USB-I²C Bridge

'Easyspeak I-square-C' using 1-Wire

Fons Janssen (The Netherlands), Maxim Benelux

Modern computers no longer come with parallel or serial ports fitted. Although these ports see declining use in practice, to designers, a portless PC is a nuisance. To solve at least a part of the problem, we present an interface for the I²C bus for connection to an USB connection on your PC. It works a treat when used in the LabVIEW environment.

Quick project specs

- Simple and compact design
- Compatible with Windows XP and Vista
- Compatible with LabVIEW 7 or newer (must have .net support)
- Works on USB-power
- All ICs can be sampled at Maxim

Sometimes you simply want your PC to communicate directly to hardware via an I²C interface. There are several options to do this, but not all of them offer the right features. For instance, there is a way to use the parallel port to emulate an I²C master (see [1]). A problem with this solution is that new PCs often lack a parallel port (a.k.a. 'LPT", 'Centronics' or 'printer'). Another problem is that contemporary Windows releases like XP and Vista do not allow easy access to the old serial or parallel ports, making them cumbersome to use if they are fitted in the first place.

Enter USB

A hopefully future-proof approach would be to use the USB port. However, this requires a complex driver and most probably a microcontroller, which requires the appropriate firmware. This is not something you can build in an afternoon. There are numerous commercial versions that don't come cheap.
The following solution can be built quickly and offers the possibility to use it in a variety of programming languages (LabVIEW, Visual Basic, etc.). This solution employs 1-Wire technology from Maxim Integrated Products (or 'Maxim' for short).

Hardware

The hardware setup is shown in **Figure 1**. The circuit connects to the PC via USB connector K1. The circuit supply voltage arriving via the USB is filtered by L1 and L2. It would probably work without these inductors, but I like to firmly rule out any potential problems. The USB data lines D+ and D– are connected to IC2; the DS2490S+, via R2 and R3. This IC contains a complete USB slave and takes care of the USB enumeration process. R1 pulls up the D+ line to indicate that IC2 is a full-speed (i.e. 12 Mbit/s) USB slave. IC2 also contains a 1-Wire master, so that any 1-Wire slave device can be addressed from the PC. The only 1-Wire slave in the circuit is IC3 (a DS2413 from Maxim), which is an I/O Extender offering two bidirectional open-drain ports (PIOA & PIOB). R4 and R5 are

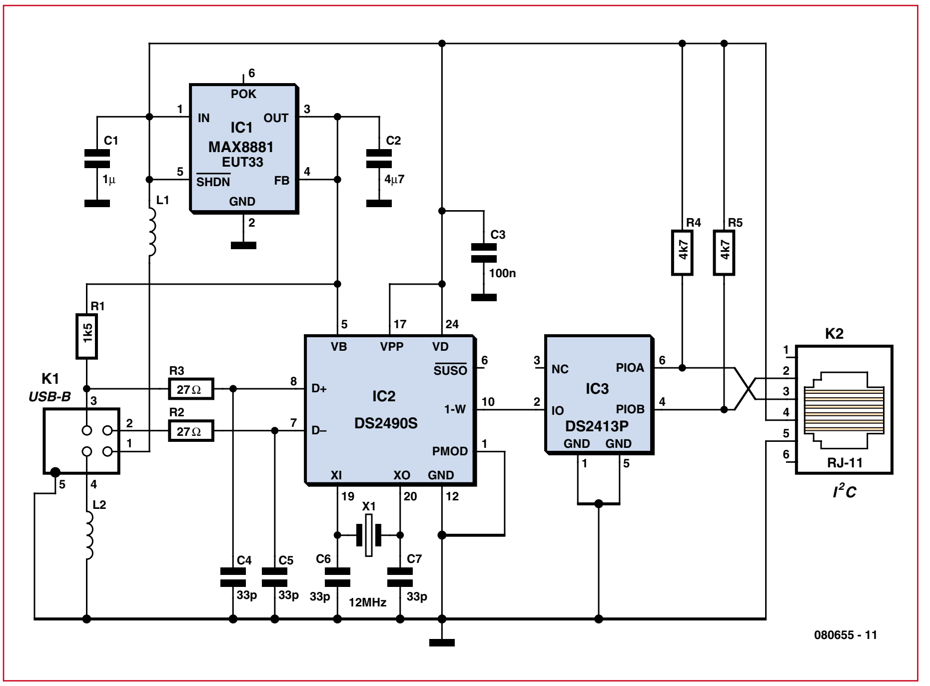
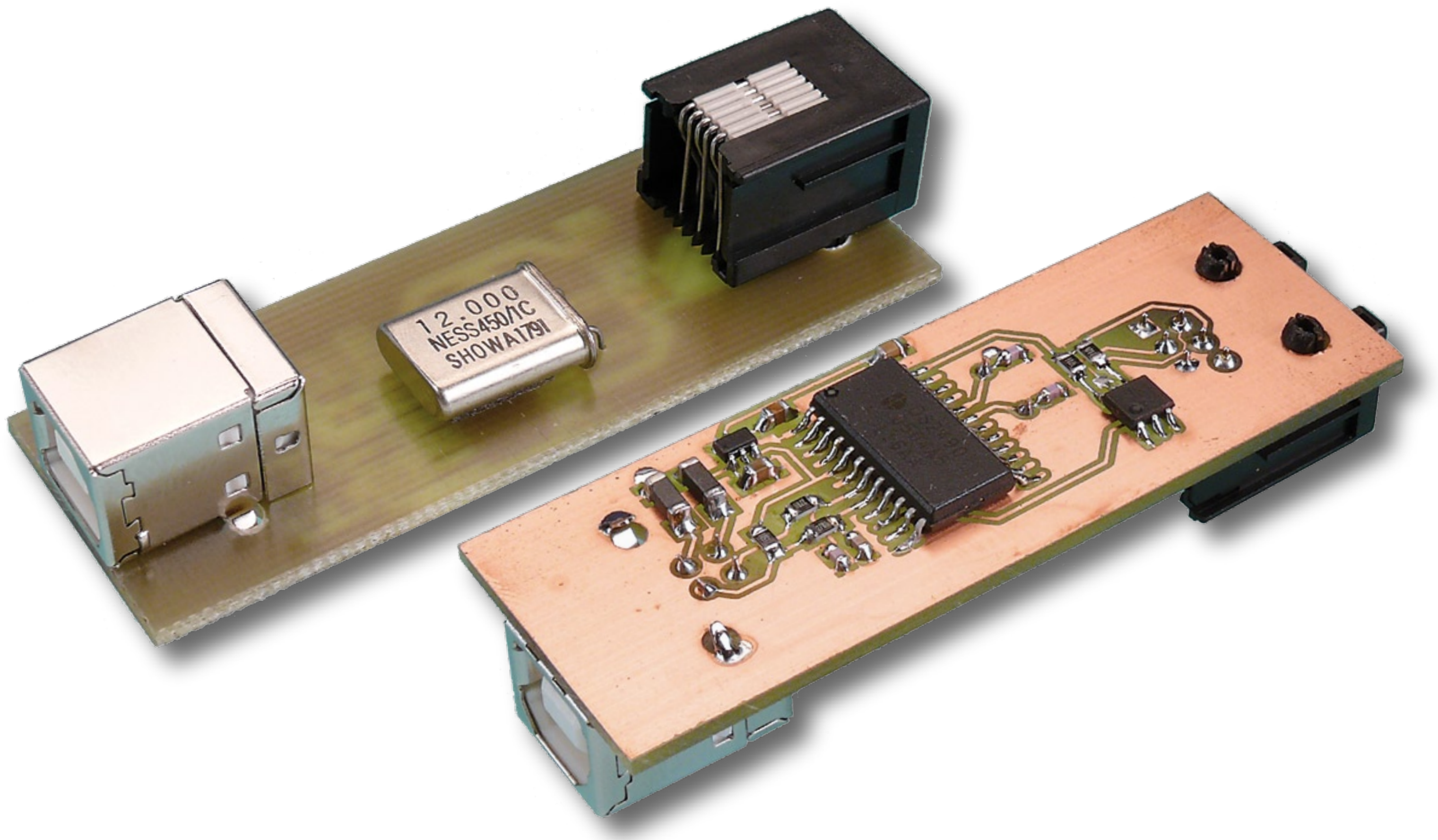


Figure 1. This circuit provides an I²C interface to a computer's USB port.



pull-up resistors for these two open-drain ports. This is exactly the right amount of hardware needed to create an I²C master. There is a PCB layout (see **Figure 2**) that can be downloaded free from the Elektor website [2] for home etching and drilling. Alternatively, readers can order a ready-made PCB from ThePCBShop.com [3].

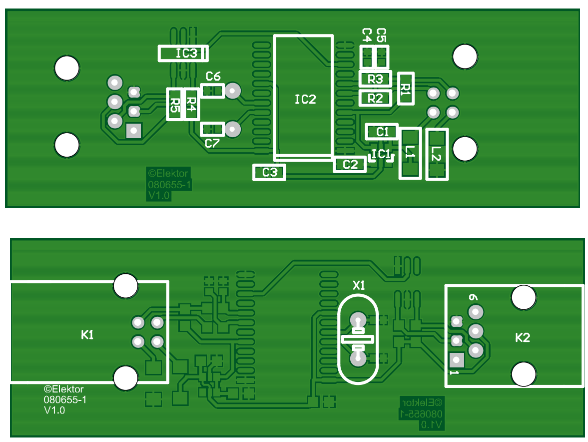


Figure 2. The component layout is very orderly. The PCB artwork can be downloaded free of charge from www.elektor.com.

Software

On microcontrollers that do not have a hardware I²C master, 'bit banging' is a proven method for implementing a software I²C master. With this very same technique, a software I²C master can be created on this platform.

The software needs to control the logic levels of the two pins to generate the I²C signals. For instance, to create a so-called start condition, SCL needs to be High, while SDA drops from High to Low. To create this condition, first both SCL (PIOA) and SDA (PIOB) need to be High, after which SDA is pulled Low. So, if the software has the ability to control PIOA and PIOB, the I²C master is a matter of software.

To change the PIO outputs, the part needs to be addressed first (ROM functions). In this case, there is only one 1-Wire slave present, so the skip ROM command ('CC'; for a full list of 1-Wire commands for various devices see [4]) can be used to skip the complex addressing procedure. Then the '5A' command is sent to allow writing to the PIO output latches (the command set for the DS2413 and its PIO function flow chart can be found in its datasheet [5]).

There is an 8-bit register of which the

COMPONENT LIST

Resistors
(all 0805 shape)
R1 = 1kΩ5
R2,R3 = 27Ω
R4,R5 = 4kΩ7

Capacitors
C1 = 1µF (0805)
C2 = 4µF7 (0805)
C3 = 100nF (0805)
C4-C7 = 33pF (0603)

Inductors
L1,L2 = BLM21PG221SN1J (Murata) (220Ω at 100 MHz)

Semiconductors
IC1 = MAX8881EUT33+ (Maxim Integrated Products)
IC2 = DS2490S+ (Maxim Integrated Products)
IC3 = DS2413P+ (Maxim Integrated Products)

Miscellaneous
K1 = USB connector, PCB mount, type B
K2 = RJ-11 connector, PCB mount
X1 = 12MHz quartz crystal
PCB, ref. 080655-1, from www.ThePCB-Shop.com

two LSBs are connected to PIOA and PIOB (the other six bits are 'don't care'). To change the logic state, the new data is sent twice: once normal and once inverted. The DS2413 confirms error free reception by returning 'AA' to the master and concludes by sending the new output state as well.

Now the whole sequence can start over again, or it can be aborted by a reset from the 1-Wire master.

Table 1. Generating an I²C start command

Master Mode	Data	Comments
TX	(Reset)	Reset the bus
RX	(Presence)	DS2413 is present
TX	0xCC	Skip ROM
TX	0x5A	Write PIO register
TX	0xFF	SCL=1 & SDA=1
TX	0x00	Inverted data
RX	0xAA	Data is received
RX	0x0F	Return output state
TX	0xFD	SCL=1 & SDA=0
TX	0x02	Inverted data
RX	0xAA	Data received
RX	0x2D	Return output state

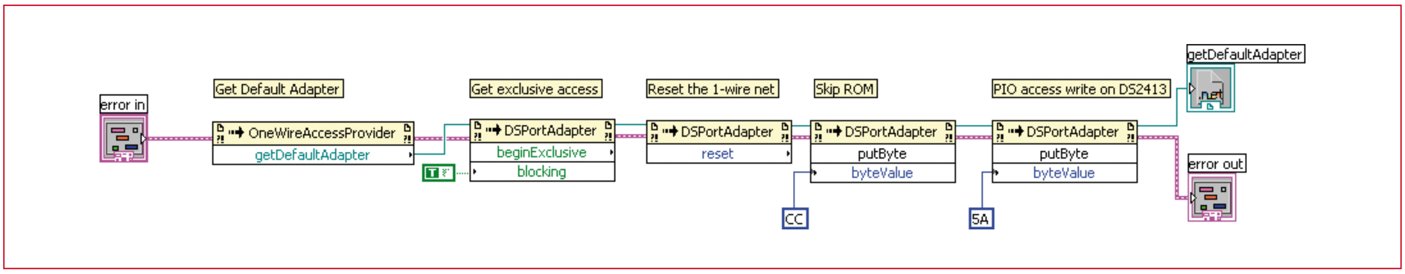


Figure 3. The Virtual Instrument called 'I²C Initialize' initialises the DS2413, after which it is ready to receive I/O data.

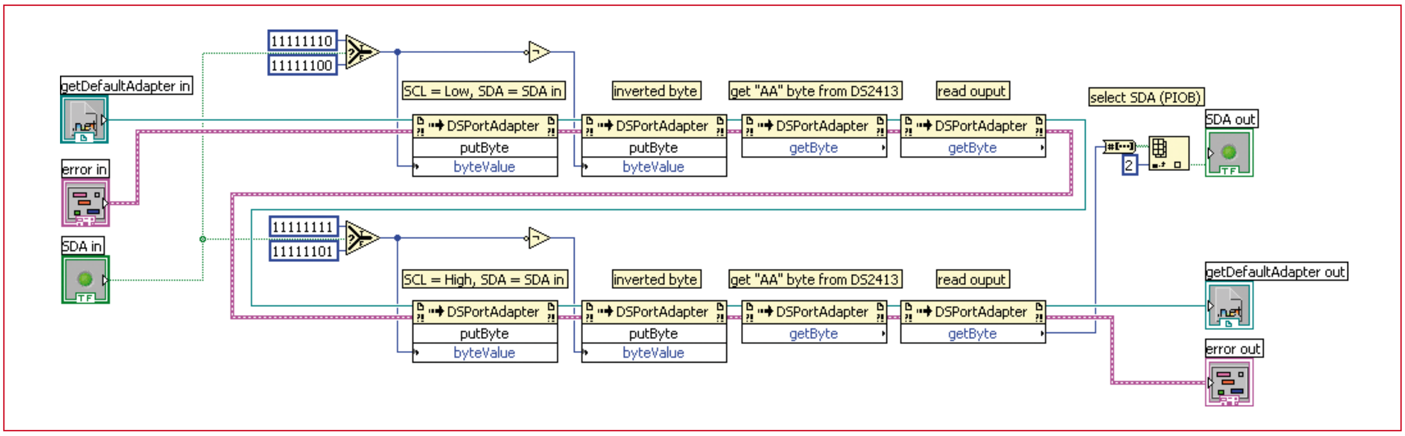


Figure 4. The VI 'I²C clock' toggles SCL from Low to High. SDA can be set or reset for 'I²C write' actions and read for 'I²C read' actions.

Thus, a total of four bytes is transferred for each output change (if the Skip ROM and 5A command are not taken into account).
To generate the I²C start command, the sequence shown in **Table 1** needs to be executed. For other events, similar sequences apply (stop, send byte, acknowledge, etc.).

.net

Maxim offers software drivers for the DS2490 (DS9490) and also a 1-Wire Software Development Kit (SDK) for Windows. This SDK offers support for Microsoft's .NET platform. As an example we will show how an I²C master can be implemented in LabVIEW, using its .NET support. This example only works if the 1-wire driver is installed with .NET support. Also the Microsoft .NET framework and Visual J# redistributable package need to be installed first (see [6] for details).

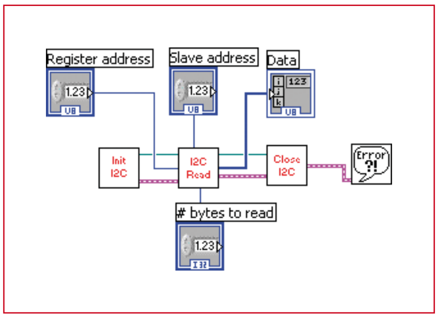


Figure 5. Example of how to read a register.

Virtual Instruments

The LabVIEW Virtual Instrument (VI) 'I²C Initialize' (**Figure 3** displays the block diagram) initializes the DS2490 and gives LabVIEW exclusive access to the 1-Wire net. This is to prevent other applications from accessing the 1-Wire net. After this, a 1-Wire reset is issued followed by a skip command (CC) and concluded by a PIO write command (5A).

Now the DS2413 is ready to accept data for the PIOs. The VI 'I²C clock' (**Figure 4**) toggles the SCL line from Low to High; SDA can be set High or Low and the VI returns the SDA state. A complete byte can be written by calling this function eight times in a row (once for every bit transmitted). If the slave is returning data, SDA is pulled High by the master. The slave can pull

SDA Low in case it wants to transmit a ‘0’ (this is okay, since PIOB is an open-drain terminal). Since the DS2413 automatically returns the updated output state, no special read action is required (see VIs ‘Send I^{2}C byte’ and ‘Get I^{2}C byte’ in the supplementary software download [2]).

The VIs ‘I^{2}C start’, ‘I^{2}C stop’, and ‘I^{2}C acknowledge’ — also in the supplementary download — use the same structure to generate the appropriate SDA and SCL signals. By combining these VIs, a complete I^{2}C read or write session can be programmed.

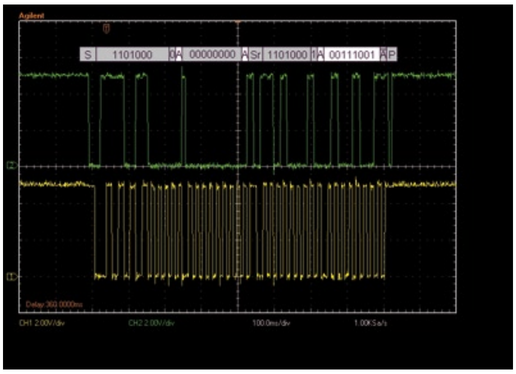


Figure 6. Read register 0 of the DS1337. S=start, A=acknowledge, Sr=repeated start, and P=Stop. Traffic from master to slave is shaded; from slave to master, white.

In **Figure 5** we see an example of a communication session, where Register 0 of a DS1337 (I^{2}C Real Time Clock) is read. The DS1337 answers with 0x39, which is the content of the register that counts the seconds. The register address is set to 0x00, the slave address to 0xD0, and the number of bytes to read is set to 1. With these settings, the scope image in **Figure 6** shows the signals generated by the circuit.

Because the overhead is quite big, the rate at which the signals are generated is severely limited (of the order of 20 bits/s). However, the solution is quite functional. The VIs that are used in the figures can be downloaded freely from Elektor’s website [2]. Samples of all the ICs mentioned in this article are available using Maxim’s sampling service for design engineers and students. It should be noted that the DS2490 is not recommended for new designs, but samples will still be available for a while.

(080655-I)

Internet Links

[1] www.maxim-ic.com/appnotes.cfm/an_pk/3230

[2] www.elektor.com/080655

[3] www.thepcbshop.com

[4] http://owfs.sourceforge.net/family.html

[5] http://datasheets.maxim-ic.com/en/ds/DS2413.pdf

[6] http://pdfserv.maxim-ic.com/en/an/AN155.pdf, page 16-17.

NEWS & NEW PRODUCTS **INFO & MARKET**

Multiprocessing, magnetism and the milky way

3L recently announced that a new broad-band spectroscopic data acquisition system has been built and deployed for the 26-m radiotelescope of the Dominion Radio Astrophysical Observatory (DRAO) in Penticton, Canada. Designed using 3L’s Diamond multiprocessor tool-suite and based on multiprocessing hardware from Sundance, the high sensitivity system developed at DRAO has an instantaneous bandwidth of 500 MHz and a spectral resolution of 2048 channels. The phase relation of the two input signals is measured in real-time and using 3L Diamond, the instrument design was completed in only 18 months, allowing scientists to deploy the data acquisition system earlier than would have otherwise been possible.

Radio astronomers in Penticton are using the instrument to investigate the magnetic field in the interstellar medium of our Galaxy. The large bandwidth along with spectral capabilities enabled by multiprocessing is allowing them to measure the direction and field strength of the magnetic field of the Milky Way. Based on the NRC system, DRAO astronomers are now leading a project to utilize this method with other radiotelescopes around the globe including the 100-m Effelsberg telescope in Germany and the 64-m telescope at the Parkes Observatory in Australia.

The NRC design was completed using 3L’s Diamond multiprocessor tool-suite with the hardware consisting of a Sundance SMT310Q quad site carrier card populated with four daughter modules hosted on TIM (Texas Instruments Module) sites. A C series TI DSP mounted on the SMT395, two large Xilinx Virtex FPGAs on the SMT398 and a Dual Channel 1-GHz, 8-bit ADC on the SMT391. The ADC samples two input channels at 1 GHz which are then sent to the FPGAs for Fast Fourier Transform.

www.3L.com