываются с помощью препроцессора, а затем компилируется с помощью AVR-GCC.

Для подключения Adruino к интегрированной среде выполните следующие шаги.

1. Скачайте с сайта http://arduino.cc/en/main/software Arduino IDE (arduino-1.0.3-windows.zip).
2. Распакуйте zip файл и поместите директорию arduino-1.0.3 в коневой каталог диска C. Можно и в другое место, но главное, чтобы в пути не было названий отличных от английского.
3. Подключите плату к компьютеру, посредством USB-кабеля типа A-B.
4. Когда операционная система обнаружит новое устройство (ждать надо ~5 мин.) и предложит установить драйвер, укажите драйвер находящийся в директории arduino-1.0.3\drivers\. После установки в системе появится дополнительный COM-порт:



5. Запустите Arduino IDE –C:\arduino-1.0.3\arduino.exe.
6. Выберите новый COM-порт (Tools > Serial port)



7. Выберите тип платы: Tools > Board > Arduino UNO:



8. На этом закончена установка и подключение платформы Arduino, можно начинать программирование.

ПРОГРАММИРОВАНИЕ ARDUINO

Arduino/ Freeduino программируется на языке Wiring коды которого преобразуются с минимальными изменениями в программу на языке C/C++, и затем компилируются компилятором AVR-GCC. Так что, фактически, используется специализированный для микроконтроллеров AVR вариант C/C++.

Среда разработки, и набор базовых библиотек, упрощают доступ к периферии микроконтроллера.

Например, на языке Wiring установка скорости последовательного порта 9600 бит в секунду, задается всего одной строчкой: Serial.begin(9600);

Тогда как при использовании «голого» C/C++ пришлось бы разбираться с документацией на микроконтроллер, и вызывать нечто подобное:

```
UBRR0H = ((F_CPU / 16 + 9600 / 2) / 9600 - 1) >> 8;
UBRR0L = ((F_CPU / 16 + 9600 / 2) / 9600 - 1);
sbi(UCSR0B, RXEN0);
sbi(UCSR0B, TXEN0);
sbi(UCSR0B, RXCIE0);
```

фактически, класс HardwareSerial (\hardware\cores\arduino\HardwareSerial.h) инкапсулирует данные операции в функции begin (\hardware\cores\arduino\HardwareSerial.cpp)

Объявление COM-порта в самом заголовочном файле .h файле: extern HardwareSerial Serial. А
Подключение заголовочного файла класса работы с последовательным портом в программе происходит строчкой: #include "HardwareSerial.h"

Структура программы

В любом тестовом скрипте можно увидеть две необходимые функции: setup() и loop().

Функция setup() запускается один раз, после каждого включения питания или сброса платы Arduino. Она используется для инициализации переменных, установки режима работы цифровых портов, и т.д.

Функция loop() в бесконечном цикле исполняет команды, которые описаны в ее теле.

Рассмотрите простой пример:

```
void setup()        // начальные установки
{
  beginSerial(9600);  // установка скорости работы серийного порта на 9600 бит/сек
  pinMode(3, INPUT);  // установка 3-его порта на ввод данных
}

// Программа проверяет 3-ий порт на наличие на нём сигнала и посылает ответ в
// виде текстового сообщения на последовательный порт компьютера
void loop()                 // тело программы
{
  if (digitalRead(3) == HIGH)  // условие на опрос 3-го порта
```

```
    serialWrite('H');         // отправка сообщения в виде буквы «H» на COM-порт
  else
    serialWrite('L');         // отправка сообщения в виде буквы «L» на COM-порт
  delay(1000);                // задержка 1 сек.
}
```

Константы

Константы – предопределенные значения. Объявление констант (а так же базовых макросов и функций) можно увидеть в файле \hardware\cores\arduino\wiring.h

Уровни сигналов порта HIGH и LOW

```
#define HIGH 0x1          // 5V out, >3V in       “1”
#define LOW  0x0          // 0V out, <2V in       “0”
```

Таким образом, оба следующих вызова будут эквивалентны:

```
digitalWrite(13, HIGH);   // можно так,
digitalWrite(13, 1);      // а можно и так
```

Настройка цифровых портов на ввод (INPUT) и вывод (OUTPUT) сигналов

```
#define INPUT 0x0
#define OUTPUT 0x1
```

ВНИМАНИЕ:

- Порты поддерживают положительное или отрицательное направление тока до 40 мА
- Порты, сконфигурированные как выводы, могут быть повреждены, если их замкнуть накоротко на «землю» (общая шина питания), на источник питания +5 В, или подсоединить к мощной нагрузке с малым сопротивлением.

Пример:

```
pinMode(13, OUTPUT);  // 13-й вывод будет выходом
pinMode(12, INPUT);   // а 12-й – входом
```

Цифровой ввод/вывод

Функции Arduino для цифрового ввода-вывода объявлены в файле \hardware\cores\arduino\wiring.h, а реализованы в \hardware\cores\arduino\wiring_digital.c.

```
void pinMode(uint8_t, uint8_t);
```

Вызов: pinMode (порт, режим), где порт – значение целого типа от 0 до 13; режим – либо INPUT (ввод) либо OUTPUT (вывод).

Примечание: Аналоговые входы могут использоваться как цифровые входы/выходы, при обращении к ним по номерам с 14 по 19 (аналоговый вход 0 .. 5)

```
void digitalWrite(uint8_t, uint8_t);
```

Вызов: digitalWrite(порт, значение), где порт: номер порта; значение: HIGH или LOW.

Пример:

```
digitalWrite(13, HIGH);  // выставляем 13-й вывод в «высокое» состояние
```

```
int digitalRead(uint8_t);
```

Вызов: value = digitalRead (порт); Считывает значение на указанном порту, возвращает текущее значение на порту (HIGH или LOW) типа int

Пример:

```
int val;
val = digitalRead(12);  // опрашиваем 12-й вывод
```

Аналоговый ввод/вывод сигнала

int analogRead(uint8_t);

Вызов: value = analogRead(порт); Считывает значение с указанного аналогового порта. (Arduino содержит 6 каналов АЦП по 10 бит каждый. Входному напряжению от 0 - 5В соответствует возвращаемое число АЦП типа int от 0 до 1023). Требуется приблизительно 0.1 мс, чтобы считать значение аналогового ввода.

Пример:

```
int val;
val = analogRead(0);  // считываем значение на 0м аналоговом входе
```

Примечание: Аналоговые порты по умолчанию определенны на ввод сигнала и в отличие от цифровых портов их не требуется конфигурировать с помощью вызова функции pinMode.

void analogWrite(uint8_t, int);

Вызов: analogWrite(порт, значение); Выводит на порт аналоговое значение. Эта функция работает на: 3, 5, 6, 9, 10, и 11 цифровых портах Arduino в режиме ШИМ ~490 Гц в интервале 0 .. 255.

Пример:

```
analogWrite(9, 128);  // устанавливаем на 9 контакте значение ШИМ эквивалентное 2,5В
```

Примечание: Нет необходимости вызвать функцию pinMode, чтобы установить порт на вывод сигналов перед вызовом функции analogWrite.

Платформа Arduino имеет внутренний источник опорного напряжения (ИОН) для АЦП. Через резистор 5k к порту AREF можно подключить и внешний источник. Функция void analogReference(uint8_t mode); определяет источник опорного напряжения для АЦП.

Дополнительные функции ввода/вывода сигнала

void shiftOut(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, byte val); Сдвиг байта данных по одному биту за раз.

unsigned long pulseIn(uint8_t pin, uint8_t state, unsigned long timeout); Считывает импульс (высокий или низкий) с цифрового порта и возвращает продолжительность импульса в микросекундах, импульс ожидается в течении таймаута (если не указать — будет ждать 1 сек.)

Работа со временем

unsigned long millis(void);

Вызов: time = millis(); Возвращает число миллисекунд, с момента исполнения Arduino текущей программы. Счетчик переполнится и обнулится приблизительно через 1193 часов или же 50 дней (49.7).

Пример:

```
unsigned long time; // объявление переменной time типа unsigned long
time = millis();    // передача количества миллисекунд
```

unsigned long micros(void); Возвращает число микросекунд, с момента исполнения Arduino/Freeduino текущей программы. Переменная переполнится (обнулится), приблизительно через 70 минут.

Примечание: На 16 МГц плате Arduino, данная функция работает с разрешением в 4 микросекунды.

Пример:

```
time = micros();
Serial.println(time);  // выводим число микросекунд с момента запуска программы
```

void delay(unsigned long);

Вызов: delay(время_мс); Приостанавливает программу на заданное число миллисекунд.

Пример:

```
delay(1000);  //пауза 1 секунда
```

void delayMicroseconds(unsigned int us);

Вызов: delayMicroseconds(время_мкс); Приостанавливает программу на заданное число микросекунд.

Математические функции

min(x, y)	Возвращает меньшее из двух чисел

Пример:

```
sensVal = min(sensVal, 100); // sensVal не меняет значения, если он меньше 100
                             // получается, что sensVal не сможет превысить 100.
```

max(x, y)	Возвращает большее из двух чисел
abs(x)	Возвращает модуль (абсолютную величину) числа
constrain(x, a, b)	Проверяет находится ли число x в диапазоне (a,b) Возвращаемое значение: x: если x лежит между a и b a: если x меньше, чем a b: если x больше, чем b
map(value, fromLow, fromHigh, toLow, toHigh)	Отображает число из одного диапазона в другой. Оперирует целыми числами

Пример:

```
y = map(x, 1, 50, 50, 1);
y = map(x, 1, 50, 50, -100);
/* отображает аналоговое значение к 8 битам (диапазон от 0 до 255) */
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(9, val);
```

pow(base, exponent) Функция предназначена для возведения числа в заданную степень.

sq(x) Вычисляет квадрат числа: число умноженное на себя.

sin(rad) Вычисляет синус угла (в радианах). Результат будет между -1 и 1.

cos(rad) Вычисляет косинус угла (в радианах). Результат будет между -1 и 1.

tan(rad) Вычисляет тангенс угла (в радианах). Результат будет между плюс и минус бесконечностью :)

Псевдослучайные числа

void randomSeed(unsigned int seed) инициализирует генератор псевдослучайных чисел

Пример:

```
long randNumber;
void setup(){
  Serial.begin(9600);
  randomSeed(analogRead(0));
}

void loop(){
  randNumber = random(300);
  Serial.println(randNumber);
  delay(50);
}
```

long random(long howbig)

long random(long howsmall, long howbig) Генерирует псевдослучайное число

Последовательная передача данных

Arduino имеет встроенный контроллер для последовательной передачи данных, который может использоваться как для связи между Arduino устройствами, так и для связи с компьютером. На компьютере соответствующее соединение представлено USB COM-портом, который появляется в системе после установки необходимого драйвера.

Связь происходит по цифровым портам 0 и 1, и поэтому Вы не сможете использовать их для цифрового ввода/вывода если используете функции последовательной передачи данных.

Serial.begin(long);

Вызов: Serial.begin(скорость_передачи); Устанавливает скорость передачи COM порта: 300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, или 115200 бит в секунду.

Serial.available(void);

Вызов: count = Serial.available(); Функция возвращает количество накопленных в буфере микроконтроллера байт принимаемых по последовательному порту. Последовательный буфер может хранить до 128 байт.

Пример:

```
if (Serial.available() > 0) { // Если в буфере есть данные
   // здесь должен быть прием и обработка данных
```

```
}
```

Serial.read(void);

Вызов: char = Serial.read(); Считывает следующий байт из буфера последовательного порта. Возвращаемое значение: первый доступный байт входящих данных с последовательного порта, или -1 если нет входящих данных.

Serial.write(uint8_t c)

Вызов: Serial.write(val); // val: переменная для передачи, как единственный байт

Serial.write(str); // str: строка для передачи -последовательность байт

Serial.write(buf, len); Записывает данные в последовательный порт. Данные посылаются как байт или последовательность байт; для отправки символьной информации следует использовать функцию print(). buf: массив для передачи, как последовательность байт, len: длина массива

Serial.flush(void)

Вызов: Serial.flush(); Очищает входной буфер последовательного порта.

Serial.print() Вывод данных на последовательный порт в ASCII кодах.

Функция имеет несколько форм вызова в зависимости от типа и формата выводимых данных.

Пример:

Serial.print(b, format); b – число, format – формат выводимого числа:

DEC - десятичное представление числа b.
HEX - шестнадцатеричное представление числа b.
OCT - восьмеричное представление числа b.
BIN - двоичное представление числа b.
BYTE - выводит младший байт числа b.

Пример

```
int b = 79;
Serial.print(b, HEX); //выдаст в порт строку «4F»
```

Serial.print(str) если str – строка или массив символов, побайтно передает str на COM-порт.

Пример

```
char bytes[3] = {79, 80, 81};    //массив из 5 байт со значениями 79,80,81
Serial.print("Here our bytes:"); //выводит строку «Here our bytes:»
Serial.print(bytes);             //выводит 3 символа с кодами 79,80,81 –
                                 //это символы «OPQ»
```

Serial.print(b) если b имеет тип byte или char, выводит в порт само число b.

Serial.println() выводятся символ возврата каретки (ASCII 13, или '\r') и символ новой линии (ASCII 10, или '\n').

Пример

```
int b = 79;
Serial.print(b, DEC); //выдаст в порт строку «79»
Serial.print("\r\n"); //выведет символы "\r\n" – перевод строки
Serial.println(b, DEC); //выдаст в порт строку «79\r\n»
```

Прерывания

Прерывание (англ. interrupt) — сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается, и управление передаётся обработчику прерывания, который выполняет работу по обработке события и возвращает управление в прерванный код.

Функции Arduino для работы с прерываниями объявлены в файле \hardware\cores\arduino\wiring.h и реализованы в файле \hardware\cores\arduino\WInterrupts.c

void attachInterrupt(uint8_t, void (*)(void), int mode); Определяет, какую функцию вызывать, когда происходит внешнее прерывание. Замещает предыдущую функцию, если таковая была привязана к данному прерыванию. Большинство плат Arduino имеют два внешних прерывания с номерами 0 (на digital pin 2) и 1 (на digital pin 3). Arduino Mega имеет дополнительно ещё четыре: с номерами 2 (pin 21), 3 (pin 20), 4 (pin 19) и 5 (pin 18).

Вызов: attachInterrupt(interrupt, function, mode); где interrupt: номер прерывания (int); function: функция, которая должны вызываться при прерывании. Функция не должна принимать параметров и не должна ничего возвращать. mode: определяет, когда должно сработать прерывание:

LOW — вызов прерывания всякий раз, когда на порту низкий уровень напряжения;
CHANGE – прерывание вызывается при изменении значения на входе;
RISING – вызов прерывания при изменении уровня напряжения с низкого (LOW) на высокое(HIGH)
FALLING – вызов прерывания при изменении уровня напряжения с высокого (HIGH) на низкое (LOW)

Пример:

```
// светодиод, подключённый к digital pin 13 будет изменять своё
// состояние при изменении напряжения на digital pin 2
//
int pin = 13;
volatile int state = LOW; // переменная которая не оптимизируется компилятором

void setup()
{
  pinMode(pin, OUTPUT);             // порт как выход
  attachInterrupt(0, blink, CHANGE); // привязываем 0-е прерывание к функции blink().
}

void loop()
{
  digitalWrite(pin, state);         // выводим state
}

void blink()
{
  state = !state;                   // меняем значение на противоположное
}
```

void detachInterrupt(uint8_t); Отключает указанное прерывание.

Вызов: detachInterrupt(interrupt); где interrupt: номер прерывания для отключения (0 или 1).

Энергонезависимая память EEPROM http://robocraft.ru/blog/arduino/82.html

EEPROM — (Electrically Erasable Programmable Read-Only Memory) электрически стираемое перепрограммируемое ПЗУ, ЭСППЗУ). Память такого типа может стираться и заполняться данными несколько десятков тысяч раз. Используется в твердотельных накопителях. Одной из разновидностей EEPROM является флеш-память (Flash Memory).

byte EEPROM.read(address) Считывает байт из энергонезависимой памяти EEPROM. Если байт до этого никогда не перезаписывался – вернёт значение 255. address: порядковый номер ячейки памяти для чтения — от 0 до 511 (int)

Пример (File-Examples-EEPROM-eeprom_read):

```
/*
        * Чтение EEPROM
        *
        * Считывает значения всех байтов энергонезависимой памяти
        * EEPROM и выводит их в COM-порт
        */

        #include <EEPROM.h>

        // начальный адрес памяти EEPROM
        int address = 0;
        byte value;

        void setup()
        {
          Serial.begin(9600);
        }

        void loop()
        {
          // считываем значение по текущему адресу EEPROM
          value = EEPROM.read(address);

          Serial.print(address);
          Serial.print("\t");
          Serial.print(value, DEC);
          Serial.println();

          // устанавливаем следующую ячейку памяти
          address = address + 1;

          // EEPROM содержит всего 512 байт: от 0 до 511, поэтому
          // если адрес достиг 512, то снова переходим на 0
          if (address == 512)
            address = 0;

          delay(500);
        }
```

void EEPROM.write(address, value) Записывает байт в энергонезависимую память. address: порядковый номер ячейки памяти для записи — от 0 до 511 (int); value: байт для записи – от 0 до 255 (byte)

Примечание: Документация (datasheet) на микроконтроллеры Atmega8/168 говорит, что возможное количество циклов перезаписи данных в памяти ограничено 100000 раз (Write/Erase Cycles). Время, требуемое для завершения цикла записи составляет 3.3 ms. Данная задержка уже учитывается библиотекой EEPROM, поэтому в дополнительном вызове delay() нет необходимости.

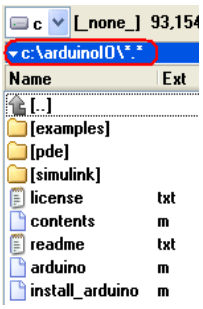
Очистка памяти производится записью нулей.

Создание своей библиотеки. http://robocraft.ru/blog/arduino/102.html

ПОДКЛЮЧЕНИЕ ARDUINO К MATLAB http://robocraft.ru/blog/741.html

Подключение Arduino к MATLAB (R2008a или более поздней версии) выполняйте в следующей последовательности:

1. Скачайте пакет ArduinoIO с официального сайта MathWorks: http://www.mathworks.com/matlabcentral/fileexchange/32374
2. Распакуйте пакет, например, в c:\arduinoIO:



3. Загрузите МатЛАБ
4. Выполните команды

   >>cd c:\arduinoIO % переход в директорию

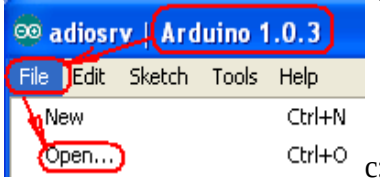
   >>install_arduino % установка

   Arduino folders added to the path

   Saved updated MATLAB path

   >>savepath

5. Запишите в платформу Arduino прошивку-ретранслятор. Ее задача принимать и выполнять команды из порта. Один вариант прошивки поддерживает только функции работы с аналоговыми и цифровыми портами, другой - еще и шаговые двигатели. Для последнего требуется специальная библиотека, которую можно взять в https://github.com/adafruit/Adafruit-Motor-Shield-library/zipball/master. Ее требуется распаковать в папку Arduino-XXX/libraries/ Открываем файл



c:\arduinoIO\pde\adiosrv\adiosrv.pde или c:\arduinoIO\pde\srv\srv.pde из среды Arduino и прошиваем  .

6. В консоле MATLAB создайте объект: >> a=arduino('COM3')
7. Проверьте правильность подключения следующими командами обращения МатЛАБ к Arduino

```
>>a.pinMode(13,'OUTPUT')
>> a.digitalWrite(13,1) % запись в порт “1”, на плате должен загореться светодиод
>> a.digitalWrite(13,0) % запись в порт “0”, диод на плате должен погаснуть
>>foo=a.analogRead(1)        % чтение с аналогового входа, при открытом входе foo > 0
```

ПОДКЛЮЧЕНИЕ ARDUINO К SIMULINK (http://robocraft.ru/blog/741.html)

Для подключения Arduino к Simulink (+SimulinkCoder + EmbeddedCoder) MATLAB (R2010a) или более поздней версии выполните следующие шаги:

1. Скачайте пакет SimulinkSupportPackageforArduino
   http://www.mathworks.com/matlabcentral/fileexchange/30277
2. Распакуйте пакет, например в c:\arduino_simulink
3. Загрузите MatLAB
4. Выполните следующие команды для добавления путей:

```
>>cd c:\arduino_simulink
>> addpath(fullfile(pwd,'arduino'),fullfile(pwd,'blocks'),fullfile(pwd,'demos'))
>>savepath
```

5. Обновите кастомизацию:

```
>>sl_refresh_customizations
```

6. Подключите платформу Arduino к USB порту компьютера .
7. Укажите путь к среде arduino

```
>>arduino.Prefs.setArduinoPath('c:\ArduinoTarget')
```

8. Выведите список всех доступных платформ

```
>> arduino.Prefs.setBoard

    Specify one of the following board labels:
        'uno' (Arduino Uno)
        'mega2560' (Arduino Mega 2560)
        'mega' (Arduino Mega (ATmega1280))
```

9. Задайте текущую платформу 'uno' командой

```
>>arduino.Prefs.setBoard('uno')
```

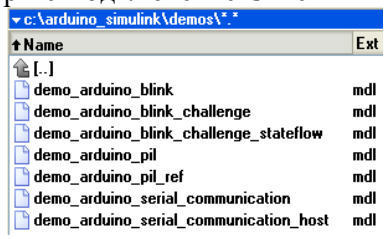
10. Проверьте доступные порты:

```
>>comPorts=arduino.Prefs.searchForComPort
```

    В результате получите, например, comPorts ='COM3'

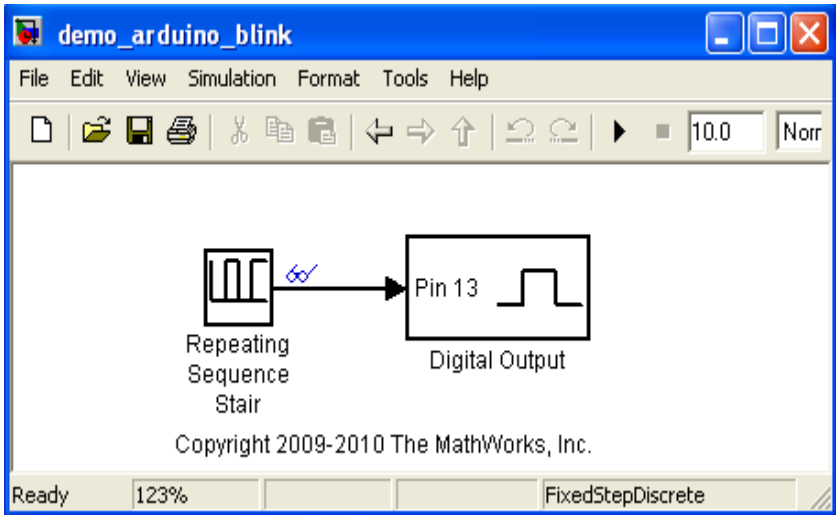
11. Подключите Arduino к COM порту:

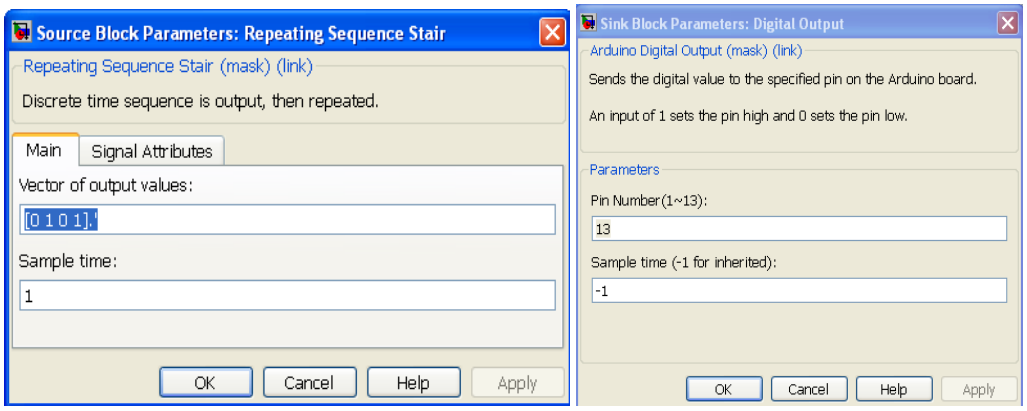
```
>>arduino.Prefs.setComPort('COM3')
```

12. Проверьте подключение Simulink к Arduino, по работе демонстрационных примеров папки 

для этого:

12.1 Откройте первый пример Simulink модели командой >>demo_arduino_blink и установите параметры модели:

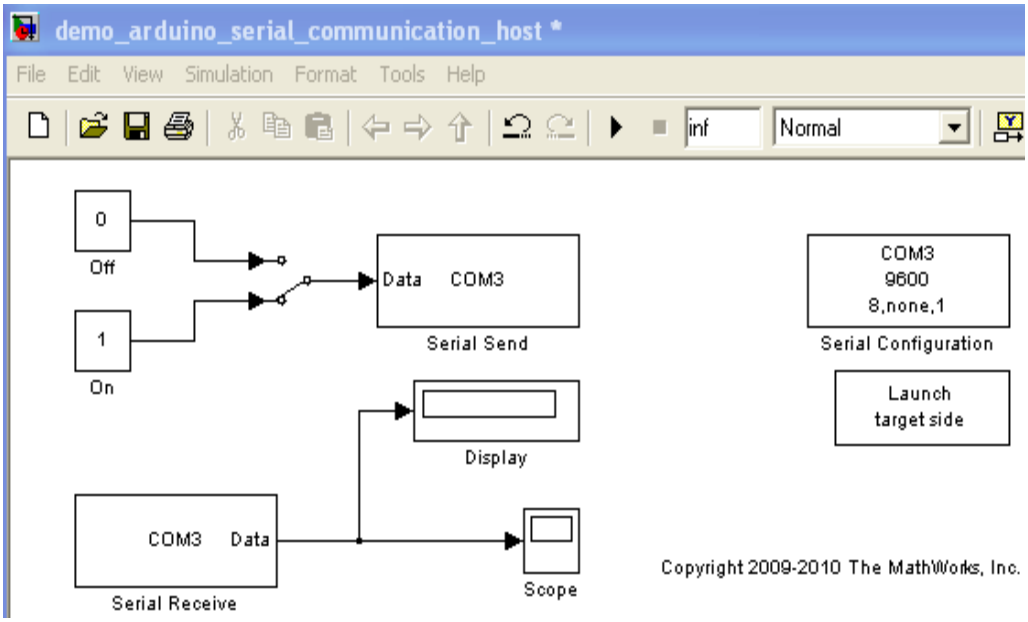




12.2 Комбинацией Ctrl+B или командой Tools>>Codegeneration>>BuildModel запустите кросс-компилятор.

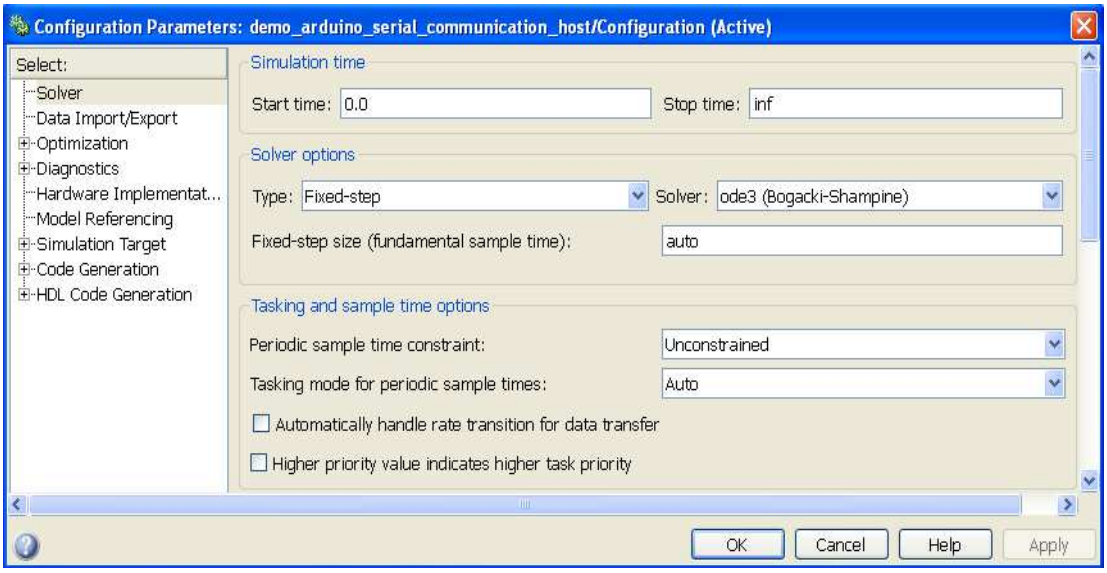
12.3 Нажмите кнопку “Start simulation”  код платформы прошивается и выполняется в Arduino. Контрольный светодиод Arduino (pin 13) начинает мигать.

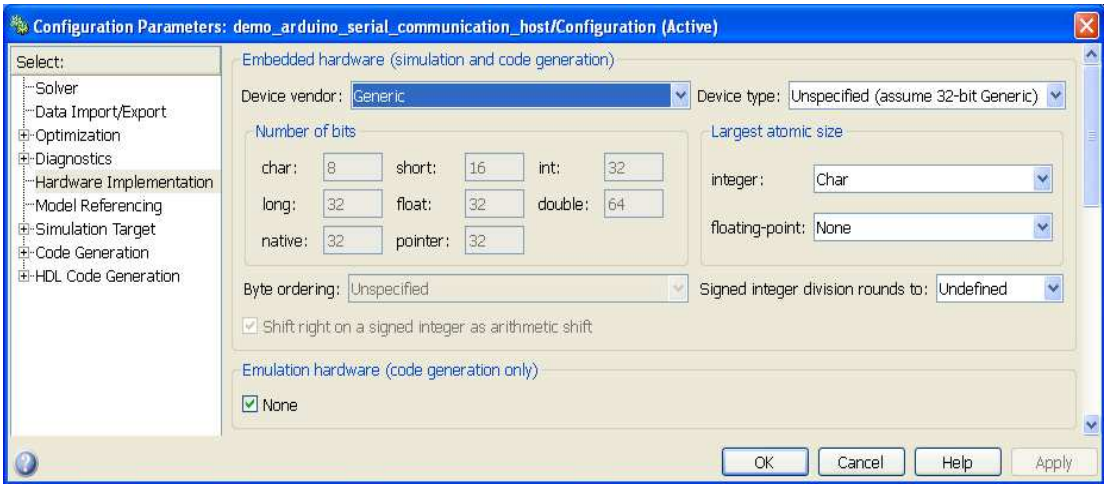
12.4 Для ручного управления светодиодом Arduino через Simulink и наблюдением за входом Pin 2 АЦП Arduino откройте другую модель demo_arduino_serial_communication_host.mdl:

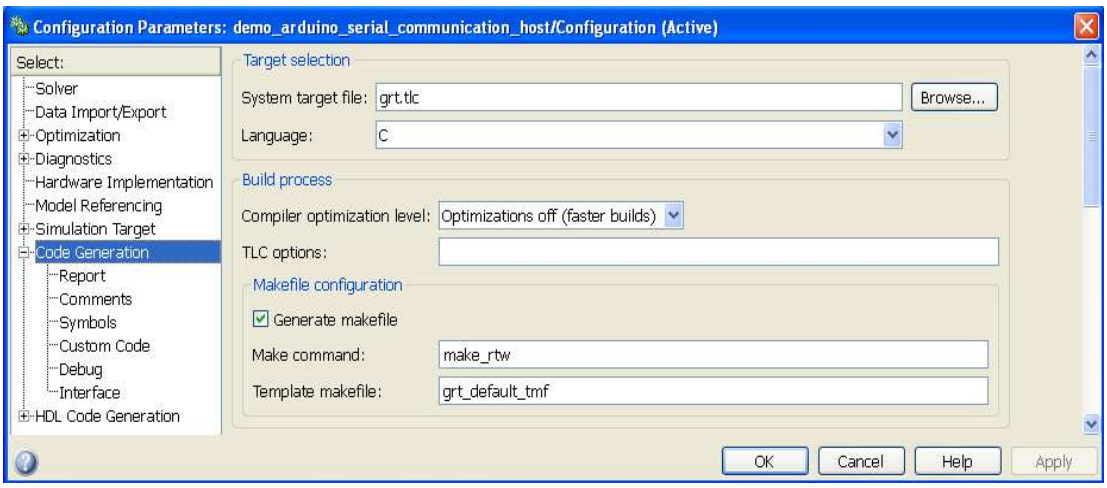


В блоке “Lunch target side” этой модели Simulink содержится код Arduino который загружается в него перед запуском модели.

Модель Simulink имеет следующие параметры.

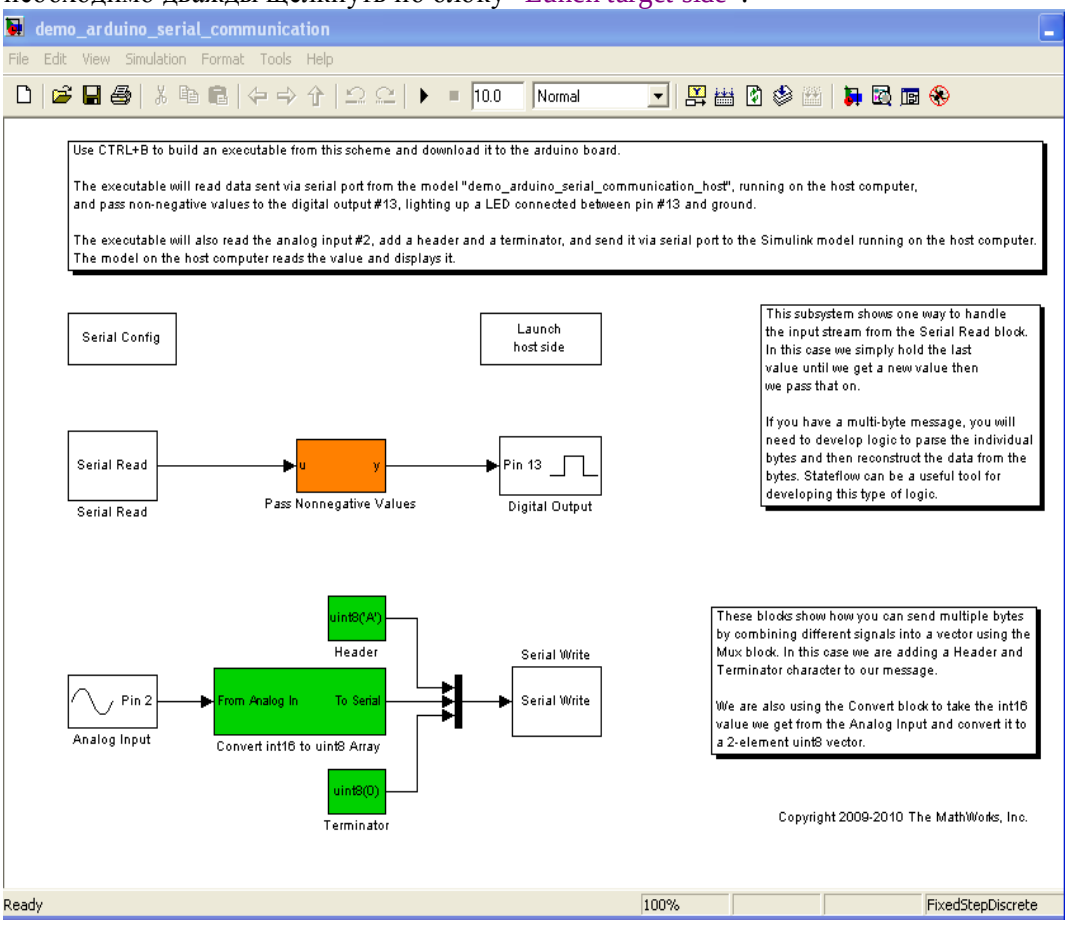






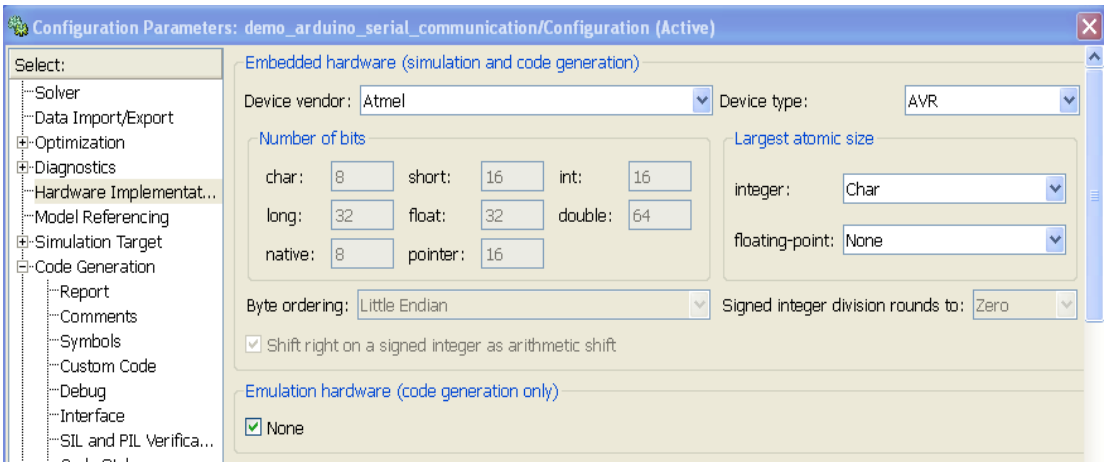
12.5 Настройте номера COM порта модели на номер COM порта платформы Arduino.

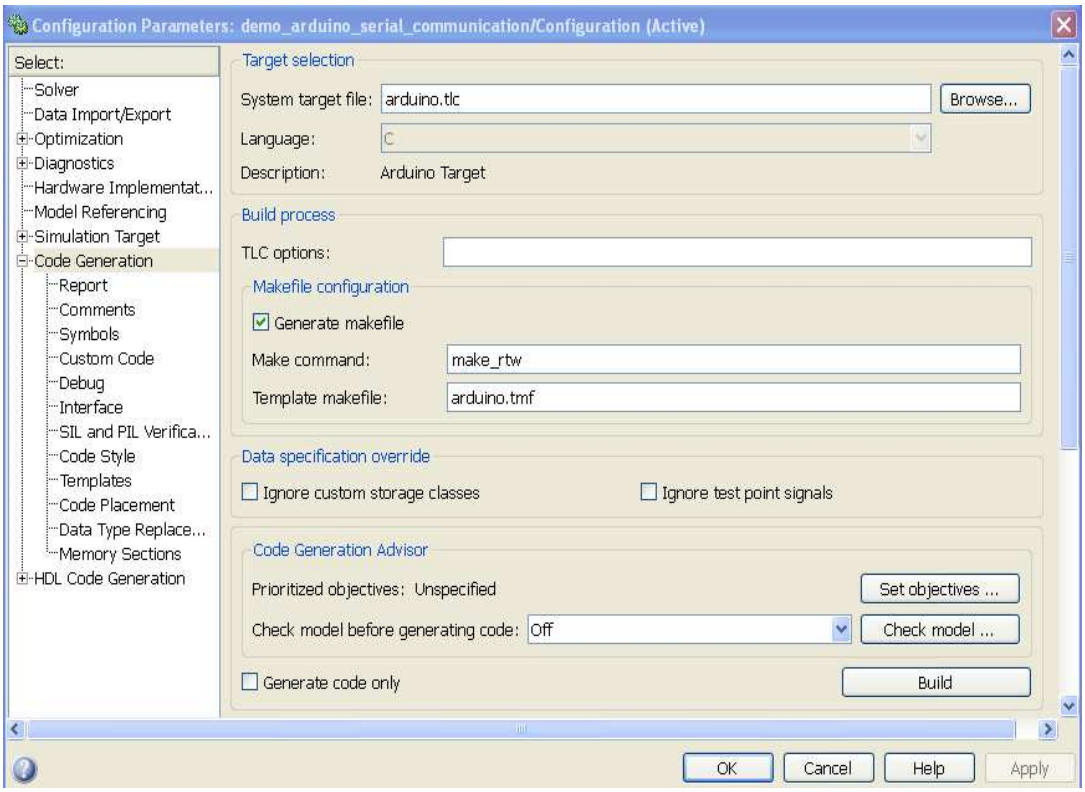
12.6 Для изучения, редактирования или компиляции кода прошиваемого в Arduino необходимо дважды щелкнуть по блоку “Lunch target side”.



Обратный переход в модель Simulink можно выполнить через активацию блока “Lunch host side” окна кода прошиваемого в Arduino.

Для успешной компиляции кода Arduino необходимо настроить следующие параметры.



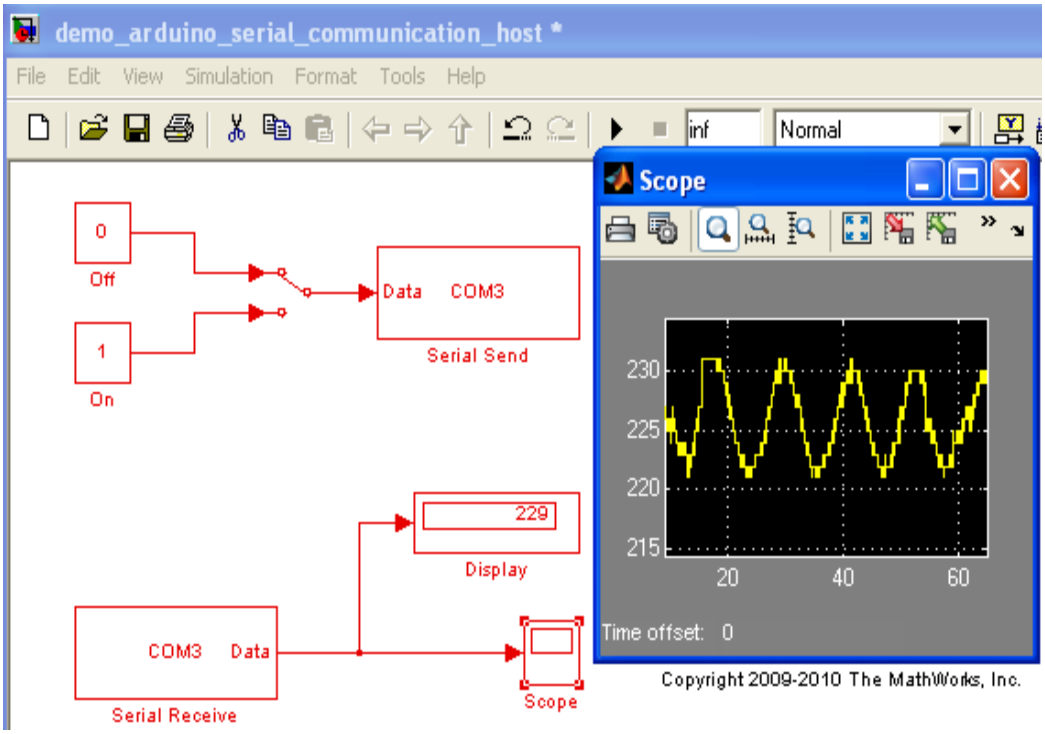


12.7 Комбинацией Ctrl+B или командой Tools>>Codegeneration>>BuildModel запустите кросс-компилятор.

Успешная компиляция сопровождается выводом следующего сообщения в командном окне.

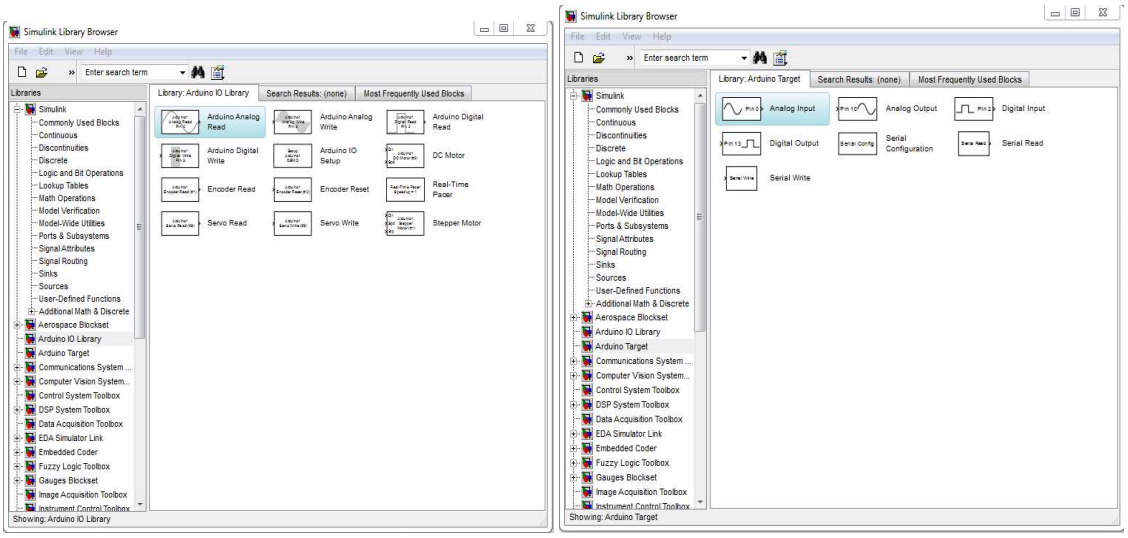
Successful completion of build procedure for model: demo_arduino_blink

12.8 Нажатием клавиши меню  загрузите код в Arduino и запустите Simulink модель на выполнение.



12.9 Поверьте возможность переключения светодиода Pin 13 arduino и отображение кода АЦП Pin 2 при помощи demo_arduino_serial_communication_host.mdl модели

13. Проверьте наличие разделов Arduino (Arduino IO Library и Arduino Target) в библиотеке Simulink:



Эти блоки могут быть использованы для написания кода прошиваемого в Arduino.

ПОДКЛЮЧЕНИЕ ARDUINO К LabVIEW

Для работы с платформой Arduino в LabView установите NI LabVIEW Interface for Arduino Toolkit http://sine.ni.com/nips/cds/view/p/lang/ru/nid/209835

Этот пакет имеет следующие особенности:

- Обеспечивает доступ к цифровым и аналоговым портам, ШИМ, интерфейсам I2C и SPI
- Обеспечивает работу с двигателями под управлением программ загруженных на Arduino

- Имеет примеры для решения разнообразных задач и ввода данных с датчиков
- Поддерживает работу с Arduino через USB, serial, Bluetooth или XBee
- Частота USB петли - 200 Hz, беспроводной - 25 Hz

ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Программирование Arduino в интегрированной среде.

1. Запустите Arduino IDE –C:\arduino-1.0.3\arduino.exe
2. Настройте номер COM-порта среды (Tools > Serial port) на работу с COM портом Adruino из списка устройств компьютера
3. Выберите тип платы Tools > Board > Arduino UNO
4. Откройте  пример программы повторяющегося включения / выключения светодиода File > Open > c:\arduino-1.0.3\examples\01.Basics\Blink\Blink.ino и запустите  его.
5. По миганию светодиода убедитесь, что платформа Arduino работает.
6. Рассмотрите код программы.

ВОПРОСЫ:

- Каков размер прошитой программы?
- Каков размер памяти программ Arduino?
- Какова частота мигания светодиода?

7. Откройте и запустите пример File > Open > c:\arduino-1.0.3\examples\01.Basics\ AnalogReadSerial\ AnalogReadSerial.ino. Пример читает аналоговый вход N0 и выводит результат в канал последовательной передачи, связанный с COM портом компьютера.
8. Запустите программу COM Port Toolkit. Обратите внимание на поток принимаемых данных.
9. Сравните форматы следующих команд Arduino, передающие данные в канал COM порта компьютера: println; print, write.
10. Осторожно подсоедините аналоговый вход к земле.
11. Запишите значения считываемые из COM порта.
12. Не выключая COM Port Toolkit попробуйте снова выполнить программу Blink.ino
13. Закройте COM Port Toolkit и попробуйте снова выполнить программу Blink.ino

Задание 2. Связь платформы Arduino с МатЛАБ

1. Разработайте программу (m-код MatLAB) приема и отображения данных, принимаемых от Arduino через COM порт.

2. Разработайте программу (m-код MatLAB) переключения светодиода Arduino (pin 13).

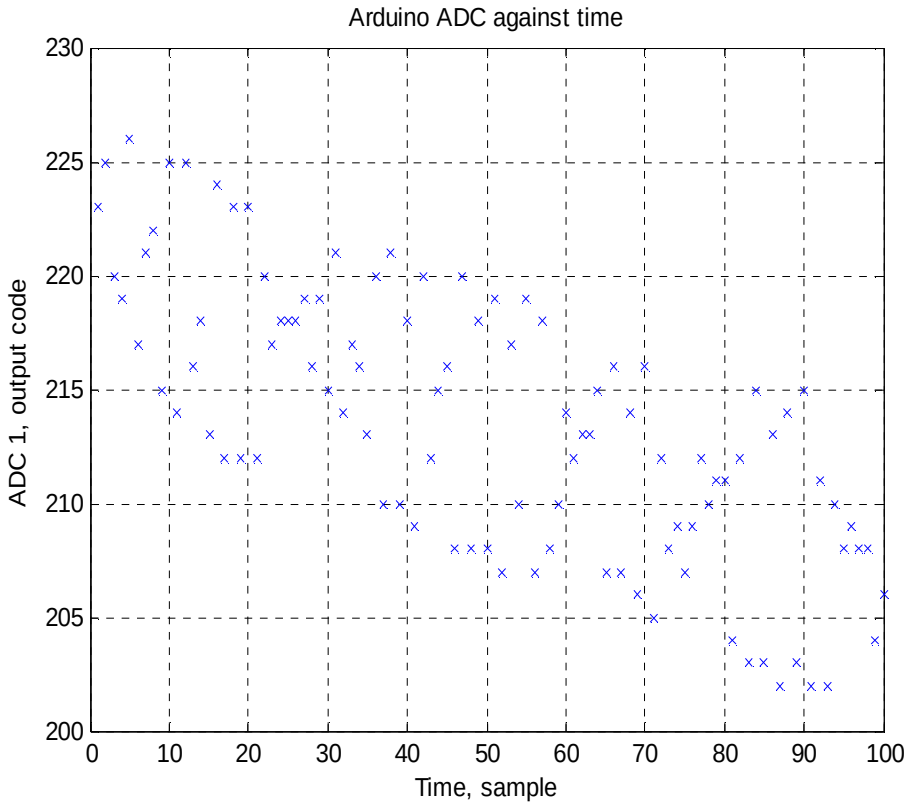
Задание 3. Управление портом Arduino из Simulink

1. Разработайте Simulink модель приема и отображения данных, принимаемых от Arduino через COM порт.
2. Разработайте Simulink модель переключения светодиода Arduino (pin 13).

Задание 4. Ввод и отображение в МатЛАБ аналогового сигнала платформы Arduino

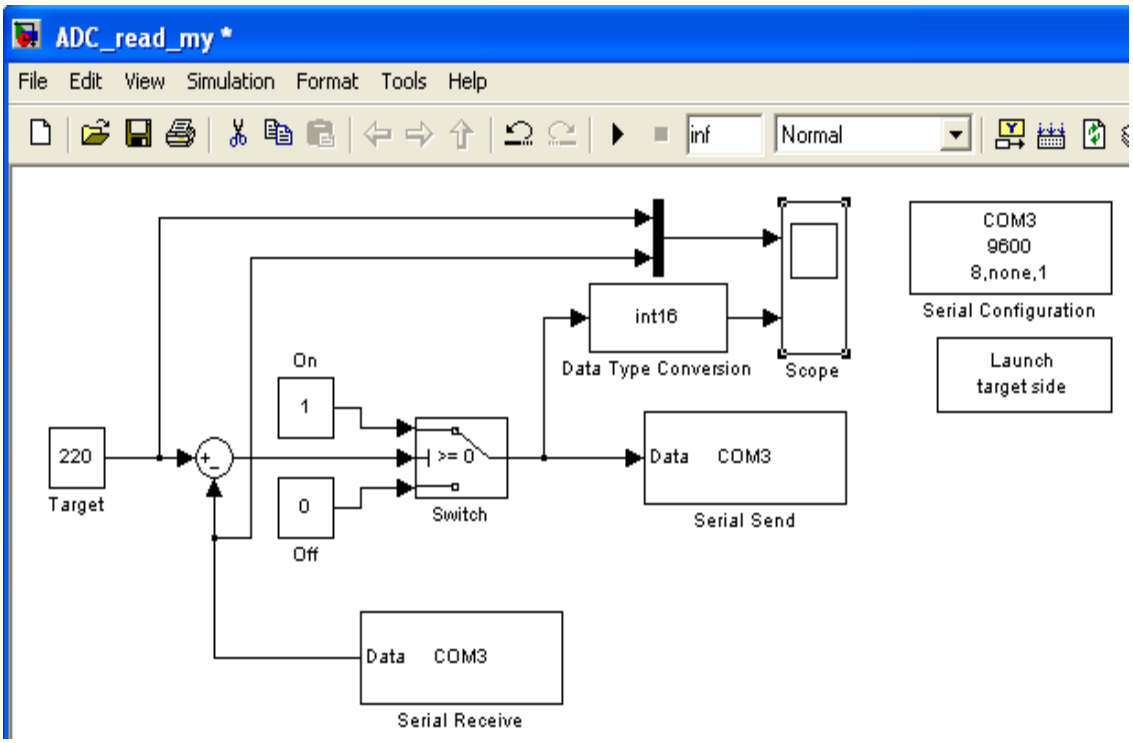
1. Подключите Arduino к MatLAB (R2008a или более поздней версии) как показано выше в разделе “Общие сведения”.
2. В среде MatLAB постройте осциллограф аналогового сигнала АЦП 1 платформы Adruino. Для решения задачи могут пригодиться следующие команды МатЛАБ: for, plot, pause, a.analogRead(1). Пример MatLAB кода и график результата показан ниже.

```
figure
for i = 1:100
    plot(i,a.analogRead(1),'xb');
    hold on;
end
grid
xlabel ('Time, sample');
ylabel ('ADC 1, output code');
title ('Arduino ADC against time')
```

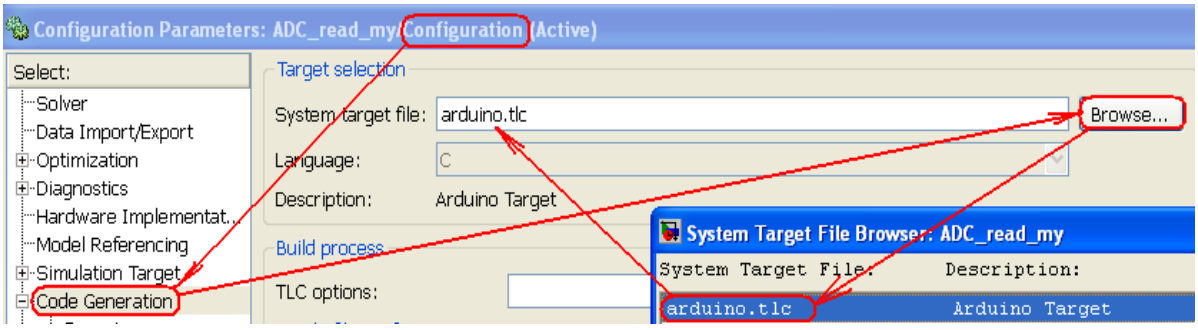


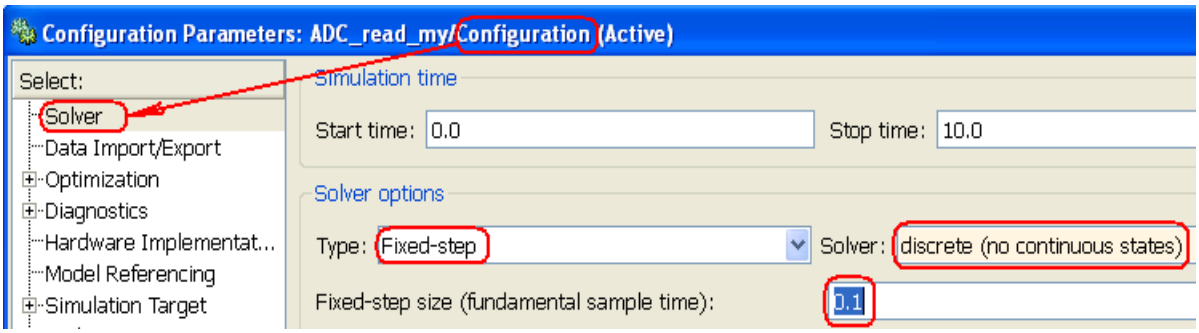
Задание 5. Создание интерактивной среды МатЛАБ – Simulink на базе платформы Arduino

1. Подключите Arduino к Simulink MatLAB (R2010a или более поздней версии) как показано выше в разделе “Общие сведения”.
2. Прошейте в Arduino модель simulink demo_arduino_serial_communication.mdl
3. Используя блоки библиотеки Simulink и модели demo_arduino_serial_communication_host.mdl соберите модель управления светодиодом с обратной связью, показанную ниже.

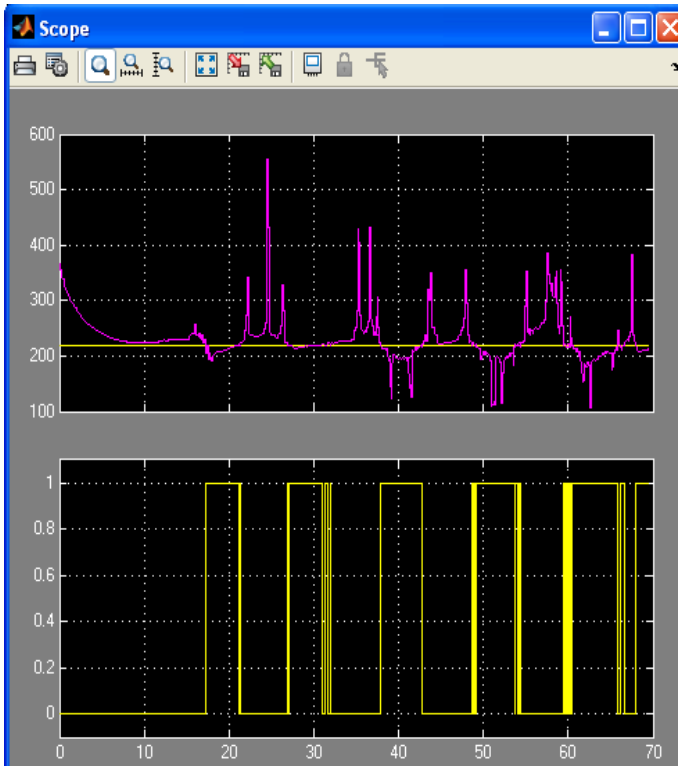


4. Настройте конфигурационные параметры вашей модели:





5. Запуская модель  и наблюдая за состоянием обратной связи, настройте значение блока Target так, чтобы светодиод включался когда уровень АЦП ниже заданного воздействия и выключался в противном случае, например, как показано ниже. Значение сигнала обратной связи можно изменять приближением / удалением пальца руки ко входу АЦП платформы Arduino.



6. Отсоедините USB кабель от Arduino.
7. Вместе с преподавателем подключите датчик температуры LM35 (5V – красный провод, выход 10 мВ/С - зеленый провод, земля – синий провод) к платформ Arduino.

LM35:



8. Доработайте последнюю модель:
9. Подключите Arduino USB кабелем к компьютеру.
10. Запустите модель.
11. Изменяя температуру на датчике убедитесь в работоспособности среды Simulink – платформа Arduino.

Задание 6. Построение системы реального времени Simulink – Arduino на базе таймера Arduino

1. Используя следующие примеры постройте систему, которая работает в режиме реального времени с тактом 0.1 сек. Система ручным переключателем модели Simulink управляет состоянием светодиода платформы (Pin 13), принимает код переключателя от Arduino в Simulink и отображает его.

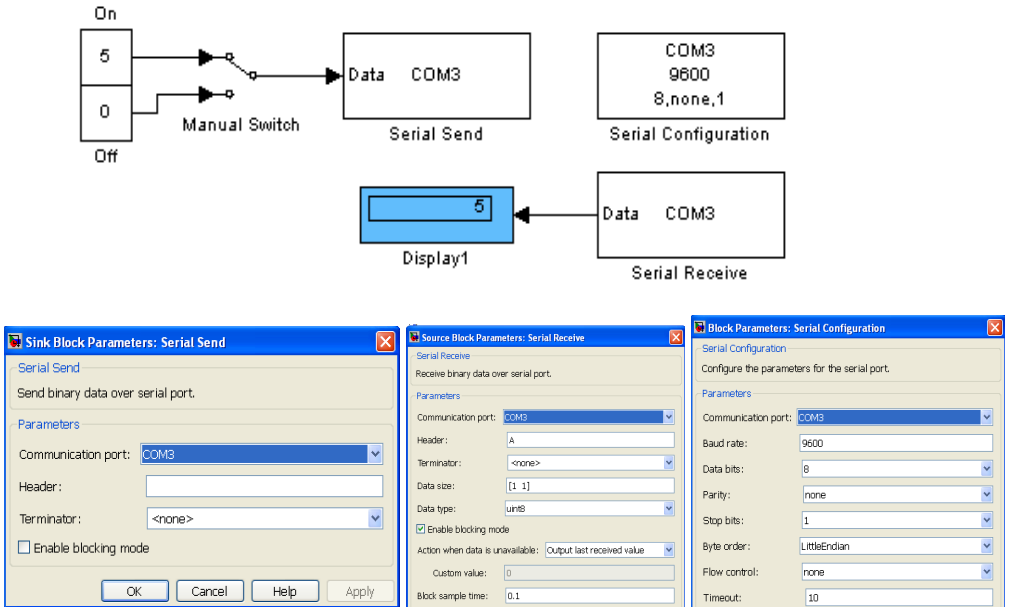
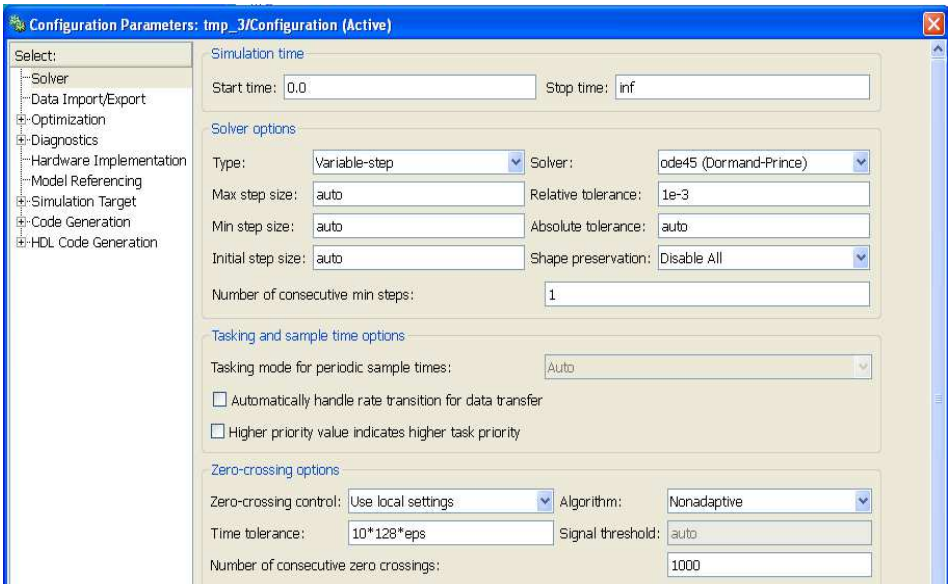


Рис. 3. Модель Simulink и параметры блоков модели.



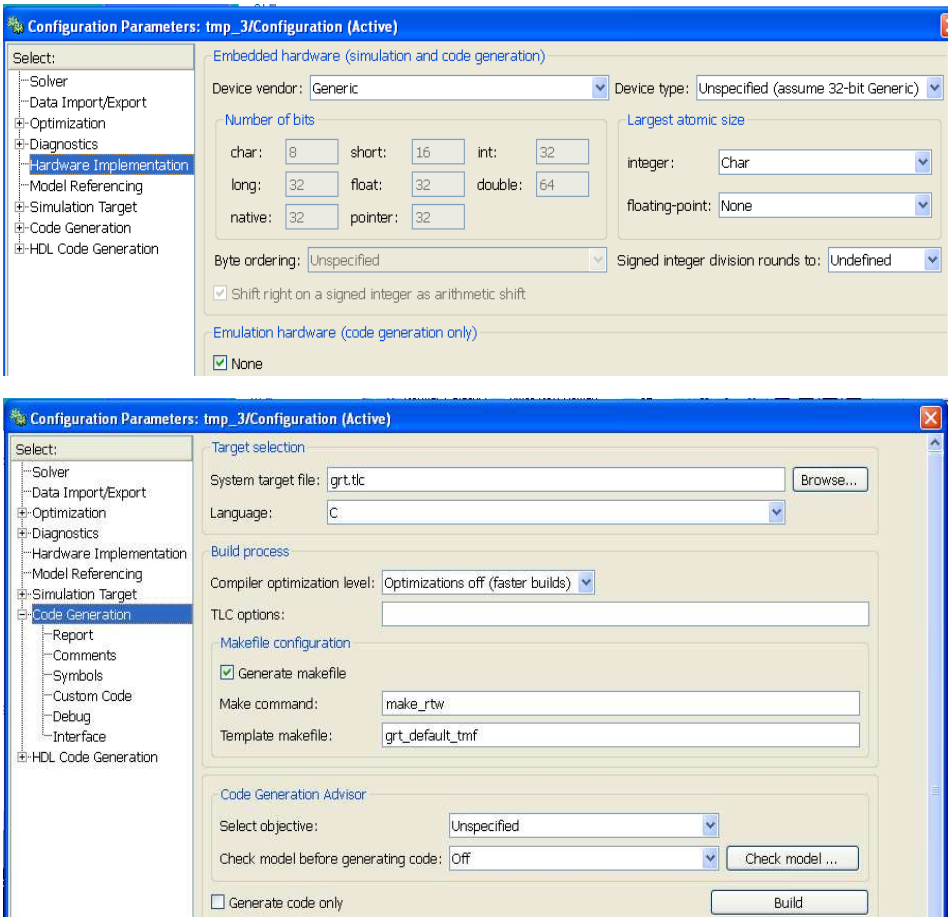


Рис. 4. Параметры конфигурации Модели Simulink.

Код RT программы Arduino.

```
/*
 Communication with Simulink model in RT mode

 created 20 April 2013
 modified
 by Bob Davidov

 This example code is in the public domain.
 */
int led = 13;
unsigned long set_time = 0;
int thisChar = 14;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  Serial.flush();  // clear input buffer
  pinMode(led, OUTPUT);
}

void loop() {
```

```
  // get any incoming bytes:
  if (Serial.available() > 0) {
  thisChar = Serial.read();
  if (thisChar) digitalWrite(led, HIGH);
    else digitalWrite(led, LOW);
  }
  unsigned long time = millis();
  if (time > set_time) {
    set_time = set_time + 100;
    Serial.print("A"); // DEC, HEX, OCT, BIN, BYTE //8.599E+8
    Serial.write(thisChar);
//    Serial.write(12); // LOWBYTE
//    Serial.write(34); // HIGH BYTE
  }
}
```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как обеспечить связь среды проектирования систем управления MatLAB и Smulink с внешней средой через платформу Ardulino ?
2. Как построить реальную систему термостатирования релейного типа на базе MatLAB, MatLAB-Smulink и платформы Ardulino ?
3. Какова скорость считывания АЦП платформы Arduino?
4. Какова максимальная частота контурного управления на базе платформы Arduino?
5. Почему при работающей программе COM Port Toolkit невозможно перепрограммировать Arduino?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. MathWorks. Arduino Support from MATLAB. http://www.mathworks.com/academia/arduino-software/arduino-matlab.html.
2. Arduino, site: http://www.arduino.cc/
3. Site RoboCraft: http://robocraft.ru/blog/projects/318.html
4. Site DFRobot: http://www.dfrobot.com/index.php
5. Dr. Bob Davidov. Компьютерные технологии управления в технических системах http://portalnp.ru/author/bobdavidov.

Dr. Bob Davidov

Кафедра Автоматики и процессов управления Санкт-Петербургского государственного электротехнического университета “ЛЭТИ”

Многоканальное устройство ввода и накопления аналоговых данных на базе MS Excel

Цель работы: познакомиться с macros технологиями для решения задач управления и мониторинга в неспециализированной среде MS Excel.

Задача работы: построение системы ввода, обработки и накопления аналоговых данных на базе электронных таблиц и USB/RS-232 интерфейса Arduino.

Приборы и принадлежности: Персональный компьютер, MS Excel, платформа Arduino UNO, МатЛАБ.

Устройство было разработано и проверено в среде Windows XP.

ВВЕДЕНИЕ

Электронные таблицы MS Excel можно использовать не только как средство обработки и накопления данных но и для управления реальными объектами и мониторинга, например, через интерфейс USB/RS-232. В этой работе раскрываются особенности системы чтения, обработки и накопления реальных аналоговых данных на базе Excel и платформы Arduino структурная схема которой показана на Рис. 1.

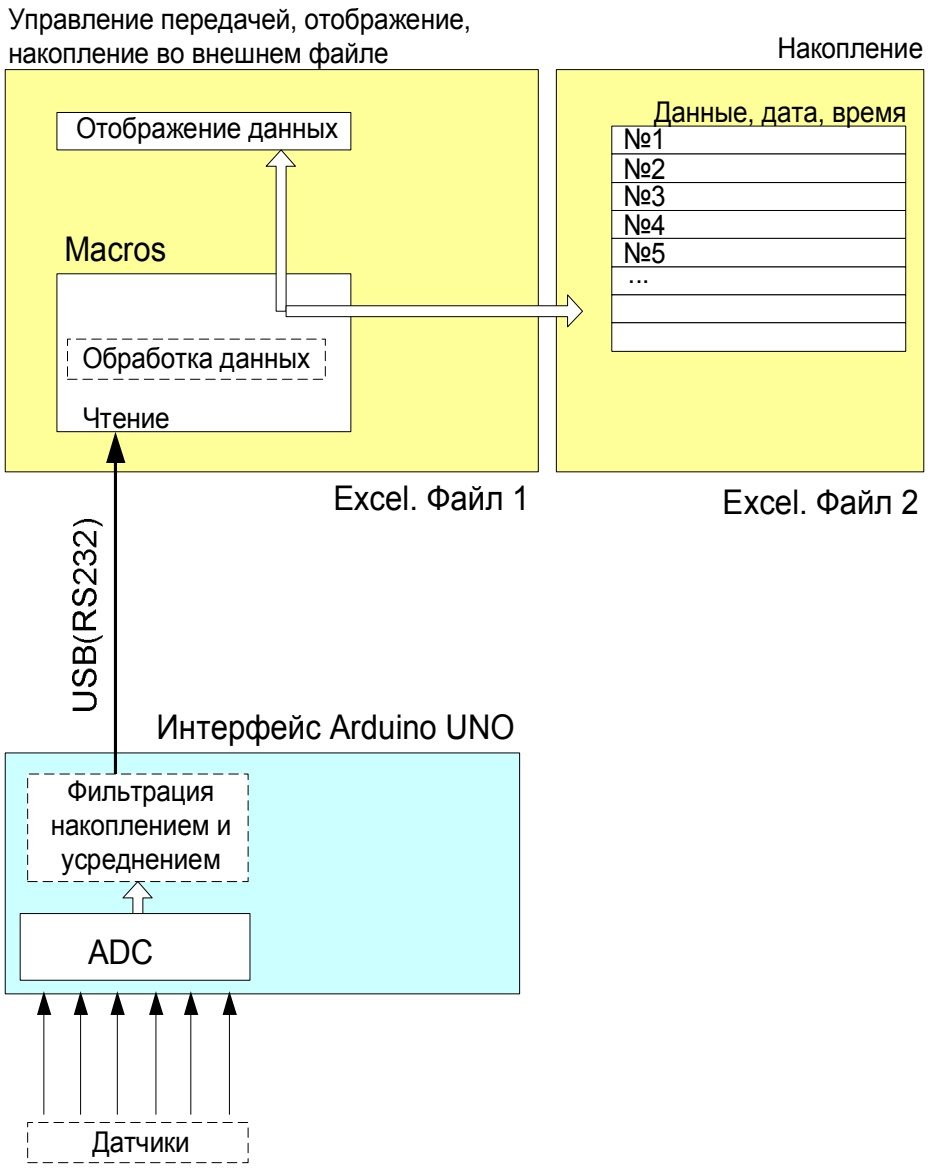


Рис. 1. Блок-схема системы накопления аналоговых данных, их фильтрации, обработки и накопления на базе MS Excel и платформы Arduino.

ОБЩИЕ СВЕДЕНИЯ

Платформа Arduino UNO

В этой работе Arduino UNO работает как устройство ввода аналоговых сигналов на максимальной частоте, накопления и усреднения данных перед их отправкой в последовательный канал передачи.

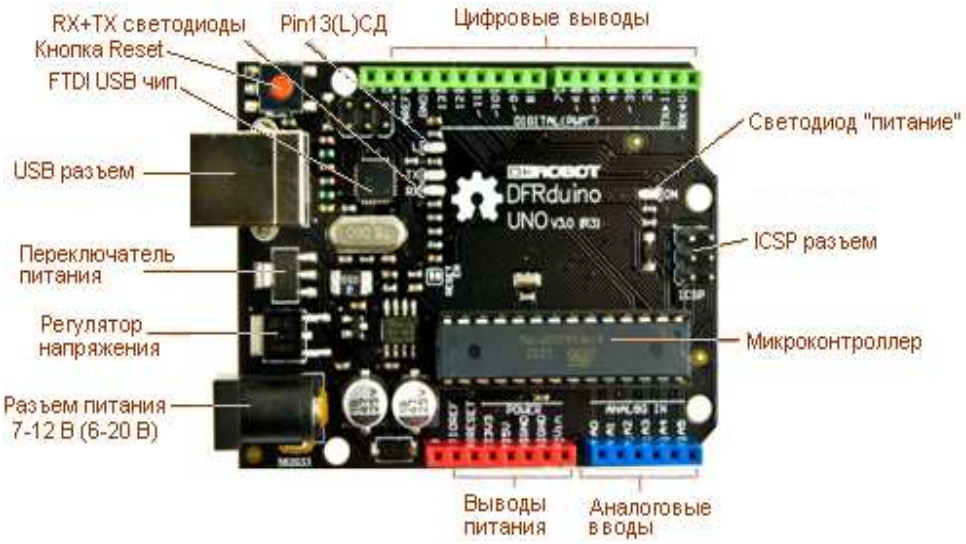


Рис. 2. Периферия платформы ARDUINO (DFRduino) UNO.

Платформа Arduino UNO имеет следующие характеристики.

- Микроконтроллер ATmega 328, частота 16 MHz,
- Напряжение питания: 7-12 В (6-20 В - предел). Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB).
- 32 Кб флэш память (2К занято загрузчиком),
- 2Кб ОЗУ
- 1Кб EEPROM
- 6 аналоговых (0-5В, 10бит, 0.1мс) вводов и 14 цифровых вводов/выводов с 6 PWM (ШИМ) выходами (~490 Гц, 0 .. 255)
- Входной ток ADC: 0 .. 1 мкА
- Встроенные USB-COM (300, .., 115200 бод), SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), и I2C: 4 (SDA) и 5 (SCL)каналы связи;
- 2 внешних прерывания

Примечание. Порядок подключения драйвера платформы, среды разработки, и базовых библиотек можно найти в “Создание интерактивных объектов и сред на базе платформы Arduino. Часть 1” на http://www.vr-online.ru/blog/sozdanie-interaktivnyh-obektov-i-sred-na-baze-platformy-arduino-chast-2-7969 и в “Создание интерактивных объектов и сред на базе платформы Arduino. Часть 2” на http://www.vr-online.ru/blog/sozdanie-interaktivnyh-obektov-i-sred-na-baze-platformy-arduino-chast-2-7969

Датчик температуры LM335

Особенности датчика температуры LM335:

- Непосредственное измерение температуры в град.К
- Диапазон измеряемых температур -40 .. +100 ^{o}C
- Точность 1^{o}C
- Отношение выходного напряжения к температуре 10мВ/ ^{o}C

- Выходное рабочее напряжение при 25C и токе 1мА 2.92 .. 2.98 В
- Постоянная времени в
  - стоячем воздухе 80 сек
  - движущемся воздухе (~5 мм/сек) 10 сек
  - движущемся масле 1 сек
- Динамический импеданс < 1 Ом
- Низкая стоимость ~18 руб.(август 2013 г.)

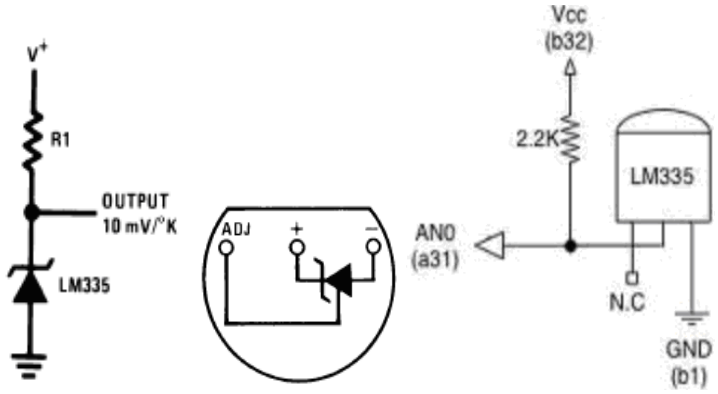


Рис. 3. Схемы включения датчика температуры LM335 без регулировки (контакт ADJ, N.C.).

Датчик влажности HIH-4000-002

Датчик влажности HIH-4000-002 имеет следующие характеристики.

- Диапазон рабочих температур -40 °C to 85 °C
- Диапазон измеряемой влажности 0% RH to 100 %RH,
  здесь и далее **RH** – относительная влажность
- Погрешность ±5 %RH (0 .. 59% RH), ±8 %RH (60 .. 100% RH)
- Гистерезис ±3 %RH
- Время отклика 15 сек (тау) в медленно движущемся воздухе
- Повторяемость ±0.5 % RH
- Время включения 70 мсек макс.
- Питание +5.8 В / 500 µA
- Дрейф ±1.2 %RH за 5 лет; ±0.25 %RH за год
- Стабильность при 50% RH ±1.2 %RH

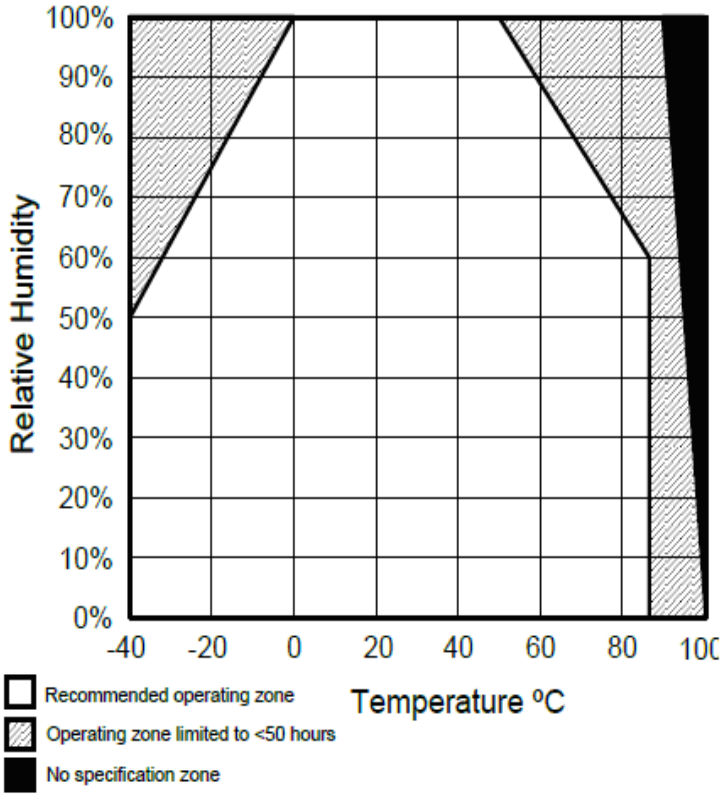


Рис. 4. Рекомендуемая рабочая зона температуры и влажности.

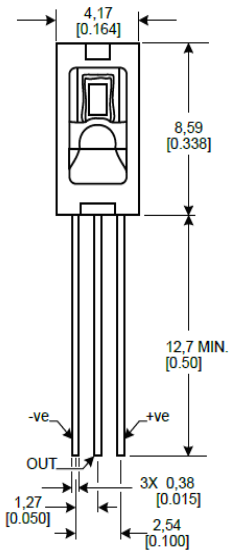


Рис. 5. Размеры датчика HIH-4000-002.

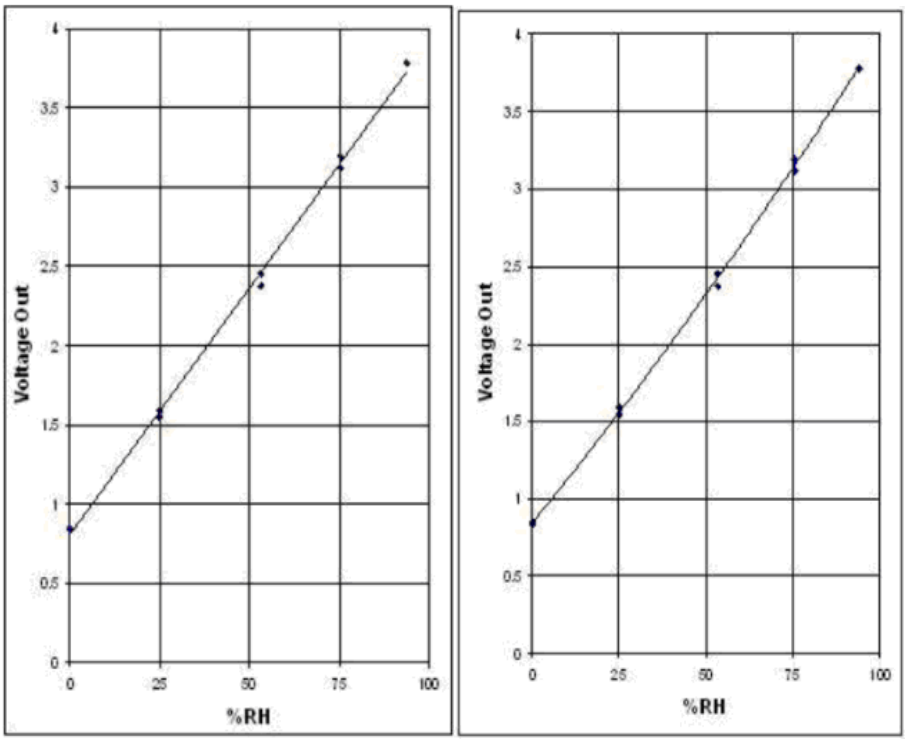


Рис. 6. Зависимость выходного напряжения датчика от влажности аппроксимированная полиномом первой (слева) и второй (справа) степени.

Для расчета влажности по выходному напряжению датчика и введению поправок по напряжению питания и температуры датчика необходимо использовать следующие отношения.
Выходное напряжение датчика:
Vout = Vsupply (0.0062 (sensor RH) + 0.16), полином 1-ой степени
Vout = 0.00003*(sensor RH)^2 + 0.0281*(sensor RH) + 0.820 для 25C, полином 2-ой степени

Температурная компенсация:
Vout=(1.0305+0.000044T-0.0000011T^2)*(Sensor RH) + (0.9237- 0.0041T+0.000040T^2), где T – температура в ^{o}C

Точка росы

Для вычисления точки росы (температуры, до которой должен охладиться воздух, чтобы достичь состояния насыщения водяным паром при данном влагосодержании и неизменном давлении) можно использовать показания датчиков температуры и влажности и графики, приведенные на Рис. 7.

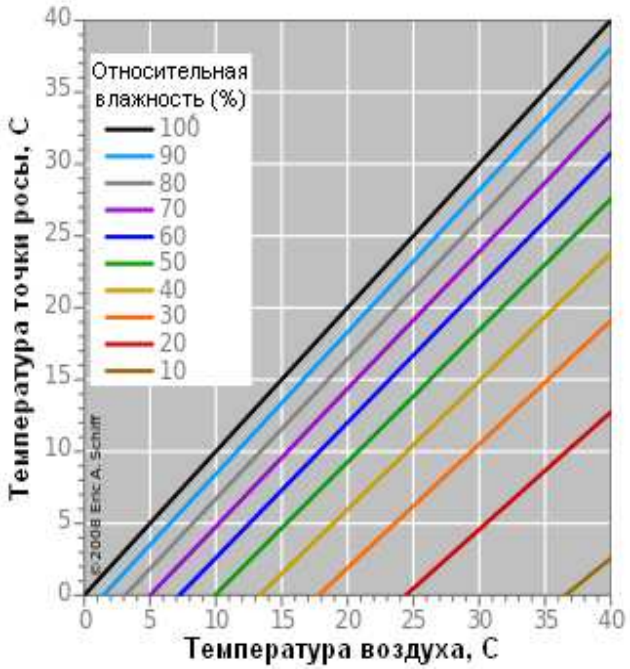


Рис. 7. Вычисление точки росы по температуре и относительной влажности.

Подключение датчиков к ADC платформы Arduino UNO

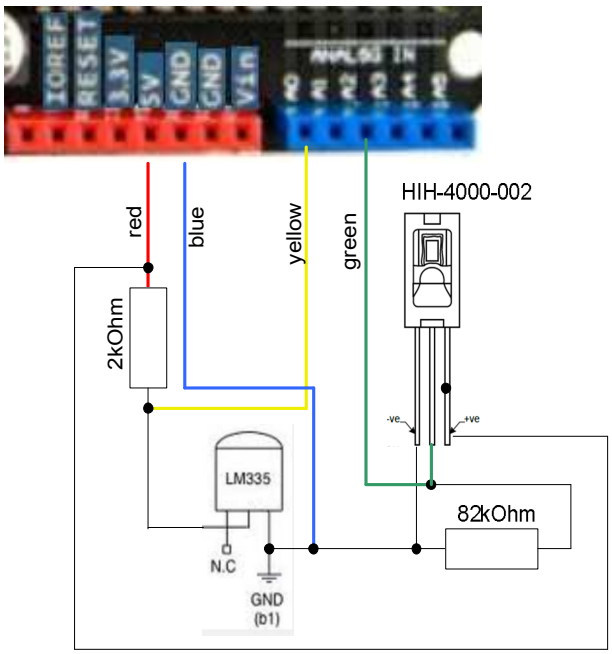


Рис. 8. Подключение датчика температуры LM335 к 1-му входу ADC (A0) и датчика влажности HIH-4000-002 к 3-му входу ADC (A2).

Сигнальный провод

Для подключения удаленных (до 15 метров) датчиков используется 4-х жильный сигнальный провод с сопротивлением 28 Ом / 50 м.

Калибровка датчиков.

Для калибровки датчиков можно использовать показания аттестованных калиброванных датчиков, установленных в той же точке измерения.

Для приближенного контроля показаний датчика температуры можно использовать следующие характерные температуры.

- 0°C – температура подтаявшего льда. Внимание: температура льда из морозильной камеры холодильника может быть значительно ниже.
- 36.6 °C – температура тела под языком.
- ~100 °C – температура кипящей воды. Держите контакты датчика сухими. Вода может изменить межконтактное сопротивление и, соответственно, показания датчика.

Для приближенного контроля показаний датчика влажности можно использовать дыхание параметры которого обсуждаются в статье “Parameters of the exhaled breath” http://www.sciencebits.com/exhalecondense: “Распространенным заблуждением является то, что выдыхаемый воздух имеет 100% влажность при 37 °C. Если бы это было так, мы бы увидели конденсацию при любых условиях наружного воздуха холоднее, чем 37 ° C. Очевидно, что это не так!”, “… Таким образом, воздух, выходящий из наших уст имеет температуру около 35°C и 95% влажность.

Полезные команды Arduino UNO

Команда	Описание
Serial.begin(9600);	Установка скорости передачи данных через COM порт (RS-232) из ряда: 300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200 бод (бит/с)
Serial.flush();	Очистка входного буфера COM порта
int led = 13; pinMode(led, OUTPUT);	Установка режима работы TTL порта Arduino (OUTPUT - на выход). К 13 порту подключен светодиод который используется для демонстрации цикличной работы Arduino
digitalWrite(led, LOW); digitalWrite(led, HIGH);	Включение и выключение светодиода платформы подключенного к 13-му TTL-порту
void loop() {}	Циклическое выполнение операторов помещенных в фигурные скобки
int adc_0; adc_0 = analogRead(0);	Чтение первого (нулевого) канала 10 разрядного 5В 0.1 мс АЦП
Serial.write(adc_0_Lo);	Запись в COM-порт байта 0 .. 127

unsigned long adc_0_sum = 0;
int loop_num = 0;
adc_0_sum = adc_0_sum + adc_0;
	Накопление и подсчет ADC данных

loop_num = loop_num + 1;	

if (loop_num > 30000) {
adc_0_sum = 0;
}
 	
	Защита от переполнения накопления данных

if (Serial.available() > 0) {}	Проверка прибытия данных через COM порт . Используется для синхронизации Excel средой процессов в Arduino
ctrl_num = Serial.read();	Считывание данных из COM порта
adc_0 = (adc_0_sum * 16) / loop_num;	Вычисление среднего значения, используется для фильтрации АЦП сигналов и повышения виртуальной разрядности АЦП до 14 бит
adc_0_Hi = ((adc_0 >> 7) & 0x7F);	Выделение младшей половины 14-разрядного среднего
adc_0_Hi = ((adc_0 >> 7) & 0x7F);	Выделение старшей половины 14-разрядного среднего

Serial.write(adc_0_Hi); // in 0 .. 127 data range
Serial.write(adc_0_Lo)
	Передача в COM-порт старшей и младшей половины 14-разрядного значения усредненного результата накопления 10-разрядных данных АЦП

Полезные команды Macros и MS ActiveX control

Команда	Описание
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)	Подключение системного таймера для выполнения задержек. Пример команды “задержка на одну секунду”: Sleep 1000&
Sub CommandButton1_Click() End Sub	Начало и конец подпрограммы
Dim StopMacro As Boolean	Инициализация глобальной / локальной переменной в зависимости от того где расположен оператор Dim - за пределами или внутри программного блока.
For i = 1 To 10000 Next i	Пример оператора цикла
DoEvents If StopMacro Then Exit For	Пример выхода из цикла For. Оператор DoEvents необходим перед использованием глобальных переменных
MSComm1.CommPort = 3 MSComm1.Settings = "9600,N,8,1" MSComm1.InputLen = 0 On Error Resume Next MSComm1.PortOpen = True	Подключение к COM порту и настройка его параметров

If Err Then
MsgBox "Com" & MSComm1.CommPort & ": not available. Change CommPort property to another port."
Exit Sub
End If
	Выдача сообщения об ошибке соединения с COM-портом

MSComm1.PortOpen = False	Закрытие COM порта
Set oXL = New Excel.Application Set oBook = oXL.Workbooks.Open("D:\pc\LONG_ADC_Arduino_Read\DataBase_1.xls") Set oSheet = oBook.Worksheets("Sheet1")	Пример открытия страницы внешней (по отношению к macros) электронной таблицы Excel
Set oSheet = Nothing oBook.Close True ' without False displays Message Set oBook = Nothing oXL.Quit Set oXL = Nothing 'ActiveWorkbook.Close True 'ActiveWorkbook.Close False	Пример закрытия внешнего Excel файла
MSComm1.Output = Chr$(1)	Запись единицы в COM порт для синхронизации процессов Arduino
b = MSComm1.Input	Считывание данных из буфера COM порта
Sheet1.Cells(5, 1).Value = i	Запись в ячейку Excel (в которой работает macros) переменной i
exampleDate = Now	Чтение системной даты и времени
Sheet1.Cells(5, 8).Value = Hour(exampleDate) Sheet1.Cells(5, 12).Value = Month(exampleDate)	Выделение из переменной exampleDate значений текущего часа и месяца
oSheet.Cells(Offset, 1).Value = Offset - 4	Запись в ячейку внешней Excel значения Offset - 4
Application.DisplayAlerts = False ThisWorkbook.Save Application.DisplayAlerts = True Application.Quit	Закрытие Excel с macros
UserForm1.Show ThisWorkbook.Save	Команда Workbook для автоматического запуска macros

Примечание. Команда <Ctrl + Break> останавливает работу macros.

Примеры ActiveX control можно найти в работе “COM/DCOM клиент и сервер автоматизации Microsoft ActiveX control” [4].

ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Подключение и настройка платформы Arduino UNO.

1. Подключите два датчика температуры к первому и второму входам АЦП (A0 и A1) и четыре датчика влажности к остальным входам по схеме показанной на Рис. 8.

   Примечание. Можно подключить любые другие датчики в любом порядке или не подключать их вовсе. Эти варианты потребуют соответствующей коррекции скрипта обработки и отображения аналоговых данных.

2. Загрузите интегрированную среду разработки и настройте драйвер платформы Arduino, например, как показано в разделе ПОДГОТОВКА К РАБОТЕ С ARDUINO ЧЕРЕЗ ИНТЕГРИРОВАННУЮ СРЕДУ РАЗРАБОТКИ работы “Создание интерактивных объектов и сред на базе платформы Arduino. Часть 1” http://www.vr-online.ru/blog/sozdanie-interaktivnyh-obektov-i-sred-na-baze-platformy-arduino-chast-1-7947

3. Используя команды Arduino, приведенные в таблице выше, разработайте программу которая для каждого канала 6-канального 10-разрядного АЦП считывает показания, накапливает и подсчитывает их, а в момент поступления запроса (числа) от Excel через COM-порт, вычисляет среднее значение АЦП увеличенное в 16 раз (4 разряда), округляет его до 14-разрядов, и двумя байтами посылает среднее значение в COM-порт.

   Ниже дан пример реализации заданного алгоритма.

```
/*
 Communication with Excel in RT mode

 created 24 August 2013
 modified
 by Bob Davidov

 Reads from 10-bit ADC (6 channels)
 Passes the ADC averages of 14 bits to COM port
 by two bytes each in 0 .. 127 range
 Order of sensors connected to ADC ports:
   adc_0, adc_1, adc_2, adc_3, adc_4, adc_5,
   temp1, temp2,  hum1,  hum2,  hum3,  hum4

 This example code is in the public domain.
 */
byte ctrl_num;  // control num of 0 .. 127 from Excel
int led = 13;   // control LED

int adc_0; // 10 bit ADC 0 .. 5V
int adc_1;
int adc_2;
int adc_3;
int adc_4;
```

```
int adc_5;

unsigned long adc_0_sum = 0; // for sum of adc_i
unsigned long adc_1_sum = 0;
unsigned long adc_2_sum = 0;
unsigned long adc_3_sum = 0;
unsigned long adc_4_sum = 0;
unsigned long adc_5_sum = 0;

int loop_num = 0;  // number of asc_i reading

byte adc_0_Hi;  // high byte (in 0 .. 127 range) of 14 bit average
byte adc_1_Hi;
byte adc_2_Hi;
byte adc_3_Hi;
byte adc_4_Hi;
byte adc_5_Hi;

byte adc_0_Lo;
byte adc_1_Lo;
byte adc_2_Lo;
byte adc_3_Lo;
byte adc_4_Lo;
byte adc_5_Lo;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600); //300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200
  Serial.flush();  // clear input buffer

  pinMode(led, OUTPUT);
}

void loop() {
  // read ADC
  adc_0 = analogRead(0);
  adc_1 = analogRead(1);
  adc_2 = analogRead(2);
  adc_3 = analogRead(3);
  adc_4 = analogRead(4);
  adc_5 = analogRead(5);

  // sum of captured data
  adc_0_sum = adc_0_sum + adc_0;
  adc_1_sum = adc_1_sum + adc_1;
  adc_2_sum = adc_2_sum + adc_2;
  adc_3_sum = adc_3_sum + adc_3;
```

```
adc_4_sum = adc_4_sum + adc_4;
adc_5_sum = adc_5_sum + adc_5;

// number of adc reading
loop_num = loop_num + 1; // ~ 0 .. 180 lopps per 0.1 ms

if (loop_num > 30000) { // protection against long delay in data reading
  adc_0_sum = 0;
  adc_1_sum = 0;
  adc_2_sum = 0;
  adc_3_sum = 0;
  adc_4_sum = 0;
  adc_5_sum = 0;
}

// get any incoming bytes:
if (Serial.available() > 0) {
  ctrl_num = Serial.read(); // 0 .. 255 from excel
  switch (ctrl_num) {
  case 1:
    digitalWrite(led, HIGH); // switch on the LED

    adc_0 = (adc_0_sum *16) / loop_num;  // int: average of adc x 16
    adc_1 = (adc_1_sum *16) / loop_num;
    adc_2 = (adc_2_sum *16) / loop_num;
    adc_3 = (adc_3_sum *16) / loop_num;
    adc_4 = (adc_4_sum *16) / loop_num;
    adc_5 = (adc_5_sum *16) / loop_num;

    adc_0_Hi = ((adc_0 >> 7) & 0x7F);  // byte: high half (7 bit) of 14 bit adc data (int word)
    adc_1_Hi = ((adc_1 >> 7) & 0x7F);
    adc_2_Hi = ((adc_2 >> 7) & 0x7F);
    adc_3_Hi = ((adc_3 >> 7) & 0x7F);
    adc_4_Hi = ((adc_4 >> 7) & 0x7F);
    adc_5_Hi = ((adc_5 >> 7) & 0x7F);

    adc_0_Lo = (adc_0 & 0x7F);        // byte: low half of 14 bit adc data (int word)
    adc_1_Lo = (adc_1 & 0x7F);
    adc_2_Lo = (adc_2 & 0x7F);
    adc_3_Lo = (adc_3 & 0x7F);
    adc_4_Lo = (adc_4 & 0x7F);
    adc_5_Lo = (adc_5 & 0x7F);

    Serial.write(adc_0_Hi); // in 0 .. 127 data range
    Serial.write(adc_0_Lo);
    Serial.write(adc_1_Hi);
    Serial.write(adc_1_Lo);
```

```
      Serial.write(adc_2_Hi);
      Serial.write(adc_2_Lo);
      Serial.write(adc_3_Hi);
      Serial.write(adc_3_Lo);
      Serial.write(adc_4_Hi);
      Serial.write(adc_4_Lo);
      Serial.write(adc_5_Hi);
      Serial.write(adc_5_Lo);

      // reset sum of adc data
      adc_0_sum = 0;
      adc_1_sum = 0;
      adc_2_sum = 0;
      adc_3_sum = 0;
      adc_4_sum = 0;
      adc_5_sum = 0;

      loop_num = 0;

      digitalWrite(led, LOW); // switch off the LED

      break;

    default:
      break;
    }
  }
}
```

4. При помощи интегрированной среды запишите программу в память платформы Arduino.

Задание 2. Настройка MS Excel на работу с COM портом.

1. Установите соединение Excel с COM-портом, например, как показано в разделе ОБЩИЕ СВЕДЕНИЯ и в задании 2 работы “ Управление из Excel через COM порт” http://www.vr-online.ru/blog/upravlenie-iz-excel-cherez-com-port-8475
2. Проверьте работу соединения

Задание 3. Считывание аналоговых данных в Excel.

1. Используя команды Macros, приведенные в таблице выше, разработайте программу которая циклически устанавливает связь с COM-портом, через COM порт передает платформе Arduino код синхронизации, принимает усредненные данные АЦП платформы, системную дату и время операционной среды и отображает принятые данные в ячейках электронной таблицы и разрывает соединение с COM-портом.

Примечание. Для получения корректного значения усредненного сигнала АЦП не забудьте поделить на 16 увеличенное в Arduino значение АЦП.

Ниже дан пример реализации заданного алгоритма.

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Dim StopMacro As Boolean
_____________________________________________________________________
Private Sub CommandButton1_Click()

  Dim b() As Byte    ' for COM port Input buffer reading
  ' for saving data in external xls file:

 For i = 1 To 10000
   MSComm1.CommPort = 3
   MSComm1.Settings = "9600,N,8,1"
   MSComm1.InputLen = 0
   On Error Resume Next

   MSComm1.PortOpen = True
   Sleep 1000& 'Delay for 1 second

   'start Arduino sending of ADC 14 bit averages into COM port
   MSComm1.Output = Chr$(1)  'LED High
   Sleep 500& 'Delay for 0.5 second

   b = MSComm1.Input 'reading ADC data
   'ADC 10 bit average = (Hi byte x 128 + Lo byte) / 16
   'Arduino averages ~730 ADC samples each excel loop with 0.5 s delay

   Sheet1.Cells(5, 1).Value = i
   '                                                    VVVV  Offset in voltage
   Sheet1.Cells(5, 2).Value = ((((b(UBound(b) - 23)) * 128 + b(UBound(b) - 21)) / 16) * (5 / 1024) - 2.92) * 100 ' Temperature 1
   Sheet1.Cells(5, 3).Value = ((((b(UBound(b) - 19)) * 128 + b(UBound(b) - 17)) / 16) * (5 / 1024) - 2.92) * 100 ' Temperature 2
   Sheet1.Cells(5, 4).Value = ((((b(UBound(b) - 15)) * 128 + b(UBound(b) - 13)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity
1
   Sheet1.Cells(5, 5).Value = ((((b(UBound(b) - 11)) * 128 + b(UBound(b) - 9)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity 2
   Sheet1.Cells(5, 6).Value = ((((b(UBound(b) - 7)) * 128 + b(UBound(b) - 5)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity 3
   Sheet1.Cells(5, 7).Value = ((((b(UBound(b) - 3)) * 128 + b(UBound(b) - 1)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity 4
   '                                                    ^^^ sensor supply
   exampleDate = Now
   Sheet1.Cells(5, 8).Value = Hour(exampleDate)
   Sheet1.Cells(5, 9).Value = Minute(exampleDate)
   Sheet1.Cells(5, 10).Value = Second(exampleDate)
   Sheet1.Cells(5, 11).Value = Day(exampleDate)
```

```
    Sheet1.Cells(5, 12).Value = Month(exampleDate)
    Sheet1.Cells(5, 13).Value = Year(exampleDate)

    ' close COM port
    MSComm1.PortOpen = False

  Next i
End Sub
```

2. Настройте отображение данных как показано ниже.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	*Измерение и запись температуры и влажности*												
2													
3		Температура, С		Влажность, %				Время			Дата		
4		1	2	1	2	3	4	**Час**	**Мин**	**Сек**	День	Месяц	Год
5	7	12.5	12.7	87.4	88.3	86.2	82.1	8	11	42	3	9	2013
6													
7													

Задание 4. Запись данных в отдельный Excel файл.

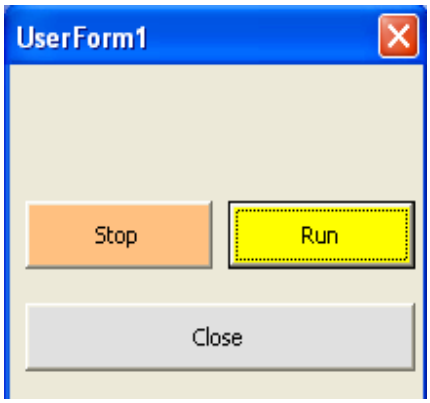
1. Добавьте в программу задания 3 команды открытия внешнего Excel файла (см. таблицу Macros выше), записи данных во внешний файл и закрытия внешнего файла.

2. Настройте таблицу внешнего файла как показано ниже.

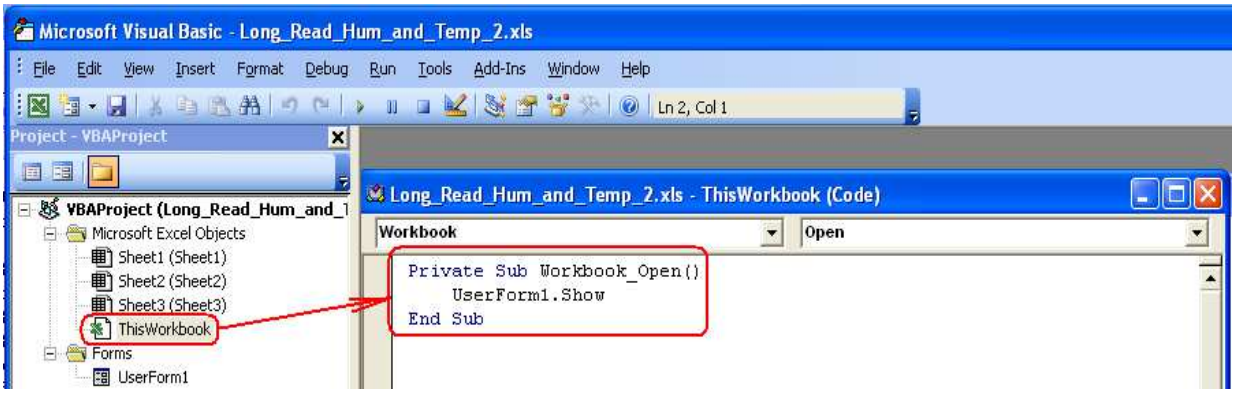
	A	B	C	D	E	F	G	H	I
1	*Наблюдение за температурой и влажностью*								
2									
3	Start cell	Дата, Время	Температура, С		Влажность, %				Время, сек
4	12		1	2	1	2	3	4	
5	1	9/3/2013 8:11	12.4	12.8	87.4	88.3	86.2	82.1	28
6	2	9/3/2013 8:11	12.5	12.8	87.4	88.3	86.2	82.1	30
7	3	9/3/2013 8:11	12.4	12.8	87.4	88.3	86.2	82.1	33
8	4	9/3/2013 8:11	12.4	12.8	87.4	88.3	86.2	82.1	35
9	5	9/3/2013 8:11	12.5	12.7	87.4	88.3	86.2	82.1	37
10	6	9/3/2013 8:11	12.4	12.7	87.4	88.3	86.2	82.1	40
11	7	9/3/2013 8:11	12.5	12.7	87.4	88.3	86.2	82.1	42
12									
13									

Задание 5. Быстрая загрузка, остановка мониторинга и выгрузка рабочей программы.

1. Используя команды Macros, приведенные в таблице выше, разработайте показанный ниже интерфейс, в котором кнопка “Run” запускает циклический процесс чтения, отображения и накопления аналоговых данных задания 3 и 4, кнопка “Stop” останавливает циклический процесс, а кнопка “Close” закрывает Excel файл с Macros.



2. Обеспечьте автоматический запуск Macros при открытии Excel файла следующей командой Workbook.



3. Сравните собственный код с примером кода выполнения задания:

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Dim StopMacro As Boolean
_____________________________________________________________________
Private Sub CommandButton1_Click()

 Dim b() As Byte   ' for COM port Input buffer reading
 ' for saving data in external xls file:
 Dim oXL As Excel.Application, oBook As Excel.Workbook, oSheet As Excel.Worksheet, vValue As Variant
 Dim Offset As Integer  ' or Long

 StopMacro = False

 For i = 1 To 10000

  DoEvents  'statement in the loop to click the other button
  If StopMacro Then Exit For 'Sub

  MSComm1.CommPort = 3
  MSComm1.Settings = "9600,N,8,1"
  MSComm1.InputLen = 0
  On Error Resume Next
```

```
MSComm1.PortOpen = True

If Err Then
  MsgBox "Com" & MSComm1.CommPort & ": not available. Change CommPort property to another port."
  Exit Sub
End If

' using the Excel automation objects to open xls file
Set oXL = New Excel.Application
Set oBook = oXL.Workbooks.Open("D:\pc\LONG_ADC_Arduino_Read\DataBase_1.xls")
Set oSheet = oBook.Worksheets("Sheet1")

Sleep 2000& 'Delay for 2 second

'start Arduino sending of ADC 14 bit averages into COM port
MSComm1.Output = Chr$(1)  'LED High
Sleep 500& 'Delay for 0.5 second

b = MSComm1.Input 'reading ADC data
'ADC 10 bit average = (Hi byte x 128 + Lo byte) / 16
'Arduino averages ~730 ADC samples each excel loop with 0.5 s delay

Sheet1.Cells(5, 1).Value = i
'                                                       VVVV  Offset in voltage
Sheet1.Cells(5, 2).Value = ((((b(UBound(b) - 23)) * 128 + b(UBound(b) - 21)) / 16) * (5 / 1024) - 2.92) * 100 ' Temperature 1
Sheet1.Cells(5, 3).Value = ((((b(UBound(b) - 19)) * 128 + b(UBound(b) - 17)) / 16) * (5 / 1024) - 2.92) * 100 ' Temperature 2
Sheet1.Cells(5, 4).Value = ((((b(UBound(b) - 15)) * 128 + b(UBound(b) - 13)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity
1
Sheet1.Cells(5, 5).Value = ((((b(UBound(b) - 11)) * 128 + b(UBound(b) - 9)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity
2
Sheet1.Cells(5, 6).Value = ((((b(UBound(b) - 7)) * 128 + b(UBound(b) - 5)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity 3
Sheet1.Cells(5, 7).Value = ((((b(UBound(b) - 3)) * 128 + b(UBound(b) - 1)) / 16) * (5 / 1024) / 4.8 - 0.16) / 0.0062 ' Humidity 4
'                                                       ^^^ sensor supply
exampleDate = Now
Sheet1.Cells(5, 8).Value = Hour(exampleDate)
Sheet1.Cells(5, 9).Value = Minute(exampleDate)
Sheet1.Cells(5, 10).Value = Second(exampleDate)
Sheet1.Cells(5, 11).Value = Day(exampleDate)
Sheet1.Cells(5, 12).Value = Month(exampleDate)
Sheet1.Cells(5, 13).Value = Year(exampleDate)

Offset = oSheet.Cells(4, 1).Value     ' Index offset
oSheet.Cells(Offset, 1).Value = Offset - 4
oSheet.Cells(Offset, 2).Value = exampleDate
oSheet.Cells(Offset, 3).Value = Sheet1.Cells(5, 2).Value
oSheet.Cells(Offset, 4).Value = Sheet1.Cells(5, 3).Value
```

```
    oSheet.Cells(Offset, 5).Value = Sheet1.Cells(5, 4).Value
    oSheet.Cells(Offset, 6).Value = Sheet1.Cells(5, 5).Value
    oSheet.Cells(Offset, 7).Value = Sheet1.Cells(5, 6).Value
    oSheet.Cells(Offset, 8).Value = Sheet1.Cells(5, 7).Value
    oSheet.Cells(Offset, 9).Value = Sheet1.Cells(5, 10).Value

    Offset = Offset + 1

    oSheet.Cells(4, 1).Value = Offset        ' Cell offset

    ' close external xls file
    Set oSheet = Nothing
    oBook.Close True   ' without False displays Message
    Set oBook = Nothing
    oXL.Quit
    Set oXL = Nothing

    'ActiveWorkbook.Close True
    'ActiveWorkbook.Close False

    ' close COM port
    MSComm1.PortOpen = False

  Next i
  Exit Sub

  Application.CommandBars(1).Reset 'reset the main Worksheet Menu bar

End Sub
_____________________________________________________________________
Private Sub CommandButton2_Click()
    StopMacro = True
End Sub
_____________________________________________________________________
Private Sub CommandButton3_Click()
  'End 'Comment: It Exit all macros
  '<Ctrl + Break> stops running macro

  'The following closes Excel Application:
  Application.DisplayAlerts = False
  ThisWorkbook.Save
  Application.DisplayAlerts = True
  Application.Quit
End Sub
_____________________________________________________________________
```

4. Проверьте работоспособность разработанной системы мониторинга которая считывает аналоговые сигналы, фильтрует их, обрабатывает, отображает и накапливает. Система имеет простой интерфейс который может автоматически запускать при открытии Excel файла, имеет возможность ручной остановки процесса мониторинга и выхода из системы.

Задание 6. Накопление данных в xls файле под управлением МатЛАБ.

1. Замените XLS модуль управления передачей, обработки и записи данных (см. Рис. 1, Задание 3) аналогичным модулем на базе m-файла МатЛАБ (см. Рис. 9) использующего для записи данных в xls файл технологию Microsoft ActiveX control.

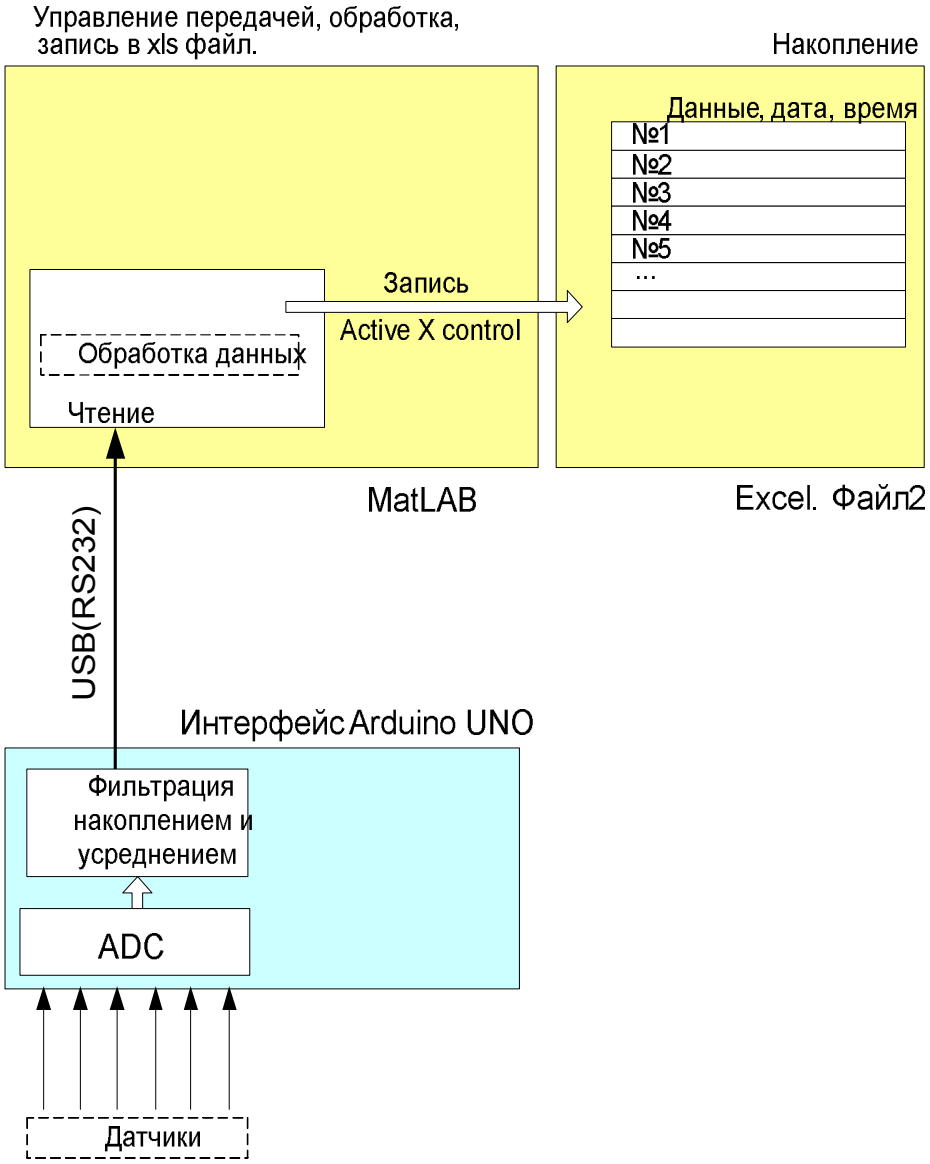


Рис. 9. Блок-схема системы накопления аналоговых данных, их фильтрации, обработки и накопления на базе МатЛАБ, MS Excel и платформы Arduino.

Пример m-кода модуля МатЛАБ использующего технологию ActiveX control для записи обработанных данных канала RS-232 в MS Excel:

```
% Arduino initializing
s = serial('COM3','BaudRate',9600,'DataBits',8,'Parity', 'none',
'FlowControl', 'none');
set (s, 'Timeout', 0.1);
fopen (s);

% xls initializing
filename = 'DataBase_1.xls'; % existing file
filename = fullfile (pwd, filename);
hExcel = actxserver ('excel.application');
%set(hExcel, 'Visible', 1);
Workbook = hExcel.Workbooks;

%Read xls data
invoke(Workbook,'Open',filename);
LineNum = hExcel.Activesheet.get ('Range', 'A4');
line_num = LineNum.value;

if isnan (line_num) % is A4 empty ?
    line_num = 7;
end
pause(1);

for i = 1:10
    fprintf(s,'1');
    In = fread(s);
    if ~isempty (In)
        temp_1 = (((In(1) * 128 + In(2)) / 16) * (5 / 1024) - 2.87) * 100;
        temp_2 = (((In(3) * 128 + In(4)) / 16) * (5 / 1024) - 2.87) * 100;
        hum_1 = (((In(5)*128+In(6))/16) *  (5 / 1024) / 4.8 - 0.16) / 0.0062;
        hum_2 = (((In(7)*128+In(8))/16) *  (5 / 1024) / 4.8 - 0.16) / 0.0062;
        hum_3 = (((In(9)*128+In(10))/16)*  (5 / 1024) / 4.8 - 0.16) / 0.0062;
        hum_4 = (((In(11)*128+In(12))/16)* (5 / 1024) / 4.8 - 0.16) / 0.0062;

        data = [temp_1 temp_2 hum_1 hum_2 hum_3 hum_4];

        range = ['A' num2str(line_num)];
        records = hExcel.Activesheet.get ('Range', range);
        records.value = line_num - 4;

        range = ['B' num2str(line_num)];
        date_time = hExcel.Activesheet.get ('Range', range);
        date_time_now = [datestr(now, 23) ' ' datestr(now, 13)];
        date_time.value = date_time_now;

        range = ['C' num2str(line_num) ':H' num2str(line_num)];
        ADC_data = hExcel.Activesheet.get ('Range', range);
        ADC_data.value = data;

        range = ['I' num2str(line_num)];
        time_sec = hExcel.Activesheet.get ('Range', range);
```

```
        time_sec.value = date_time_now (length(date_time_now)-
1:length(date_time_now));

        line_num = line_num + 1;
        pause (1);
    end
    pause (0.1);
end
LineNum = hExcel.Activesheet.get ('Range', 'A4');
LineNum.value = line_num;

% close Arduino
fclose(s);
delete(s);

% close xls
invoke(hExcel.ActiveWorkbook,'Save')
Workbook.Close;
invoke(hExcel,'Quit');
hExcel.delete;
```

2. Проверьте работоспособность m-модуля сравнивая результат с xls-модулем (Задание 4 и 5) который читает данные платформы Arduino, обрабатывает их и записывает в другой xls файл.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как обеспечить управление объектом из Excel используя порты платформы Arduino?
2. Какое максимальное количество строковых записей можно сделать в Excel файле?
3. Какими средствами можно расширить объем записываемой информации в Excel файлы?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Создание интерактивных объектов и сред на базе платформы Arduino. Часть 1 http://www.vr-online.ru/blog/sozdanie-interaktivnyh-obektov-i-sred-na-baze-platformy-arduino-chast-1-7947
2. Создание интерактивных объектов и сред на базе платформы Arduino. Часть 2 http://www.vr-online.ru/blog/sozdanie-interaktivnyh-obektov-i-sred-na-baze-platformy-arduino-chast-2-7969
3. Управление из Excel через COM порт http://www.vr-online.ru/blog/upravlenie-iz-excel-cherez-com-port-8475
4. Dr. Bob Davidov. COM/DCOM клиент и сервер автоматизации Microsoft ActiveX control. http://portalnp.ru/2013/10/1156
5. Parameters of the exhaled breath. http://www.sciencebits.com/exhalecondense
6. Dr. Bob Davidov. Компьютерные технологии управления в технических системах http://portalnp.ru/author/bobdavidov.

Dr. Bob Davidov

Кафедра Автоматики и процессов управления Санкт-Петербургского государственного электротехнического университета “ЛЭТИ”

Построение интерфейса пользователя локальной системы управления на базе контроллера Arduino UNO

Цель работы: освоение правил подключения и настройки средств отображения воздействий и состояния локальной системы.

Задача работы: построение системы отображения параметров локальной системы средствами Simulink сетевого компьютера и дисплея TFT LCD, подключенных к контроллеру Arduino UNO.

Приборы и принадлежности: Контроллер Arduino UNO, дисплей 2.4“ TFT LCD, персональный компьютер, USB кабель, пакет моделирования MatLAB Simulink, среда программирования контроллера Arduino, библиотеки TFT дисплея A137.

ВВЕДЕНИЕ

Популярный контроллер Arduino UNO, имеющий дискретные устройства ввода-вывода, аналого-цифровые преобразователи и каналы последовательной передачи данных, можно использовать в качестве интерфейса компьютера для подключения внешних устройств. Контроллер может решать задачи предварительной обработки сигналов, накопления данных и локального управления как автономно, так и совместно с компьютером.
Использование недорогих и выразительных средств отображение данных и параметров системы повышает качество таких систем. В этой работе затронуты вопросы, связанные с подключением, настройкой и адаптацией к задачам пользователя цветного графического сенсорного дисплея с микро SD считывателем для Arduino UNO R3.

В разделе “Общие сведения” освещаются следующие вопросы.

- 2.4” TFT LCD дисплей для Arduino UNO
- Подключение дисплея к Arduino UNO • Библиотека примитивов
- Изменение шрифтов
- Установка и перекодировка цветов дисплея
- Определение координат сенсорной точки.
- Особенности построения сенсорных клавиш

- Работа с микро SD картой
- Управление боковой подсветкой
- Высвобождение входа ADC4 для нужд пользователя
- Мерцание выводимых на дисплей значений

ОБЩИЕ СВЕДЕНИЯ

2.4” TFT LCD дисплей для Arduino UNO

Среди множества предлагаемых на рынке дисплеев для компьютера Arduino UNO выбран дисплей с неплохим отношением цены к качеству. Среди основных его достоинств – низкая цена. С доставкой цена составила US$5.3

2.4” цветной графический сенсорный дисплей с микро флеш считывателем A 137 компании имеет следующие характеристики.

- Полное название 2.4" TFT LCD Shield Touch Panel Module TF Reader Micro SD For Arduino UNO R3 (http://www.mcufriend.com)
- Диагональ LCD TFT дисплея — 2.4"
- Разрешение — 240x320 точек
- Количество цветов — 18-бит (262 000)
- Контроллер дисплея — SPFD5408 со встроенным видео RAM буфером
- Драйвер — Sitronix ST7783 262K Color Single-Chip TFT Controller/Driver
- Цифровой интерфейс — 8 бит данных и 4 бита управления
- Порты Arduino занятые дисплеем — DIO 2 .. 9, AI 0 .. 3*
- Свободные порты Arduino — DIO 0, DIO 1, AI 4*, AI 5 и DIO 12 (если не использовать считыватель микро SD карты)
- Напряжение — 5В, работает с логикой 3.3В или 5В
- Потребление — 3.3В / 300mA, LDO регулятор
- Боковая подсветка — 4 белых светодиода (LED), которые можно подключить через транзистор для управления подсветкой
- Сенсорный экран — 4-проводной резистивный.
- Считыватель — микро SD (проверена работа с 8 ГБ картой)

Контакты используемые для подключения TFT LCD A137 к Arduino UNO:

LCD_CS (Chip Select) - A3 (Analog 3)
LCD_CD (Command/Data) - A2 (Analog 2)
LCD_WR (Write) - A1 (Analog 1)
LCD_RD (Read) - A0 (Analog 0)

LCD_RESET	- A4* (Analog 4), можно пересоединить LCD_RESET к линии RESET Arduino UNO как описано ниже.
LCD_D0	- DIO 8
LCD_D1	- DIO 9
LCD_D2	- DIO 2
LCD_D3	- DIO 3
LCD_D4	- DIO 4
LCD_D5	- DIO 5
LCD_D6	- DIO 6
LCD_D7	- DIO 7
SD SS	- DIO 10
SD DI	- DIO 11
SD DO	- DIO 12
SD SCLK	- DIO 13

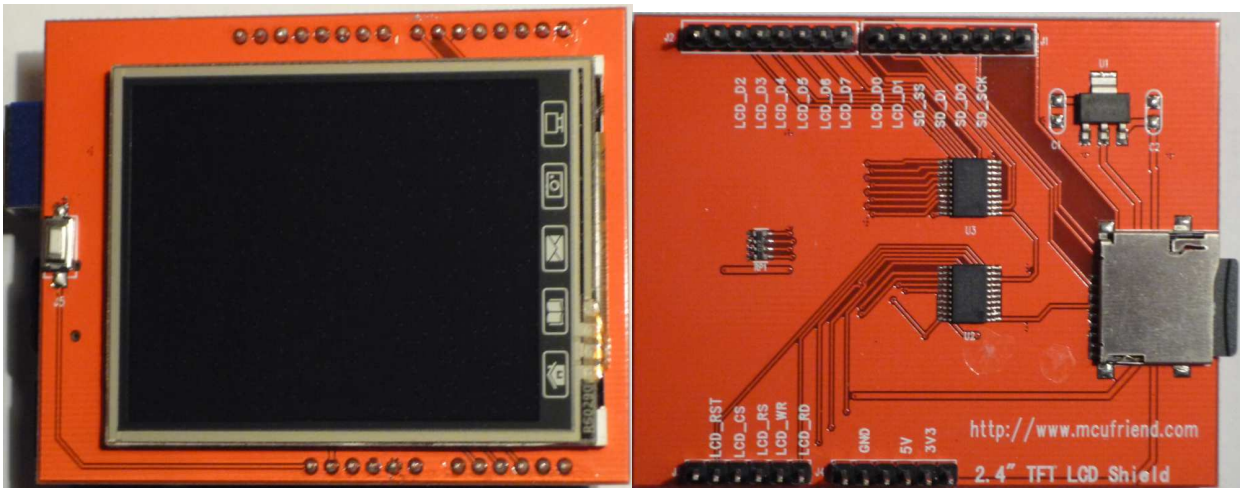


Рис. 1. Цветной графический сенсорный дисплей A137 (вид с двух сторон) с микро SD считывателем (вставлена 8 ГБ карта) для подключения “разъем в разъем” к контроллеру Arduino UNO. Размеры плат дисплея A137 и контроллера Arduino UNO совпадают.

Подключение дисплея к Arduino UNO

Перед подключением дисплея к контроллеру его библиотеки должны быть установлены на компьютере. Для ПК с ОС Windows это можно сделать следующим образом.

1. Загрузите (например, с http://arduino.cc/en/main/software) на компьютер с ОС Windows пакет программ Arduino не старше, 1.0.6 версии. Распакуйте пакет программ и поместите его в каталог, например, c:\arduino-1.0.6\
2. Подсоедините к портам обесточенного Arduino дисплейную плату совместив указанные на платах одноименные порты.

3. Подключите Arduino к USB порту компьютера. Когда операционная среда запросит местоположение драйвера укажите папку c:\arduino-1.0.6\drivers\
4. После установки драйвера, номер COM порта выделенный вашим компьютером для Arduino можно найти в списке менеджера устройств (Driver Manager).

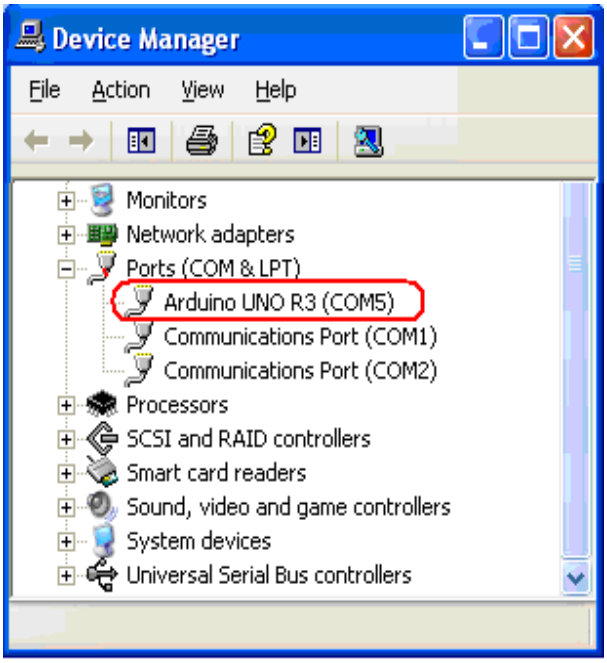


Рис. 2. Драйвер Arduino UNO R3 в списке менеджера устройств.

5. Запустите программу Arduino c:\arduino-1.0.6\arduino.exe
6. Загрузите программу управления миганием светодиода платы Arduino: Меню > File > Examples > 01.Basics> Blink

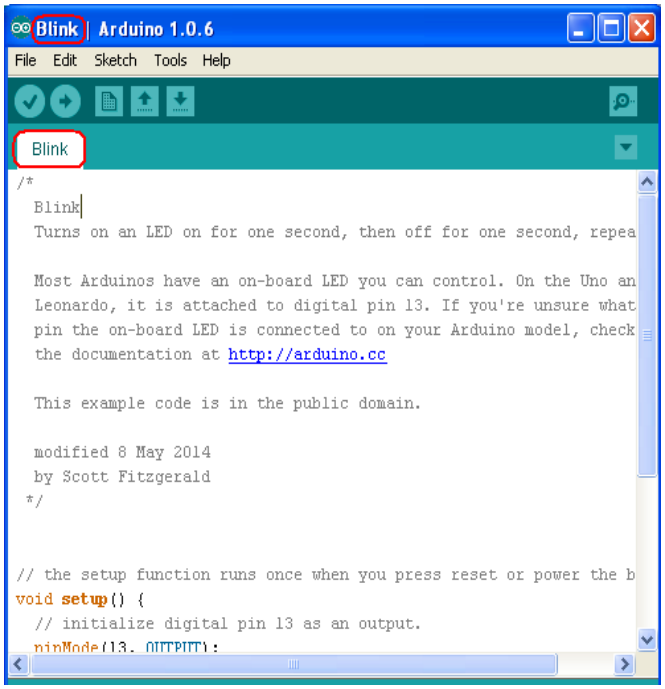


Рис. 3. Среда программирования Arduino.

7. Нажав на стрелку меню  запустите программу Blink. Если не появилось сообщений об ошибке, убедитесь, что загрузка программы прошла успешно и контроллер выполняет программу по переключению светодиода платы Arduino подключенного к порту DIO 13.
8. Подключите библиотеки для работы с устройствами дисплея платы A137. Загрузите с https://github.com/Smoke-And-Wires/TFT-Shield-Example-Code/archive/master.zip модифицированную TFT SHIELD библиотеку. Распакуйте zip и поместите его содержимое в раздел c:\arduino-1.0.6\libraries\. Убедитесь, что название распакованной папки и имена SWTFT.cpp и SWTFT.h файлов совпадают. При необходимости, исправьте имя папки.
9. Загрузите библиотеку графики с https://github.com/adafruit/Adafruit-GFX-Library/archive/master.zip. По аналогии с предыдущим пунктом, поместите папку в раздел библиотек Arduino и проверьте имя папки, сравнивая его с Adafruit_GFX.cpp и Adafruit_GFX.h файлами.
10. Аналогичным образом, подключите библиотеку сенсорного дисплея сайта https://github.com/adafruit/Touch-Screen-Library/archive/master.zip. TouchScreen - имя cpp и h файла.
11. Скопируйте эти три папки: Adafruit_GFX, SWTFT and TouchScreen

    из c:\arduino_1.0.6\libraries\ в C:\ProgramFiles\Arduino\Libraries
    в c:\ProgramFiles\Arduino\Libraries
12. Проверьте работу демонстрационных программ графического дисплея. Для этого в программе Arduino загрузите один из демонстрационных pde файлов: Меню > File > Open > c:\arduino-1.0.6\libraries\SWTFT\examples\. Данные программы помимо дисплея могут отображаться в окне COM соединения с компьютером который открывается кнопкой меню программы Arduino:  . Через это окно можно передавать данные программе которая считывает их из COM порта Arduino.

Примечание:

- graphicstest.pde – демонстрация графики дисплея с выводом времени построения графики в окно COM соединения.
- rotationtest.pde – демонстрация реакции дисплея на нажатие двух клавиш (произвольной и Enter) на клавиатуре компьютера.
- tftbmp.pde - отображает на дисплее bmp изображение с корневого каталога флеш-карты (miniwoof.bmp). Размер 24-bit bmp файла не должен превышать 260 КБ.
- tftpaint.ino - позволяет наносить цветные линии (палитра из 6 цветов) на экран сенсорного дисплея

Библиотека примитивов

Для вывода примитива на дисплей используются следующие команды.

```
// подключение библиотек
#include <Adafruit_GFX.h>  // ядро графической библиотеки
#include "SWTFT.h" // библиотека аппаратных средств

SWTFT tft;

// управление TFT дисплеем
tft.reset(); // сброс дисплея
uint16_t identifier = tft.readID();
tft.begin(identifier); // включение дисплея

// разворот направления печати дисплея
tft.setRotation(rotation); // установка угла начальных координат дисплея, rotation = 0 .. 3
tft.setRotation(tft.getRotation()+1); // разворот экрана на 90 градусов
```

Примечание: Драйвер полученного дисплея A137 имеет следующий номер

LCD driver chip: **7783**

Этот номер может использоваться для поиска и настройки других библиотек дисплея.

Таблица 1. Примитивы библиотеки Adafruit_GFX [1] tft объекта (SWIFT tft;).

Команда построения примитива	Назначение
ТЕКСТ	
tft.drawChar(int16_t x, int16_t y, unsigned char c, uint16_t color, uint16_t bg, uint8_t size)	Рисование символа шрифта
tft.void setCursor(uint16_t x0, uint16_t y0);	Установка курсора в зоне 240 x 320 точек
tft.void setTextColor(uint16_t color);	Установка цвета шрифта
tft.void setTextColor(uint16_t color, uint16_t backgroundcolor);	Установка цвета шрифта и фона
tft.void setTextSize(uint8_t size);	Установка размера шрифта
tft.void setTextWrap(boolean w);	Установка режима переноса символа длинной строки на новую или на начало текущей строки
tft.print(12.345); tft.print(“Hello”); tft.println(“World”); tft.print(0xF81F, HEX).	Вывод чисел и строк на дисплей
ТОЧКА (пиксел)	
tft.drawPixel(uint16_t x, uint16_t y, uint16_t color););	Вывод точки (пикселя)
ЛИНИЯ	
tft.drawLine(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t color);	Построение линии между точками (x0,y0) и (x1,y1)
tft.void drawFastVLine(uint16_t x0, uint16_t y0,	Построение вертикальной линии

uint16_t length, uint16_t color)	длиной length
tft.void drawFastHLine(uin86_t x0, uin86_t y0, uint8_t length, uint16_t color);	Построение горизонтальной линии длиной length
КРУГ	
tft.drawCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color);	Рисование круга радиусом r с центром в x0,y0
tft.fillCircle (uint16_t x0, uint16_t y0, uint16_t r, uint16_t color)	Заливка круга радиусом r с центром в x0,y0
ТРЕУГОЛЬНИК	
tft.drawTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, color);	Рисование треугольника с вершинами (x0,y0), (x1,y1) и (x2,y2)
tft.fillTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, color);	заливка треугольника с вершинами (x0,y0), (x1,y1) и (x2,y2)
ПРЯМОУГОЛЬНИК	
tft.drawRect(drawRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);	Рисование прямоугольника шириной w и высотой h
tft.fillRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);	заливка прямоугольника шириной w и высотой h
ПРЯМОУГОЛЬНИК СО СКРУГЛЕННЫМИ УГЛАМИ	
tft.drawRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16 color);	Рисование прямоугольника со скругленными углами
tft.fillRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16 color);	заливка прямоугольника
МОНОХРОМНОЕ РАСТРОВОЕ ИЗОБРАЖЕНИЕ	
drawBitmap(int16_t x, int16_t y, uint8_t *bitmap, int16_t w, int16_t h, uint16_t color);	Рисование небольшой монохромного (одноцветного) изображения. Bitmap располагается в памяти программ.
ВСЯ ЗОНА ДИСПЛЕЯ	
tft.fillScreen(uint16_t color);	заливка дисплея
tft.fillScreen(BLACK);	сброс (очистка) дисплея

Изменение шрифтов

Описание шрифтов выводимых на экран символов находится в glcdfont.c файлах библиотек Arduino. В файле указывается размер шрифтов, код первого символа, количество символов и приводится таблица шрифтов.

Таблица шрифтов фиксированной ширины, например, размером 5 x 7 пикселей с первого символа N 0x20 (32) по 0x7F (127) символ выглядит следующим образом

```
static const unsigned char font[] PROGMEM = {
        0x00, 0x00, 0x00, 0x00, 0x00,// (space)
        0x00, 0x00, 0x5F, 0x00, 0x00,// !
...
        0x3C, 0x4A, 0x49, 0x49, 0x30,// 6
```

```
        0x01, 0x71, 0x09, 0x05, 0x03,// 7
        0x36, 0x49, 0x49, 0x49, 0x36,// 8
...
        0x7E, 0x11, 0x11, 0x11, 0x7E,// A
        0x7F, 0x49, 0x49, 0x49, 0x36,// B
        0x3E, 0x41, 0x41, 0x41, 0x22,// C
...
        0x08, 0x08, 0x2A, 0x1C, 0x08,// ->
        0x08, 0x1C, 0x2A, 0x08, 0x08 // <-
};
```

где в каждой строке находится битовая маска символа, например, как показано на Рис. 4.

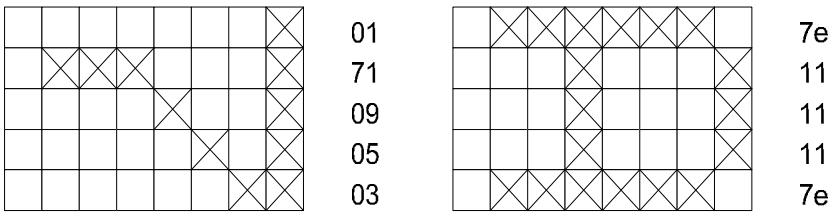


Рис. 4. Битовая маска символов 7 и A. В этой кодировка межсимвольный интервал добавляется автоматически.

Любой символ таблицы шрифтов можно изменить добавив или удалив биты отображаемых пикселов (см. Рис. 4). Это можно сделать при помощи текстового редактора или специальных программ, например, GLCDFontCreator.exe.

Установка и перекодировка цветов дисплея

Коды цветов графического дисплея A137 показаны ниже.

```
#define BLACK       0x0000
#define BLUE        0x001F
#define RED         0xF800
#define  GREEN      0x07E0
#define  CYAN       0x07FF
#define MAGENTA     0xF81F
#define YELLOW      0xFFE0
#define WHITE       0xFFFF
```

Видно, что любой цвет обозначается двумя байтами и кодировка всех цветов лежит в диапазоне 0x0000 до 0xFFFF, при чем, красному цвету выделено 5 бит (имеет 32 значения от 0 до 31), зеленому – 6 бит (64 значения) и синему – 5 бит.

Для определения кодировки дополнительных цветов можно использовать коды цветов, например, системы Microsoft Office и формулу перевода палитры 24-бит Microsoft в палитру 16-бит поддерживаемые библиотекой SWIFT дисплея A137.

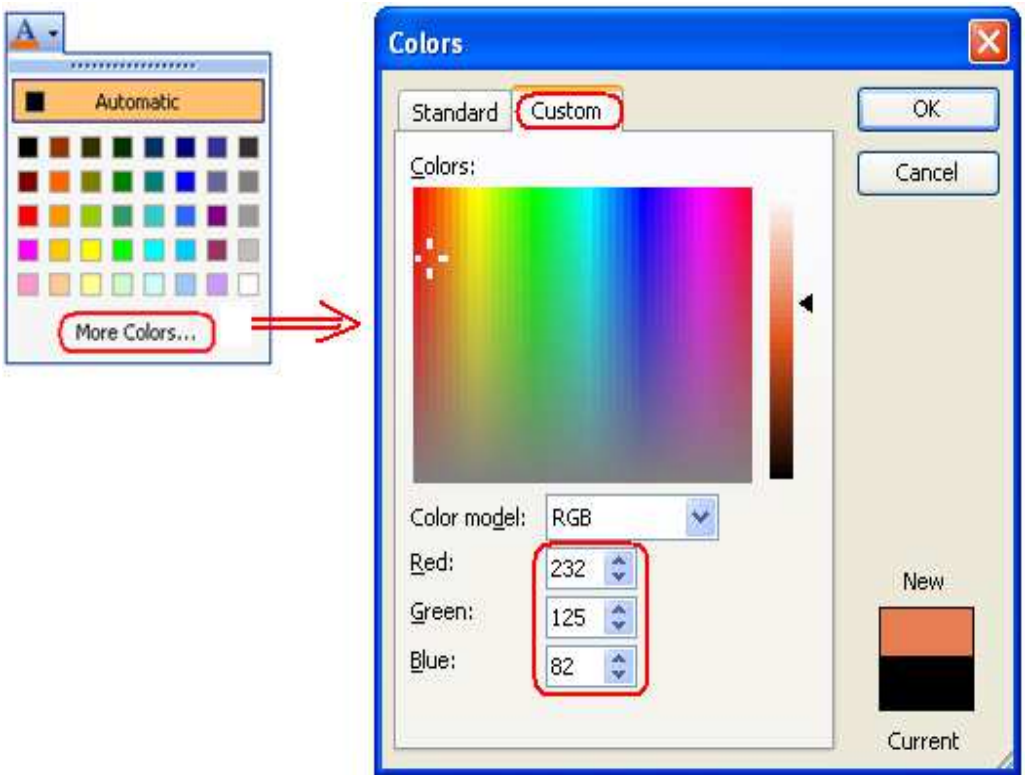


Рис. 5. Считывание RGB палитры 24 бит Windows в которой каждый цвет задается байтом (8 бит) от 0 до 255.

Преобразование кода палитры 24 бит в код палитры 16 бит можно выполнить по формуле

$$Код = floor(\frac{Red}{256} * 32) * 64 * 32 + floor(\frac{Green}{256} * 64) * 32 + floor(\frac{Blue}{256} * 32),$$

где цвета Red (красный) Green (зеленый) Blue (голубой) имеют значения от 0 до 255. Перевод десятичного кода в шестнадцатеричное или двоичное в МатЛАБ выполняется функциями dec2hex и dec2bin соответственно.

Определение координат сенсорной точки

Для обнаружения нажатия необходимо знать сопротивление между крайними координатами по X. Сопротивление можно измерить с помощью любого мультиметра. Величина сопротивления, например, 300 ом и размеры сенсорной зоны устанавливаются отношением

```
#define YP A1  // must be an analog pin
#define XM A2  // must be an analog pin
#define YM 7   // can be a digital pin
#define XP 6   // can be a digital pin
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);
```

Для работы с сенсорным экраном необходимо подключить библиотеку

```
#include <TouchScreen.h> // библиотека сенсорного экрана
```

Обычно измеренные координаты сенсорной зоны дисплея масштабируются в координатах зоны изображения.

На Рис. 6 показан результат перевода координат нажатия не откалиброванной сенсорной зоны 1024 x 1024 значений с

```
#define TS_MINX 150
#define TS_MINY 120
#define TS_MAXX 920
#define TS_MAXY 940
```

в координаты зоны изображения 240 x 320 точек с использованием следующих команд.

```
tft.fillRect(165, 160, 130, 52, ORANGE);  // построение прямоугольника
TSPoint p = ts.getPoint();  // считывание координат нажатия в зоне 1024 x 1024
p.x = tft.width()-(map(p.x, TS_MINX, TS_MAXX, tft.width(), 0)); // преобразование
p.y = tft.height()-(map(p.y, TS_MINY, TS_MAXY, tft.height(), 0)); // преобразование

// нажатие в зоне прямоугольника ?
if ((p.x > 15) and (p.x < 90) and (p.y > 120) and (p.y < 215)){…}
```

Несовпадение преобразованных координат нажатия на углы прямоугольника по оси X с координатами построения изображения прямоугольника получилось как 15 к 160 и 90 к 212. Несовпадение можно уменьшить проведя калибровку резистивной матрицы сенсорного экрана.

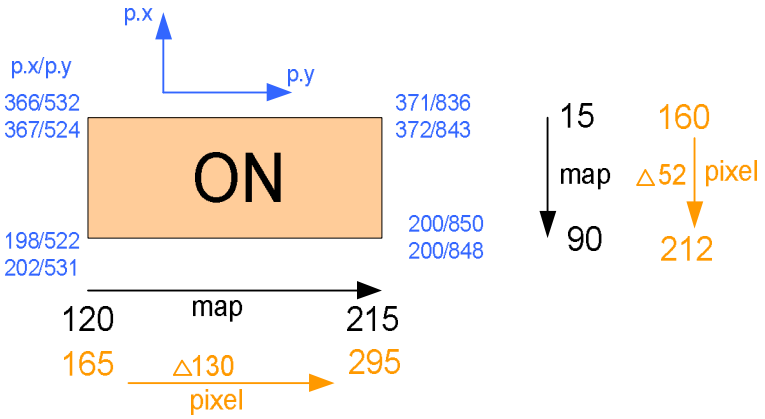


Рис. 6. Результат повторного считывания координат углов прямоугольника (числа, обозначенные голубым цветом) с использованием процедуры [p.x, p.y] = ts.getPoint() после нажатия на угловые точки. Прямоугольник в зоне 240x320 точек построен с использованием библиотечной функции tft.fillRect(165, 160, 130, 52, ORANGE). Желтые числа – усредненный результат перевода координат нажатий на углы в зоне 1024 x 1024 значений в координаты зоны изображения 240 x 320 точек полученный с использованием функции map. Режим направления печати задан командой tft.setRotation(1).

Другой вариант использования тестовых угловых координат для определения - произошло ли нажатие в прямоугольной зоне, позволяет отказаться от функции перевода (масштабирования):

```
TSPoint p = ts.getPoint();// считывание координат нажатия в зоне 1024 x 1024
// нажатие в зоне прямоугольника ?
if ((p.x > 200) and (p.x < 365) and (p.y > 525) and (p.y < 840)){…}
```

Особенности построения сенсорных клавиш

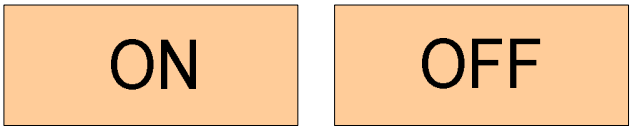
При использовании сенсорного дисплея для ввода реакции пользователя необходимо на экране показать зоны ввода (сенсорные клавиши). От варианта построения клавиш и считывая координат нажатия на экран зависит надежность срабатывания ввода. Рассмотрим два варианта организации ввода через дисплей A137.

1. Одна клавиша используется для ввода двух состояний, например, ВКЛ./ВЫКЛ. (ON/OFF)
2. Используются две клавиши: одна для ON, а другая для OFF.

В первом варианте за время нажатия происходило многократное переключение клавиши с ON на OFF и обратно, что затрудняло установку заданного положения клавиши. При введении 100 мс задержки между переключениями потребовалось значительно увеличить силу нажатия и, зачастую, требовалось многократное нажатие для изменения состояния клавиши.



Использование двух клавиш увеличило зону экрана занятую вводом, однако в этом варианте не зависимо от задержки происходило уверенное переключение даже при минимальной силе нажатия.



Работа с микро SD картой

Микро SD карата может использоваться для отображения заранее подготовленных графических изображений и хранения произвольных данных.

Таблица 2. Примеры характеристик bmp файлов для дисплея TFT LCD A137.

Bmp file	Width, pixels	Height, pixels	Horizontal resolution, dpi	Vertical resolution, dpi	Bit depth	Size, kB
woof*	240	320	71	71	24	230.5
miniwoof	120	160	71	71	24	57.7
test	240	240	71	71	24	230.5

* параметры полноэкранного изображения

Конвертирование 24-bit bitmaps в 16-bit формат (5 бит красный , 6 бит зеленый, 5 бит голубой) позволяет обеспечивать прямую загрузку в LCD полноэкранных изображений, что существенно увеличивает скорость загрузки. Код конвертора можно найти по ссылке [2].

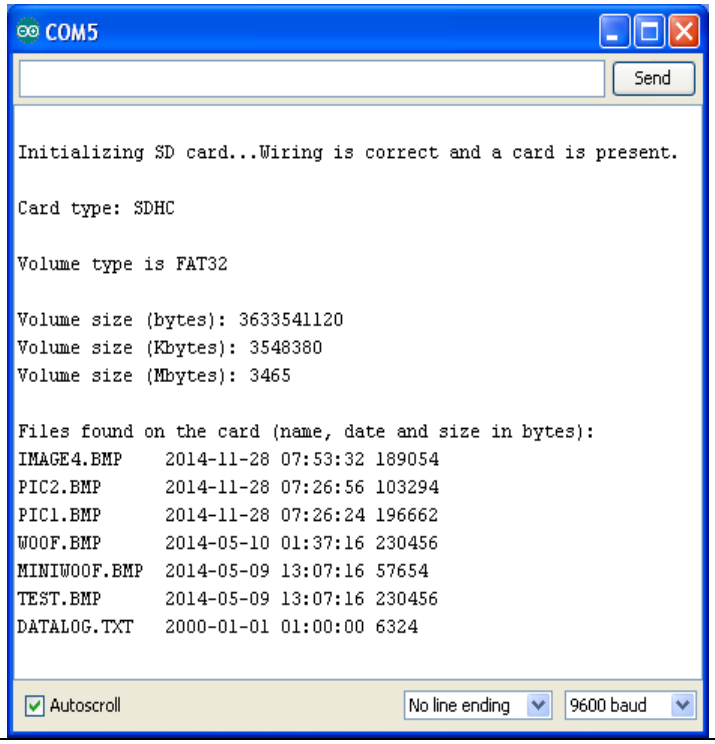
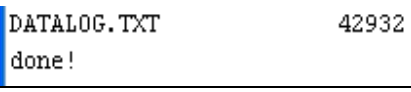
В программе работы с микро SD картой необходимо подключить библиотеку.

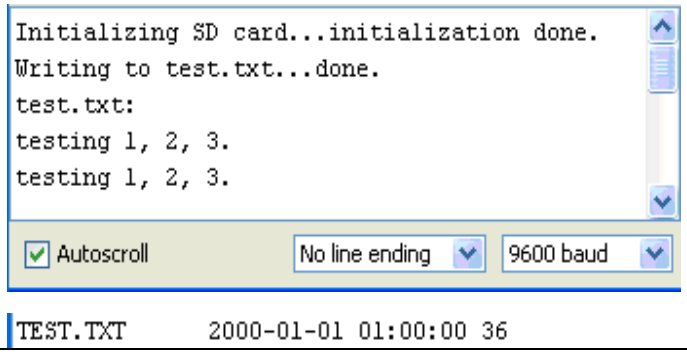
```
#include <SD.h> // библиотека SD карты
```

SD библиотека пакета arduino-1.0.6 содержит примеры работы с SD картой (см. Таблица 3).

Внимание! Для работы примеров с SD картой дисплея A137 необходимо исправить const int chipSelect = 4 на const int chipSelect = **10**; и if (!SD.begin(4)) на if (!SD.begin(**10**))

Таблица 3. Примеры SD библиотеки arduino-1.0.6\libraries\SD\examples\

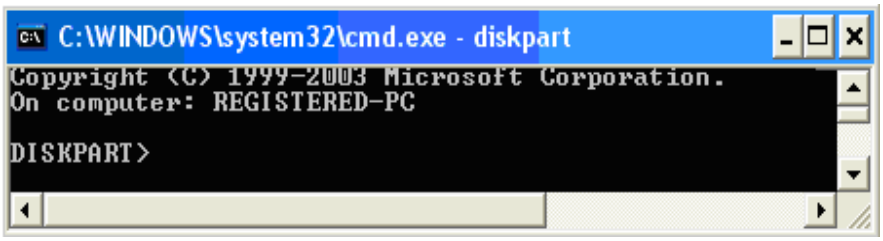
Название примера	**Назначение**
CardInfo.ino	Считывание и передача в последовательный канал параметров и содержимого SD карты, например, 
Datalogger.ini	Выполнение циклической записи показаний трех АЦП контроллера в файл SD карты "datalog.txt" с параллельной передачей показаний АЦП в последовательный канал *. 
DumpFile.ino	Считывание данных файла "datalog.txt" SD карты и передача считанных данных в последовательный канал.

Files.ino	Создание и удаление файла SD карты.
listfiles.ino	Передача списка файлов SD карты в последовательный канал.
ReadWrite.ino	Запись строки "testing 1, 2, 3." в файл "test.txt" SD карты, чтение записанных данных и передача данных в последовательный канал. 

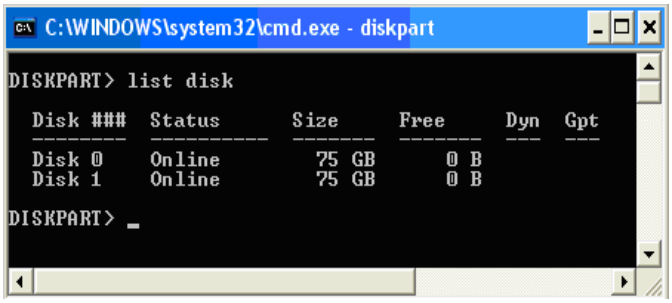
* Файл размером 76 КБ был записан примерно за 70 секунд со скоростью ~1100 байт/с. Параллельно с записью выполнялась передача данных в канал последовательной передачи на скорости 9600 бит/с. Исключение циклической передачи данных в последовательный канал параллельно с записью данных в файл увеличило скорость записи до 1375 байт/с (110 КБ за 80 с).

В некоторых случаях программа не может записать данные на SD карту, поскольку карта защищена от записи. Снять защиту от записи SD карты можно в следующей последовательности.

1. На компьютере с устройством к которому подключена карта откройте интерпретатор командной строки cmd (start > Run > Open > cmd)
2. Введите команду diskpart



3. Распечатайте список дисководов командой list disk



4. Введите номер считывателя SD карты командой list disk select disk 1. Необходимо в команде на месте 1 указать номер вашего устройства, распечатанный в списке (см. п.3).
5. Снимите защиту от записи: attributes disk clear readonly
6. Завершите работу в cmd. Введите exit , затем еще раз exit

Управление боковой подсветкой

После доработки разводки электрической платы дисплея яркость подсветки (0 .. 255 уровней) можно регулировать ШИМ сигналом порта DIO Arduino UNO. Управляющий ШИМ должен подключаться к светодиодам подсветки через транзисторный усилитель тока, поскольку выходной ток порта DIO не должен превышать 40 мА, тогда как, для регулировки подсветки может потребовать ток 100 мА.

В Интернет приводятся варианты и программного релейного включения/выключения TFT подсветки дисплеев после доработки библиотек. Варианта для дисплея A137 найти не удалось.

Вот пример [3] программы трехуровневого (100%, 50%, 0%, 100%, ...) ежесекундного изменения подсветки 3.3 В при питании от источника 5В.

```
#define PIN_BACKLIGHT 8
void SetBacklightPercent(int percent)
{
        if (percent > 100)
                percent = 100;
        else if (percent < 0)
                percent = 0;
        ///(3.3v/5v) * 255 = 168.3
        analogWrite(PIN_BACKLIGHT, (int) round(168.3 * (percent/100.0)));
}

void setup()
{       pinMode(PIN_BACKLIGHT, OUTPUT);
}

void loop()
{
        SetBacklightPercent(100);
        delay(1000);

        SetBacklightPercent(50);
        delay(1000);
```

```
        SetBacklightPercent(0);
        delay(1000);
}
```

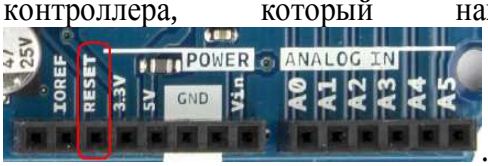
Высвобождение входа ADC4 для нужд пользователя

В оригинальной версии порт LCD_RESET дисплея A 137 подключается к порту A4 контроллера Arduino UNO. Через порт A4 контроллер производит “сброс” дисплея – установку его в начальное состояние. Доработка соединения дисплей – контроллер позволяет использовать порт A4 для ввода аналоговых сигналов в контроллер. Доработка включает следующие шаги.

В файле SWIFT.cpp библиотеки SWIFT необходимо исключить команды установки работы аналогового порта A4 в режим дискретного выхода: pinMode(LCD_RESET, OUTPUT); и команды записи в дискретный порт A4: digitalWrite(LCD_RESET, HIGH); и digitalWrite(LCD_RESET, LOW);

Описанное выше исключение команд-установок порта A4 блокирует переход порта в режим цифрового выхода которое выполнялось в момент формировании tft объекта командой: SWTFT tft. Теперь порт A4 постоянно работает в режиме АЦП – приёма аналоговых сигналов, который устанавливается “по умолчанию” при включении или перезагрузке Arduino. Однако, если на вход АЦП подать сигнал низкого уровня, то произойдет сброс дисплея. Чтобы избежать неконтролируемого сброса, необходимо физически отключить линию LCD_RST дисплея 

от порта A4 контроллера Arduino UNO (показан на рисунке ниже) и подключить его к контакту RESET контроллера, который находится рядом с контактом +3.3V:



.

Мерцание выводимых на дисплей значений

При частом выводе переменной на дисплей возникает эффект мерцания, что делает дискомфортным чтение экрана. Для устранения эффекта мерцания необходимо удалять и выводить только обновляемые символы (цифры числа). Если цифра числа, которое необходимо вывести на экран, не отличается от цифры той же позиции выведенного числа, то вывод такой цифры необходимо заблокировать.

Ниже дан пример подавления мерцания при выводе на дисплей значений 10-разрядного АЦП A5 каждые 100 мс.

```
#include <Adafruit_GFX.h>    // Графическая библиотека
#include <SWTFT.h>           // Библиотека аппаратных средств дисплея
```

```
// обозначение цветов
#define BLACK   0x0000
#define GREEN   0x07E0

SWTFT tft;

const int adc_5 = A5;  // номер используемого АЦП
int adc_5_sample, adc_5_sample_prev = 9999;
unsigned long set_time = 0;
int countdigits[] = {0, 0, 0, 0}; // массив цифр последнего числа
int prevdigits[] = {0, 0, 0, 0}; // массив цифр предыдущего числа
int digitpos[] = {175, 205, 235, 265}; // X координаты цифр
int x = 0;

void setup() {
  uint16_t identifier = tft.readID();
  tft.begin(identifier);

  tft.setRotation(1);  // направление вывода строк - по длине экрана
  tft.fillScreen(BLACK); // сброс экрана – заливка черным
  tft.setTextSize(5);
}

void loop() {
  unsigned long time = millis();  // чтение таймера
  // запуск цикла каждые 100 мс
  if (time > set_time) {
    set_time = set_time + 100;

  // чтение АЦП:
  adc_5_sample = analogRead(adc_5);

// обновление показаний если новое и предыдущие значения не совпадают
    if (adc_5_sample != adc_5_sample_prev) {
```

```
// разложить считанное число на цифры
if (adc_5_sample > 999) countdigits[0] = (adc_5_sample / 1000) % 10;
if (adc_5_sample > 99) countdigits[1] = (adc_5_sample / 100) % 10;
if (adc_5_sample > 9) countdigits[2] = (adc_5_sample / 10) % 10;
countdigits[3] = adc_5_sample % 10;

// разложить предыдущее число на цифры
if (adc_5_sample_prev > 999) prevdigits[0] = (adc_5_sample_prev / 1000) % 10;
if (adc_5_sample_prev > 99) prevdigits[1] = (adc_5_sample_prev / 100) % 10;
if (adc_5_sample_prev > 9) prevdigits[2] = (adc_5_sample_prev / 10) % 10;
prevdigits[3] = adc_5_sample_prev % 10;

// поцифровое сравнение чисел
for(x=0; x < 4; x++){

 // обновить цифру если она отличается
 if(countdigits[x] != prevdigits[x]){
  // стереть старую цифру
  tft.setCursor(digitpos[x], 114);
  tft.setTextColor(BLACK);
  tft.print(prevdigits[x]);

  // напечатать новую цифру если это не ноль в старшем разряде
  tft.setTextColor(GREEN);
  if((x == 0) and (adc_5_sample > 999)) {
   tft.setCursor(digitpos[x], 114);
   tft.print(countdigits[x]);
  }
  if((x == 1) and (adc_5_sample > 99)) {
   tft.setCursor(digitpos[x], 114);
   tft.print(countdigits[x]);
  }
  if((x == 2) and (adc_5_sample > 9)) {
   tft.setCursor(digitpos[x], 114);
   tft.print(countdigits[x]);
  }
```

```
        if(x == 3) {
          tft.setCursor(digitpos[x], 114);
          tft.print(countdigits[x]);
        }
       }
      }
     }
     // сохранить считанное значение
     adc_5_sample_prev = adc_5_sample;
   }
}
```

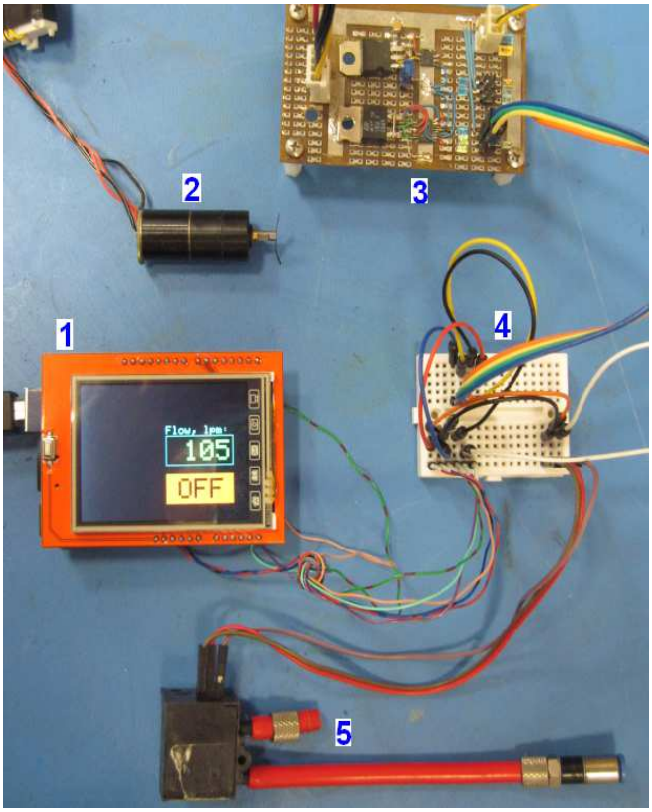


Рис. 7. Пример компонентов макета локальной системы управления газовым потоком: 1 – контроллер Arduino UNO, накрытый цветным графическим сенсорным дисплеем; 2 - шаговый двигатель AM0820; 3 - драйвер BA6845FS (усилители тока шагового двигателя по мостовой схеме [4]); 4 – коммутационная плата; 5 – датчик потока газа D6F-P0010A1.

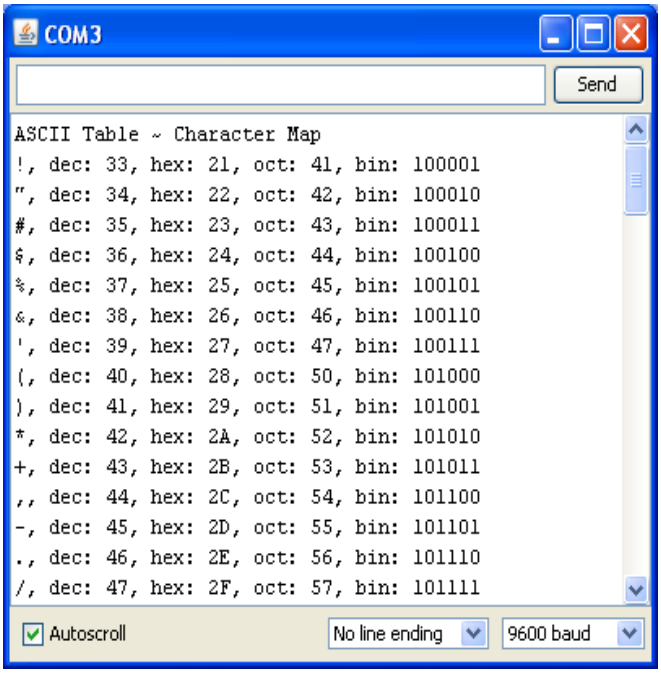
ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Отображение параметров контроллера и ввод данных через монитор последовательного канала.

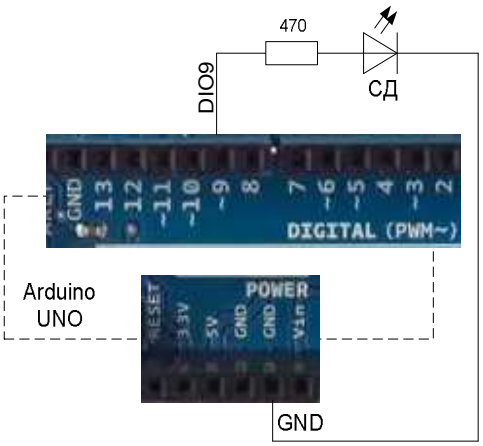
1. Запустите программу arduino.exe
2. Загрузите (Меню > File > Examples > 04.Communication > ASCIITable) пример передачи данных Arduino в канал COM - последовательной передачи данных.

   c:\arduino-1.0.6\examples\04.Communication\ASCIITable\ASCIITable.ino

3. Запустите пример  и откройте монитор: Serial Monitor 



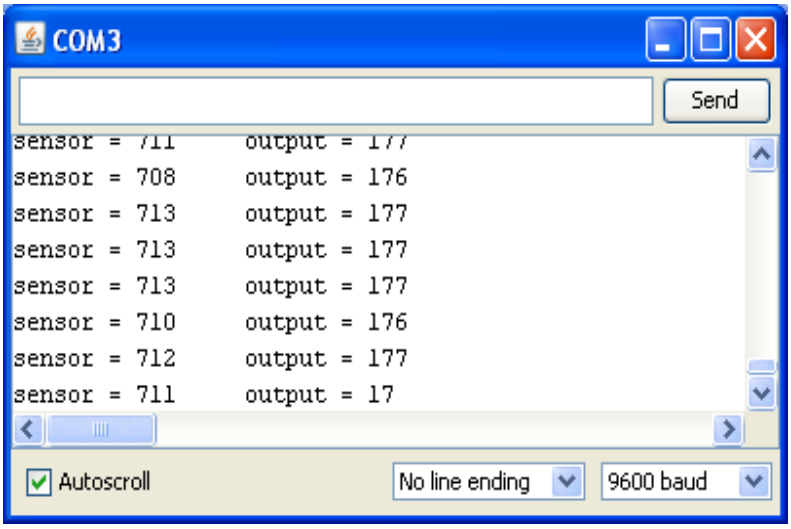
4. Рассмотрите организацию отображения данных контроллера, используя коды программы и выводимые на монитор данные.
5. К порту DIO 9 выключенного Arduino UNO подключите светодиод.



6. Загрузите (Меню > File > Examples > 03.Analog > AnalogInOutSerial) пример считывания показаний 10-бит АЦП (A0) Arduino, масштабирования данных до 8 бит функцией map программы и передачи полученных байт в канал COM.

   c:\arduino-1.0.6\examples\03.Analog\AnalogInOutSerial\AnalogInOutSerial.ino

7. Запустите пример и откройте Serial Monitor 



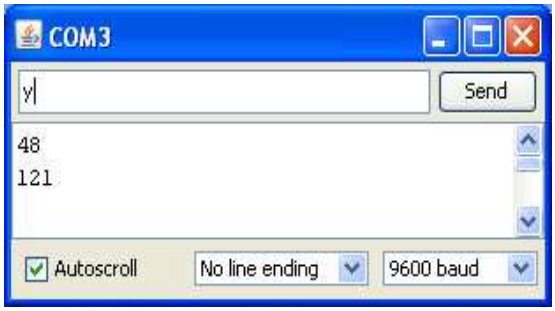
8. Рассмотрите организацию отображения данных АЦП, используя коды программы, выводимые на монитор данные и свечение светодиода. Установку светодиода в крайние состояния Вкл./Выкл. можно можно получить подключением заданного напряжения ко входу АЦП A0 разъема ANALOG IN контроллера с контактов 5V (5В) / GND (0В) разъема POWER.

9. Загрузите (Меню > File > Examples > 04.Communication > Dimmer) пример передачи данных контроллеру Arduino через Serial Monitor и COM канал и отображение передаваемого кода свечением светодиода подключенного к DIO 9.

   c:\arduino-1.0.6\examples\04.Communication\ Dimmer\Dimmer.ino

10. Используя первые два примера доработайте пример Dimmer.ino так, чтобы монитор отображал данные принятые контроллером.

```
Serial.println(brightness);  // добавлена передача монитору
```



Задание 2. Отображение параметров контроллера в среде Simulink компьютера в реальном времени по тактам контроллера Arduino без использования специальных средств MatLAB для обеспечения работы систем в реальном времени rtwintgt (Real-Time Windows Target).

Примечание. Преимущество рассматриваемого соединения в котором реальное время задается контроллером Arduino в сравнении с вариантом использования rtwintgt состоит в том, что

- для запуска модели в режиме реального времени не надо тратить время на компиляцию с rtwintgt.
- можно использовать большее количество блоков библиотеки Simulink
- случается, что модель Simulink не может успешно пройти компиляцию с rtwintgt.

1. Запустите МатЛАБ.
2. В среде Simulink постройте следующую модель

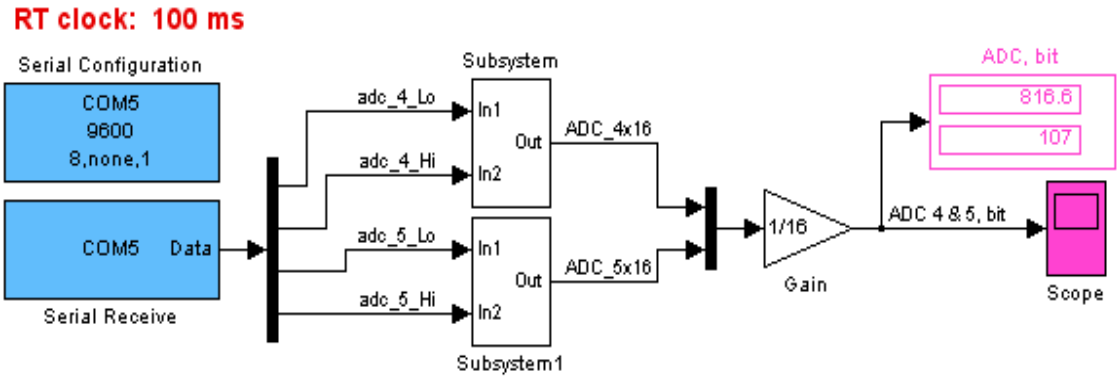


Рис. 8. Модель считывания и отображения данных последовательного канала. В модели вычисляется дробная часть усредненных показаний АЦП контроллера с разрешением 1/16 бит.

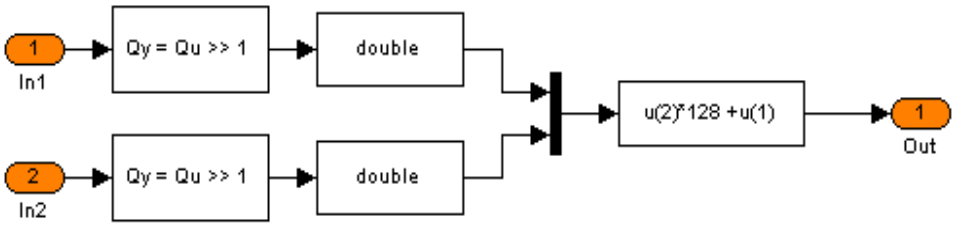


Рис. 9. Структура блоков Subsystem(s) модели. Восстановление двухбайтового числа.

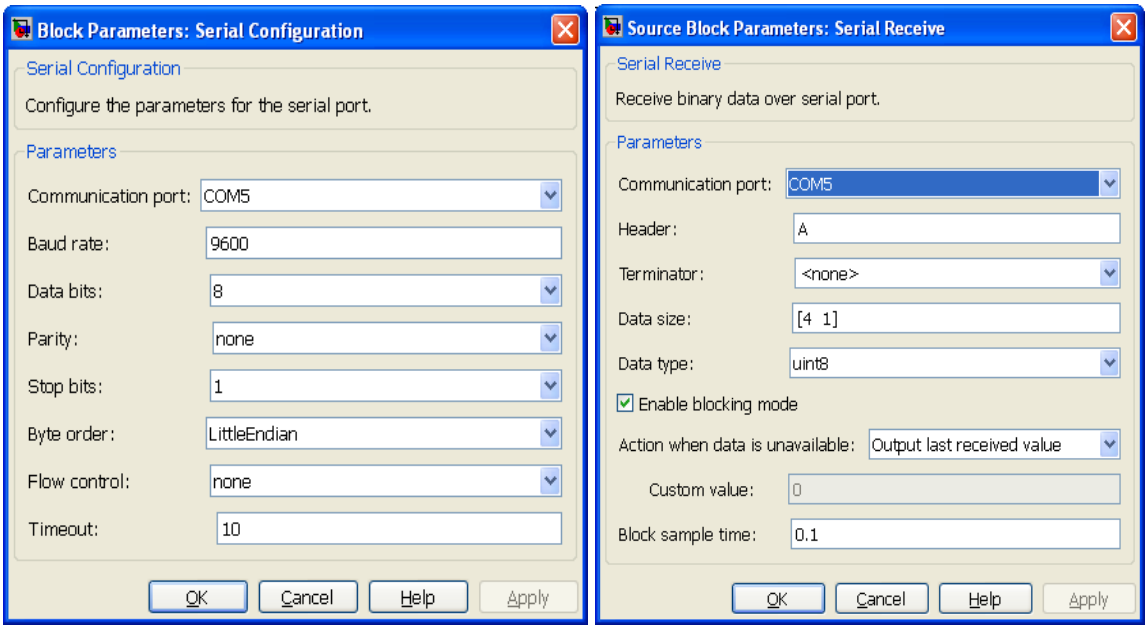


Рис. 10. Параметры настройки последовательного канала COM модели для приема данных контроллера Arduino на частоте 9600 бод пакетом из пяти байт: 4 байта данных + заголовок - символ A.

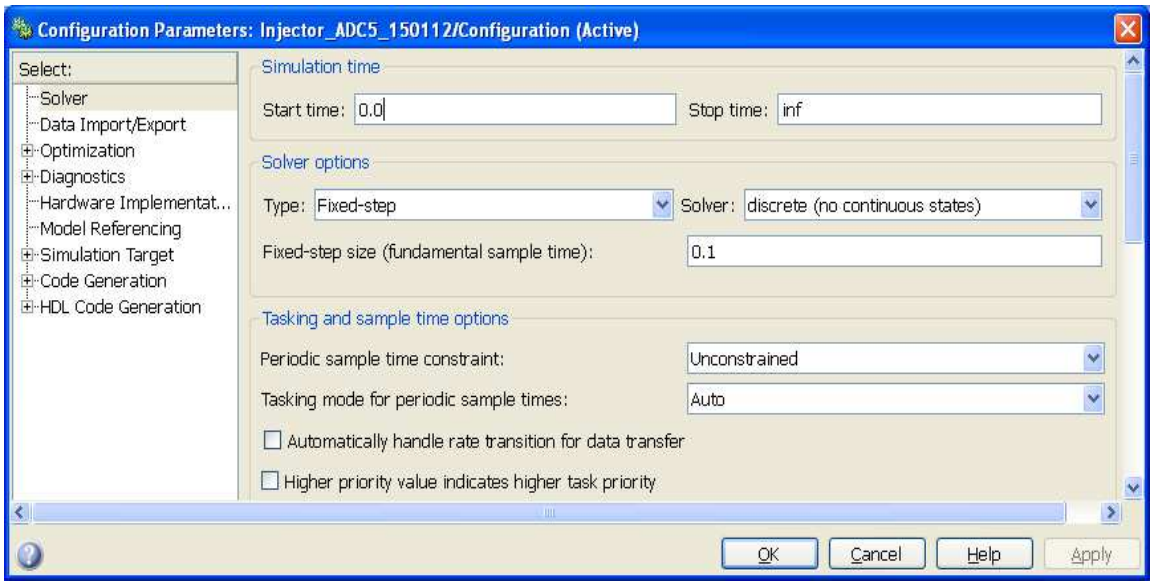


Рис. 11. Параметры моделирования с шагом 0.1 сек. Шаг моделирования должен совпадать с периодом передачи данных в последовательный канал контроллером Arduino.

3. Создайте программу считывания показаний двух АЦП контроллера Arduino, суммирования показаний каждого АЦП и передачу усредненных показаний в последовательной канал. Для повышения точности усредненные показания перед передачей должны быть увеличены в 16 раз. Байты данных должны передаваться пакетом с уникальным заголовком - символом “A”.

```
const int adc_4 = A4, adc_5 = A5;
int adc_4_sample, adc_5_sample;
unsigned long adc_4_sum = 0, adc_5_sum = 0;
unsigned long set_time = 0;
int loop_num = 0;

void setup() {
 Serial.begin(9600);
}
void loop() {
 // read ADC with maximum rate:
 adc_4_sample = analogRead(adc_4);
 adc_4_sum = adc_4_sum + adc_4_sample;
 loop_num = loop_num + 1; // ~ 120 loops per 100 ms

 adc_5_sample = analogRead(adc_5);
 adc_5_sum = adc_5_sum + adc_5_sample;

 unsigned long time = millis();  // read time in ms
 // run calculation each 100 ms
 if (time > set_time) {
  set_time = set_time + 100;  // dt = 100 ms

  adc_4_sample = (adc_4_sum * 16) / loop_num;
  adc_5_sample = (adc_5_sum * 16) / loop_num;

  // bytes for Simulink
  byte adc_4_Hi = ((adc_4_sample >> 6) & 0xFE); // high byte
  byte adc_4_Lo = ((adc_4_sample << 1) & 0xFE); // low byte
  byte adc_5_Hi = ((adc_5_sample >> 6) & 0xFE);
  byte adc_5_Lo = ((adc_5_sample << 1) & 0xFE);

  //Send data into COM port
  Serial.print("A"); // it is header
```

```
    Serial.write(adc_4_Lo);
    Serial.write(adc_4_Hi); // output byte: uint8
    Serial.write(adc_5_Lo);
    Serial.write(adc_5_Hi); // output byte: uint8
  // reset averages
    adc_4_sum = 0;
    adc_5_sum = 0;
    loop_num = 0;
   }
  }
```

4. Запустите программу Arduino, затем модель Simulink.

5. Откройте графопостроитель (блок Scope) модели. Наблюдайте сигналы АЦП контроллера Arduino в среде Simulink в реальном времени (период 100 мс).

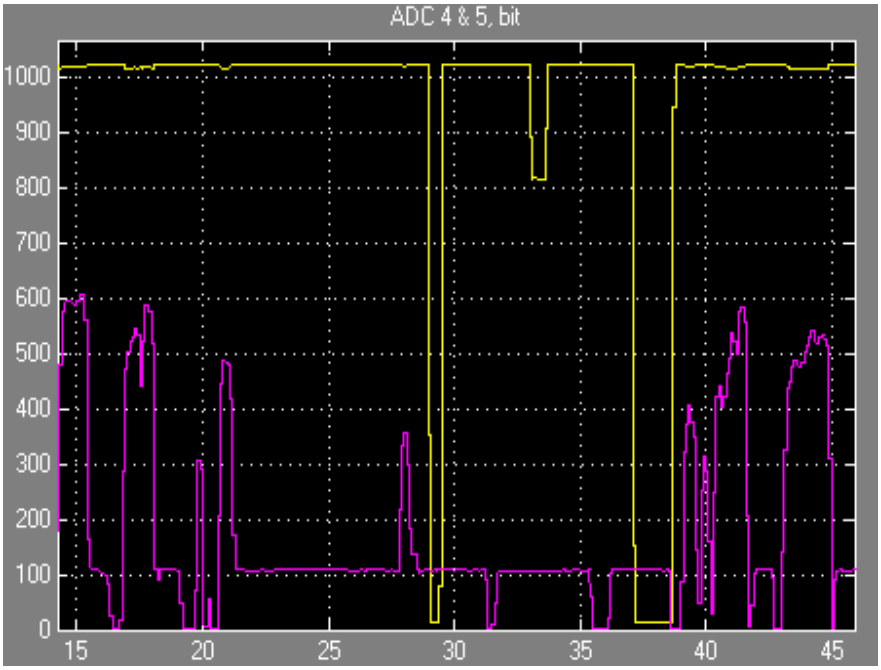


Рис. 12. Пример отображения сигналов 10 разрядных АЦП контроллера Arduino с разрешением 1/16 разряда.

Примечание. Через блок Serial Send (рис. 13) можно передавать данные Simulink контроллеру Arduino. Для приема данных, в программу контроллера следует включить следующие команды [5].

```
 if (Serial.available() > 0) {
   ctrl_num = Serial.read(); // 0 .. 255 from simulink
…
}
```

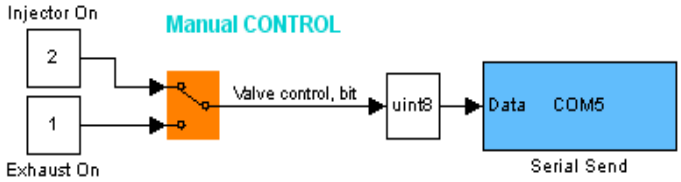


Рис. 13. Пример канал передачи данных Simulink контроллеру Arduino через COM порт.

Задание 3. Построение интерфейса пользователя на базе цветного графического сенсорного дисплея A137.

1. Создайте программу отображения показаний двух АЦП (A4 и A5) на доработанном дисплее A137 (с сигналом LCD_RESET, подключенным к порту RESET контроллера). Необходимо блокировать мерцание выводимых показаний (см. соответствующий раздел выше). Интерфейс должен включать две сенсорные кнопки. Одна – для очистки выводимых на дисплей показаний АЦП и остановки вывода, другая - для продолжения вывода показаний.

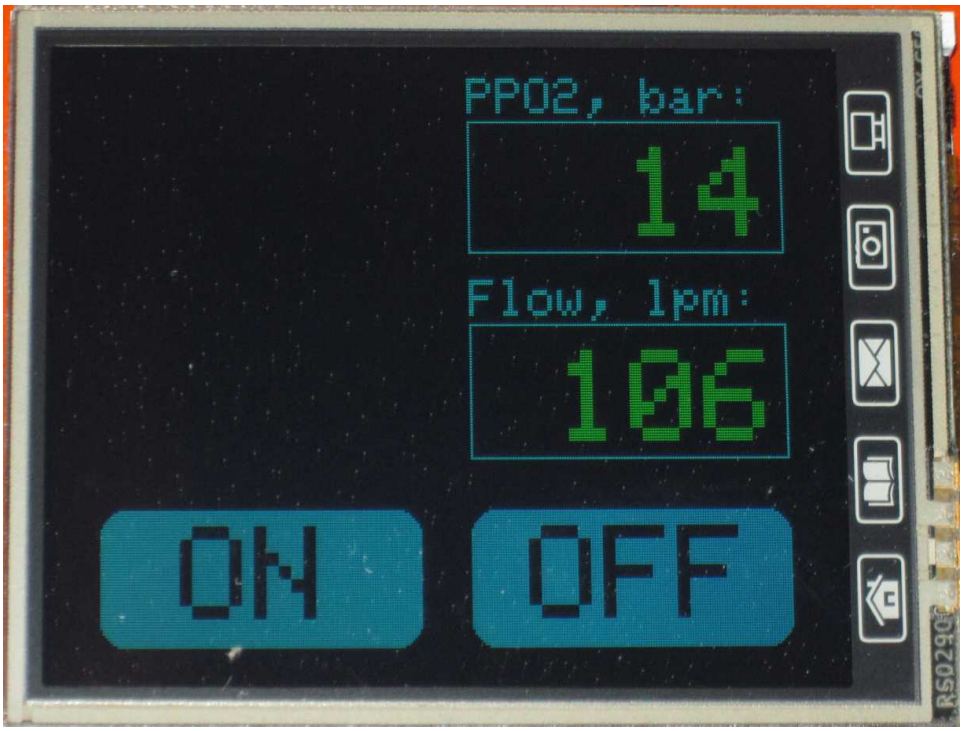


Рис. 14. Пример интерфейса пользователя контроллера Arduino UNO R3.

ПРИМЕР ПРОГРАММЫ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ:

```
#include <Adafruit_GFX.h>    // Core graphics library
#include <SWTFT.h>           // Hardware-specific library
```

```
#include <TouchScreen.h>

#define YP A1  // must be an analog pin
#define XM A2  // must be an analog pin
#define YM 7   // can be a digital pin
#define XP 6   // can be a digital pin

#define TS_MINX 150
#define TS_MINY 120
#define TS_MAXX 920
#define TS_MAXY 940

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

// Assign human-readable names to some common 16-bit color values:
#define BLACK   0x0000
#define GREEN   0x07E0
#define CYAN    0x07FF
#define SKY     0x067F

SWTFT tft;

const int adc_4 = A4, adc_5 = A5;
word adc_4_sample, adc_5_sample; //as unsigned int 0 .. 65535
word adc_4_sample_prev, adc_5_sample_prev;

unsigned long set_time = 0;
word loop_num = 0;
boolean button_state = 1;
boolean first_read = 1;

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  Serial.println(F("TFT User Interface Test"));

//  tft.reset();
  uint16_t identifier = tft.readID();
  tft.begin(identifier);
```

```
  tft.setRotation(1);  // screen rotation 0 .. 3
  tft.fillScreen(BLACK);

  //Draw static screen elements
  // Data output windows
  tft.drawRect(170, 25, 130, 52, CYAN);
  tft.drawRect(170, 104, 130, 52, CYAN);
  tft.setTextColor(CYAN);
  tft.setTextSize(2);
  tft.setCursor(170, 8);
  tft.print("PPO2, bar:");
  tft.setCursor(170, 87);
  tft.print("Flow, lpm:");

  // Buttons ON and OFF
  tft.setTextSize(5);
  tft.setTextColor(BLACK);
  tft.fillRoundRect(20, 175, 130, 52, 10, SKY);
  tft.setCursor(52, 181);
  tft.print("ON");
  tft.fillRoundRect(170, 175, 130, 52, 10, SKY);
  tft.setCursor(193, 181);
  tft.print("OFF");
}

#define MINPRESSURE 10
#define MAXPRESSURE 1000

void loop() {
  // read ADC:
  adc_4_sample = analogRead(adc_4);
  adc_5_sample = analogRead(adc_5);

  // read touch screen
  TSPoint p = ts.getPoint();
  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);
```

```
// pressure of 0 means no pressing!
if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {
  // button On: run output
  if ((p.x > 160) and (p.x < 320) and (p.y > 170) and (p.y < 475)){
    button_state = 1;
    first_read = 1;
  }
  // button Off: clear output windows, stop output
  if ((p.x > 160) and (p.x < 320) and (p.y > 525) and (p.y < 840)){
    button_state = 0;
    tft.fillRect(170, 25, 130, 52, BLACK); // clear
    tft.drawRect(170, 25, 130, 52, CYAN);  // restore botton frame
    tft.fillRect(170, 104, 130, 52, BLACK);
    tft.drawRect(170, 104, 130, 52, CYAN);
  }
}
unsigned long time = millis();
if (time > set_time) {
  set_time = set_time + 100;

  if (button_state == 1) {
    // just after run,  previous sample should be
    // different from the latest sample
    if (first_read == 1) {
      adc_4_sample_prev = 9999 - adc_4_sample;
      adc_5_sample_prev = 9999 - adc_5_sample;
      first_read = 0;
    }
    if (adc_4_sample != adc_4_sample_prev) {
      print_tft(adc_4_sample, adc_4_sample_prev, 35);
    }
    adc_4_sample_prev = adc_4_sample;

    if (adc_5_sample != adc_5_sample_prev) {
      print_tft(adc_5_sample, adc_5_sample_prev, 114);
    }
    adc_5_sample_prev = adc_5_sample;
  }
}
```

```
}

void print_tft(word sample_latest, word sample_prev, word y) {
  byte countdigits[] = {0, 0, 0, 0};
  byte prevdigits[] = {0, 0, 0, 0};
  word digitpos[] = {175, 205, 235, 265};
  byte x;
  word p;

  for(x=0; x < 4; x++){
    p = round(pow(10.0,3-x)); // need round of .9999... float
    if (sample_latest > p-1) {countdigits[x] = (sample_latest / p) % 10;}
    if (sample_prev > p-1) {prevdigits[x] = (sample_prev / p) % 10;}
  }

  // Compare each digit to the value from the previous loop.
  // The digit will only be redrawn if it has changed.
  for(x=0; x < 4; x++){
    if(countdigits[x] != prevdigits[x]){
      // clear old digit
      tft.setCursor(digitpos[x], y);
      tft.setTextColor(BLACK);
      tft.print(prevdigits[x]);

      // print new digit in green
      p = round(pow(10,3-x));
      if(sample_latest > p-1) {
        tft.setCursor(digitpos[x], y);
        tft.setTextColor(GREEN);
        tft.print(countdigits[x]);
      }
      if(sample_latest == 0) {
        tft.setCursor(digitpos[3], y);
        tft.setTextColor(GREEN);
        tft.print(countdigits[x]);
      }
    }
  }
}
```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими средствами можно обеспечить ввод воздействий пользователя в локальную систему на базе контроллера Arduino?
2. Какими средствами можно обеспечить отображение и накопление параметров и переменных состояния локальной системы на базе контроллера Arduino?
3. Для решения каких задач целесообразно использовать недорогой цветной графический сенсорный дисплей с микро SD считывателем?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Adafruit GFX Graphics Library, created by Phillip Burgess
2. Justin. Bitmap Converter for Arduino LCD. http://misc.ws/2013/11/03/bitmap-converter-for-arduino-lcd/
3. How to control the display's backlight in the sketch? http://forum.arduino.cc/index.php?topic=125856.0.
4. Dr. Bob Davidov. Стенд для разработки алгоритмов высокоскоростного управления шаговым приводом. http://portalnp.ru/wp-content/uploads/2013/12/07.06_Stand-for-design-of-high-speed-stepper-motor-control-algorithms_Ed_1b1.pdf
5. Dr. Bob Davidov. Многоканальное устройство ввода и накопления аналоговых данных на базе MS Excel. http://portalnp.ru/2014/03/1762
6. Dr. Bob Davidov. Компьютерные технологии управления в технических системах http://portalnp.ru/author/bobdavidov.

Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

Мы - команда радиолюбителей. Начинали заниматься Arduino исключительно для себя. Однако со временем мы всё чаще стали сталкиваться с такой ситуацией: в России модули и контроллеры Arduino либо продают по заоблачным ценам, либо их вообще нет на рынке! И тогда мы решили помогать другим радиолюбителям и новичкам - так появился наш интернет-магазин – www.arduino55.ru

Arduino UNO R3 Original



750 руб.

Arduino UNO R3 CH340



650 руб.

Arduino NANO USB



500 руб.

Arduino MEGA 2560



1050 руб.

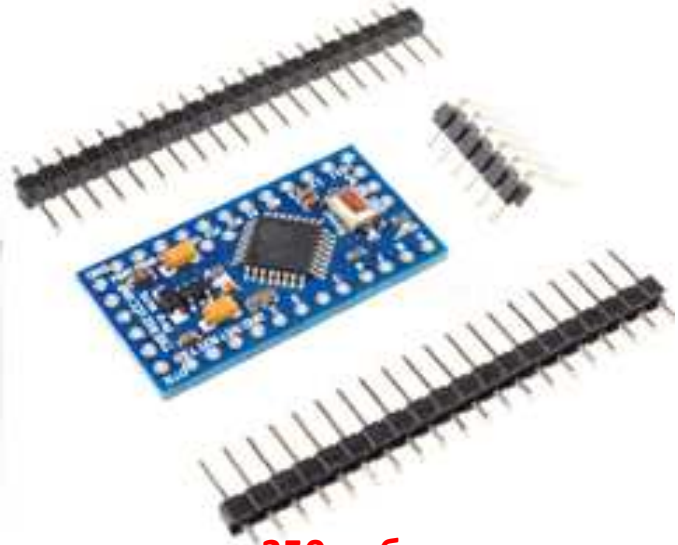
Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

Нам удаётся поддерживать низкую цену на радиодетали за счёт минимальной наценки.
Ведь наша цель – не разбогатеть с помощью интернет-магазина, а помочь таким же, как мы.
Нам нравится заниматься экспериментами и построением рабочих систем на базе Arduino.

Присоединяйтесь!

Arduino MINI PRO



350 руб.

Программатор (в т.ч. и для MINIPRO)



от 250 руб.

SHIELD – модули

Ethernet Shield W5100



550 руб.

GSM-модуль Geeetech SIM900



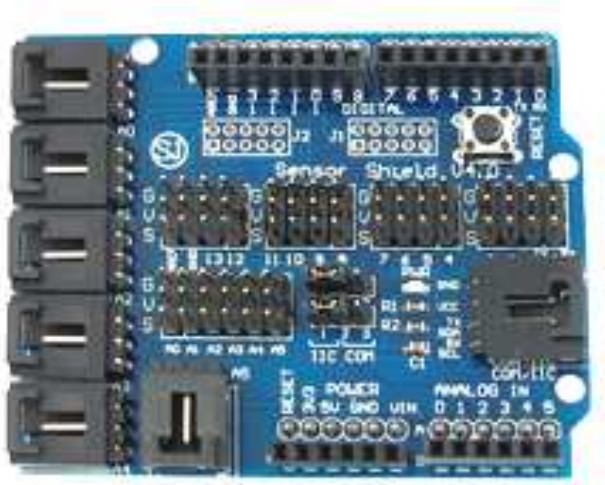
1800 руб.

Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

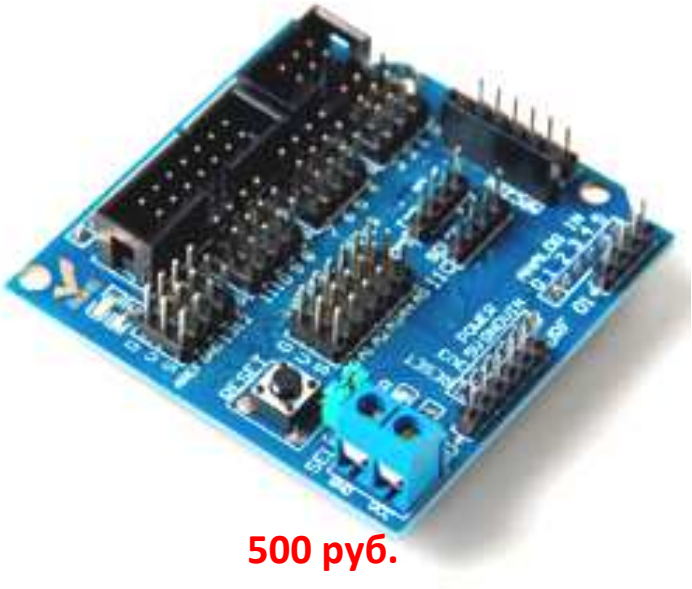
В нашем интернет-магазине присутствует более 150 единиц товара.
В том числе: модулей Arduino – более 50, сенсоров и датчиков – более 30.
С полным списком можно ознакомиться на сайте www.arduino55.ru

Sensor Shield V4



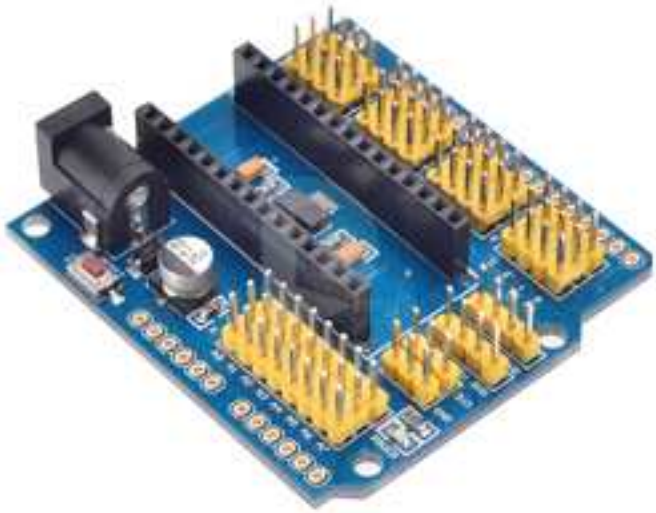
400 руб.

Sensor Shield V5



500 руб.

Sensor Shield V3 для NANO



400 руб.

2.4" SHIELD - разработки



300 руб.

Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

Экраны

2.4" TFT LCD Touch Panel TF Reader



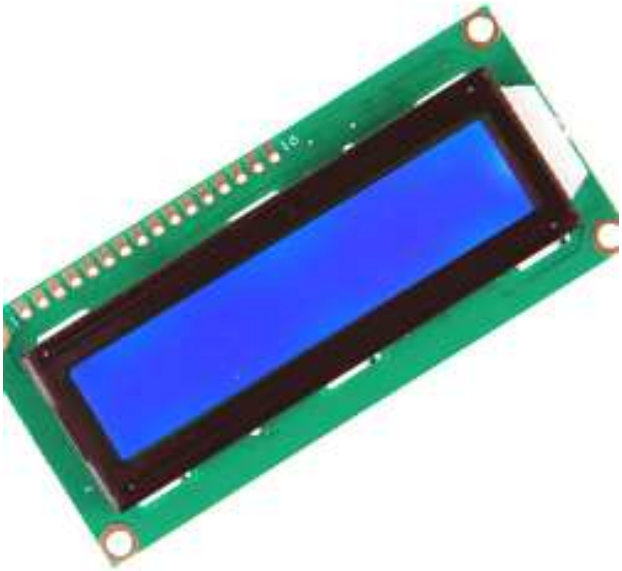
350 руб.

LCD 1602 Keypad Shield



400 руб.

Дисплей LCD 1602



200 руб.

84x48 LCD Module (Nokia 5110)



250 руб.

Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

2.2" SPI TFT LCD Module Display



450 руб.

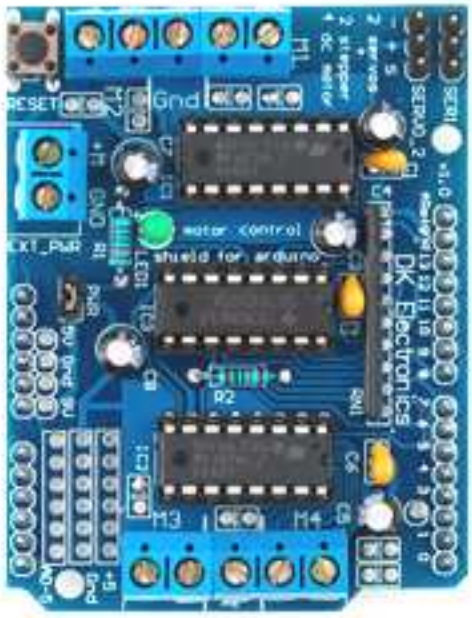
1.44" SPI TFT LCD Module Display



300 руб.

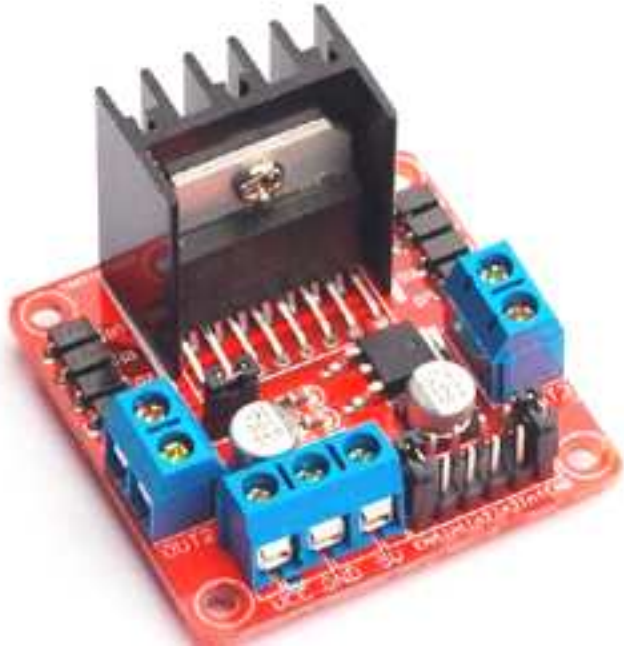
Драйверы и реле

Motor Drive Expansion Shield L293D



400 руб.

Driver L298N



250 руб.

Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

1 - канальный Relay Module



120 руб.

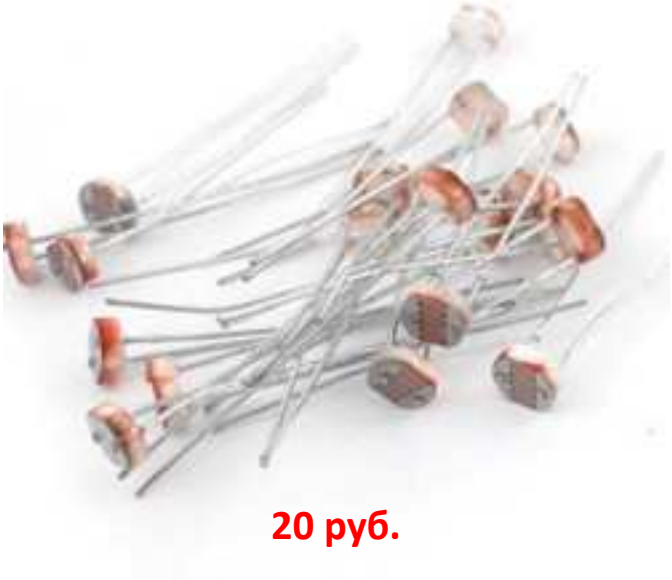
2 - канальный Relay Module



200 руб.

Сенсоры и датчики

Фоторезистор GL5516



20 руб.

ИК отражатель + улавливатель TCRT5000



20 руб.

Arduino55
интернет-магазин электронных компонентов

www.arduino55.ru
(3812) 481-486 +7(983) 528-23-96
E-mail: info@arduino55.ru

Инфракрасный датчик движения SR-501



120 руб.

Ультразвуковой измеритель расстояния HC-SR04



120 руб.

Наборы элементов

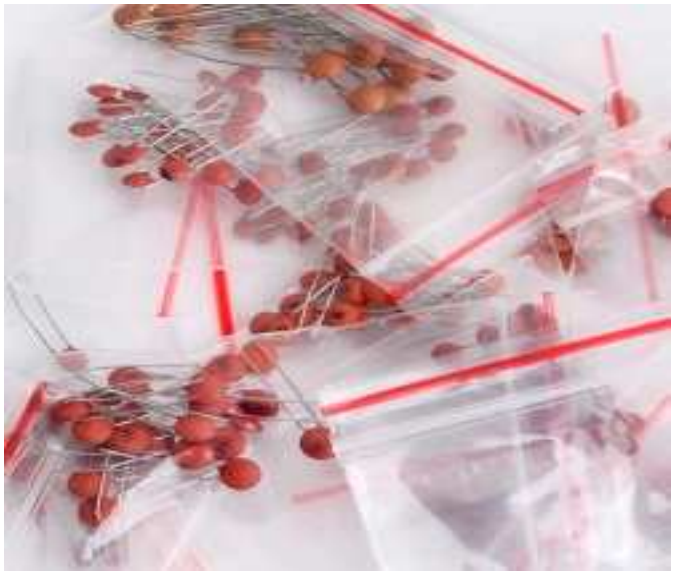
Резисторы в ассортименте



20 линеек по 5 штук = 100 резисторов

100 руб.

Конденсаторы в ассортименте



50 линеек по 5 штук = 250 конденсаторов

300 руб.

Транзисторы в ассортименте

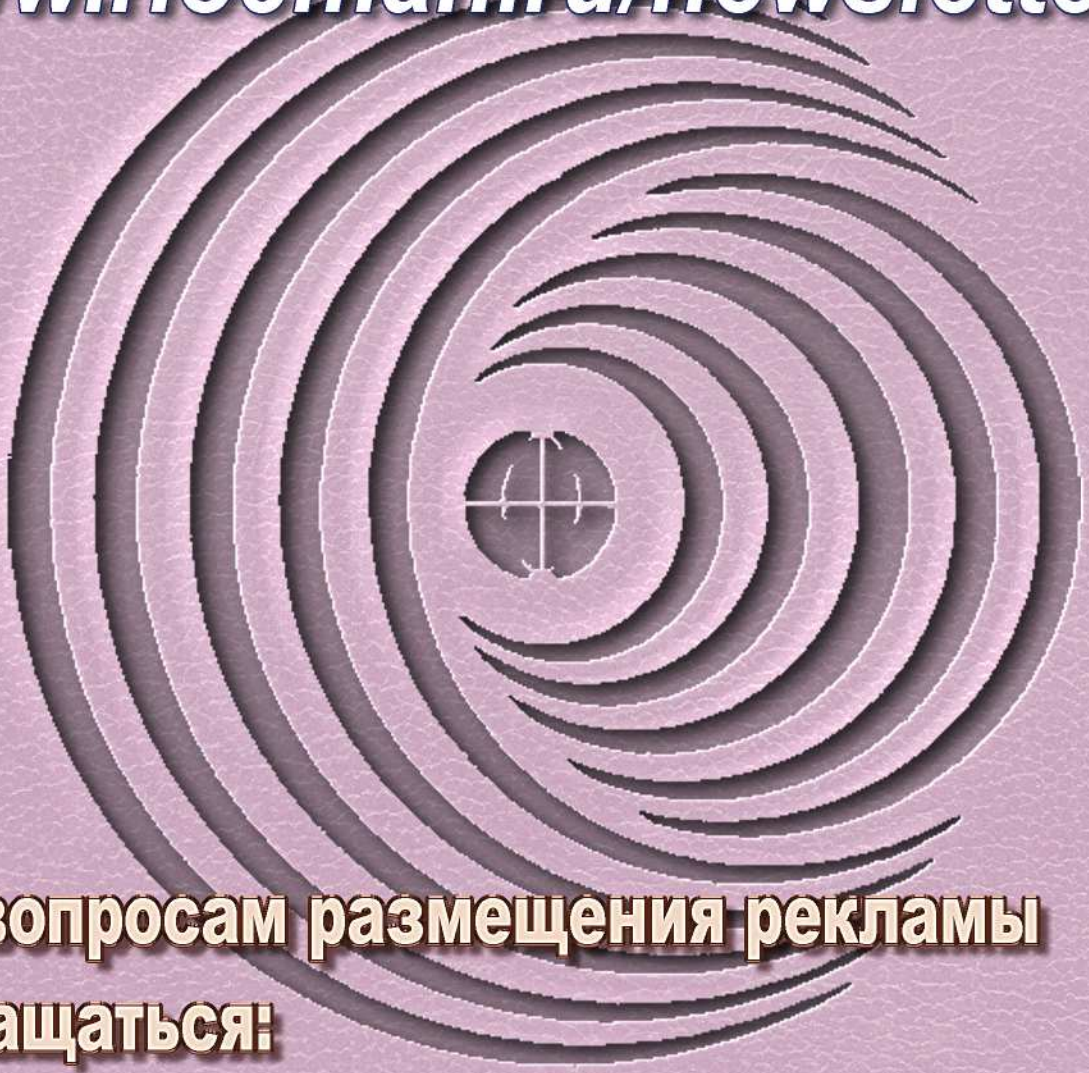
18 моделей по 5 штук в корпусе TO-92, всего 90 транзисторов **200 руб.**

Уведомление о новых
выпусках "Радиоежегодника"
на основной ленте новостей

www.rlocman.ru

и выпусках почтовой рассылки

www.rlocman.ru/newsletter



По вопросам размещения рекламы
обращаться:

rlocman@rlocman.ru

radioyearbook@gmail.com