THE No 1 UK MAGAZINE FOR ELECTRONICS TECHNOLOGY & COMPUTER PROJECTS

EPE EVERYDAY PRACTICAL ELECTRONICS

www.epemag.com

JACOB'S LADDER

Spectacular high voltage sparks

Uses High-Energy Ignition module

Safe battery operation

IGBT driven

BETA RE-FLOW OVEN CONTROLLER REVIEW

AFFORDABLE, BUT EFFECTIVE ROUTE TO SMD SOLDERING

DELUXE GPS TIMEBASE FOR FREQUENCY COUNTERS

SUPER-SMART VERSION OF THE TIMEBASE

WIN A MICROCHIP PICDEM LAB DEVELOPMENT KIT

Teach-In 2014 Raspberry Pi – Part 7

EPE's comprehensive guide to Raspberry Pi

PLUS: NET WORK, CIRCUIT SURGERY, INTERFACE TECHNO TALK AND PIC N' MIX

APRIL 2014 £4.40

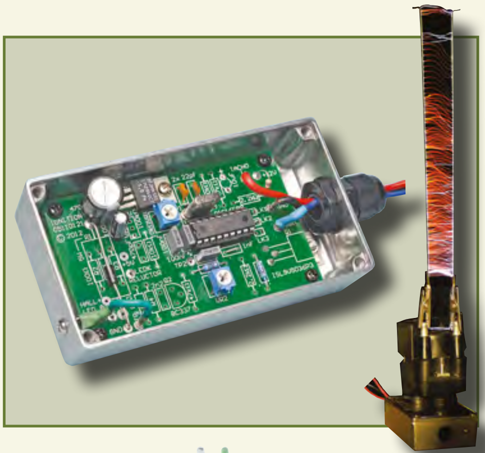
9 770262 361188 04

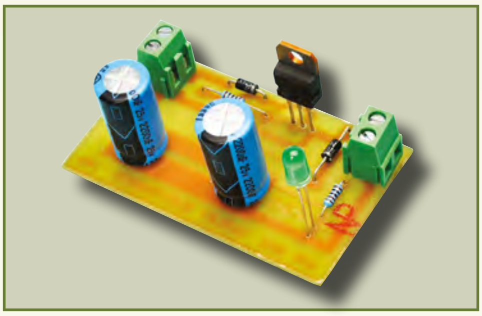
ISSN 0262 3617
• PROJECTS • THEORY •
• NEWS • COMMENT •
• POPULAR FEATURES •
VOL. 43. No 04 April 2014

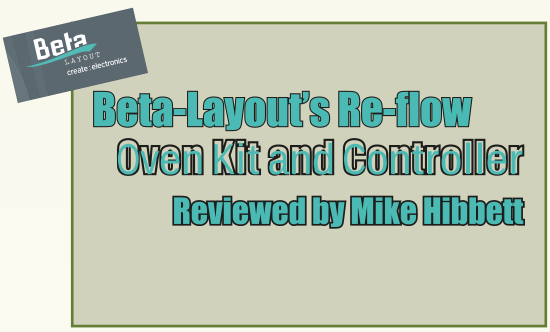
EPE EVERYDAY PRACTICAL ELECTRONICS

INCORPORATING ELECTRONICS TODAY INTERNATIONAL

www.epemag.com







Our May 2014 issue will be published on Thursday 03 April 2014, see page 72 for details.

Projects and Circuits

JACOB'S LADDER 10
by Leo Simpson
Snap, crackle and ZAP! Build this spectacular high-voltage spark machine!

DELUXE GPS 1PPS TIMEBASE FOR FREQUENCY COUNTERS 16
by Jim Rowe
Even better than last month's timebase – this one includes the NMEA 0183 data stream from GPS satellites

CAPACITOR DISCHARGE UNIT FOR TWIN-COIL POINTS MOTORS 26
by Jeff Monegal
A must-have, easy-to-build points driver for model railway enthusiasts

Series and Features

BETA-LAYOUT'S RE-FLOW OVEN KIT AND CONTROLLER 28
by Mike Hibbett
Expert review of affordable re-flow soldering equipment

TECHNO TALK by Mark Nelson 33
Blessed broadband, corrupt copper

TEACH-IN 2014 by Mike and Richard Tooley 36
Part 7: Arithmetic bases, port expansion and setting up a Pi web server

PIC N' MIX by Mike Hibbett 48
Goodbye MPLAB, Hello Kickstarter

INTERFACE by Robert Penfold 53
Using the Raspberry Pi camera

MAX'S COOL BEANS by Max The Magnificent 56
Welcome to the Pleasure Dome!... His 'n' hers...
Motorised pots... Tubes and LEDs

CIRCUIT SURGERY by Ian Bell 58
MOSFET basics – Part 2

NET WORK by Alan Winstanley 66
What's cooking?
What IFTTT?

Regulars and Services

SUBSCRIBE TO *EPE* and save money 5

EDITORIAL 7
Giant steps

NEWS – Barry Fox highlights technology's leading edge 8
Plus everyday news from the world of electronics

***EPE* BACK ISSUES** – Did you miss these? 34

MICROCHIP READER OFFER 35
EPE Exclusive – Win a PICDEM Lab Development Kit

TEACH-IN 5 51

***EPE* PIC PROJECTS VOLUME 2** 52

CD-ROMS FOR ELECTRONICS 62
A wide range of CD-ROMs for hobbyists, students and engineers

DIRECT BOOK SERVICE 68
A wide range of technical books available by mail order, plus more CD-ROMs

***EPE* PCB SERVICE** 70
PCBs for *EPE* projects

ADVERTISERS INDEX 71

NEXT MONTH! – Highlights of next month's *EPE* 72

Readers' Services • Editorial and Advertisement Departments 7

EVERYDAY PRACTICAL ELECTRONICS

UK readers you can SAVE 81p on every issue of *EPE*

How would you like to pay £3.59 instead of £4.40 for your copy of *EPE*?

Well you can – just take out a one year subscription and save 81p an issue, or £9.80 over the year. You can even save £1.08 an issue if you subscribe for two years – a total saving of £26.10.

Overseas rates also represent exceptional value.

You also:

- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Ensure your copy, even if the newsagents sell out**

Order by phone or fax with a credit card or by post with a cheque or postal order, or buy online from www.epemag.com (go to the Online Shop).

EVERYDAY PRACTICAL ELECTRONICS

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in the UK: 6 months £23.50, 12 months £43.00, two years £79.50; Europe Airmail: 6 months £27.00, 12 months £50.00, 24 months £95.00; Rest Of The World Airmail: 6 months £37.00, 12 months £70.00, 24 months £135.00.

Cheques or bank drafts (in **£ sterling only**) payable to *Everyday Practical Electronics* and sent to EPE Subs. Dept., Wimborne Publishing Ltd., 113 Lynwood Drive, Merley, Wimborne, Dorset, BH21 1UU. Tel: 01202 880299. Fax: 01202 843233. **Email:** subs@wimborne.co.uk. Also via the **Web** at: **www.epemag.com**.

Subscriptions start with the next available issue. We accept MasterCard, Maestro or Visa. (For past issues see the Back Issues page.)

ONLINE SUBSCRIPTIONS

Online subscriptions, for reading the magazine via the Internet, £19.99 for one year, visit **www.epemag.com** for more details.

SUBSCRIPTION ORDER FORM

☐ **6 Months: UK £23.50, Europe £27.00 (Airmail), Rest Of The World £37.00 (Airmail)**

☐ **1 Year: UK £43.00, Europe £50.00 (Airmail), Rest Of The World £70.00 (Airmail)**

☐ **2 Years: UK £79.50, Europe £95.00 (Airmail), Rest Of The World £135.00 (Airmail)**

To: Everyday Practical Electronics, Wimborne Publishing Ltd., 113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel: 01202 880299 Fax: 01202 843233
E-mail: subs@epemag.wimborne.co.uk

I enclose payment of £ .... (cheque/PO in £ sterling only), payable to *Everyday Practical Electronics*

☐ Please charge my Visa/Mastercard/Maestro

My card number is: ....
Please print clearly, and check that you have the number correct

Signature ....

Card Security Code .... Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....Maestro Issue No. ....

Name ....

Address ....

Post code .... Tel. ....

Subscriptions can only start with the next available issue.

MasterCard Maestro VISA

VOL. 43 No. 04 APRIL 2014

Editorial Offices:
EVERYDAY PRACTICAL ELECTRONICS EDITORIAL Wimborne Publishing Ltd., 113 Lynwood Drive, Merley, Wimborne, Dorset, BH21 1UU
Phone: (01202) 880299. **Fax:** (01202) 843233.
Email: fay.kearn@wimborne.co.uk
Website: www.epemag.com
See notes on **Readers' Technical Enquiries** below – we regret technical enquiries cannot be answered over the telephone.

Advertisement Offices:
Everyday Practical Electronics Advertisements
113 Lynwood Drive, Merley, Wimborne, Dorset, BH21 1UU
Phone: 01202 880299 **Fax:** 01202 843233
Email: stewart.kearn@wimborne.co.uk

Editor:	MATT PULZER
Subscriptions:	MARILYN GOLDBERG
General Manager:	FAY KEARN
Graphic Design:	RYAN HAWKINS
Editorial/Admin:	01202 880299
Advertising and Business Manager:	STEWART KEARN 01202 880299
On-line Editor:	ALAN WINSTANLEY
Publisher:	MIKE KENWARD

READERS' TECHNICAL ENQUIRIES
Email: fay.kearn@wimborne.co.uk
We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We regret that we cannot provide data or answer queries on articles or projects that are more than five years' old. Letters requiring a personal reply must be accompanied by a stamped self-addressed envelope or a self-addressed envelope and international reply coupons. We are not able to answer technical queries on the phone.

PROJECTS AND CIRCUITS
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in EPE employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD adaptor.

COMPONENT SUPPLIES
We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers.

We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

ADVERTISEMENTS
Although the proprietors and staff of EVERYDAY PRACTICAL ELECTRONICS take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts.

The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

TRANSMITTERS/BUGS/TELEPHONE EQUIPMENT
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

EVERYDAY PRACTICAL ELECTRONICS

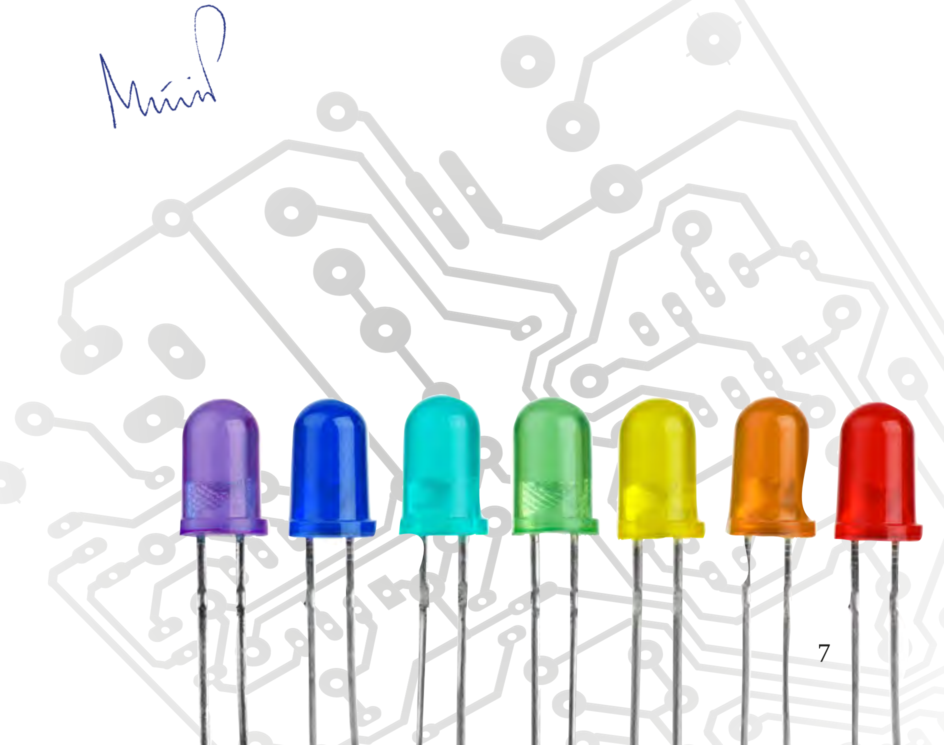
Giant steps

I was recently sent a press release by the Institute of Chemical Engineers, listing the results of a poll that asked their members what they considered to be the ten most important chemically engineered inventions and solutions of the modern era. Safe drinking water, petrol and antibiotics were the top three. Electricity generation came in at number four. Hang on, I thought – doesn't that belong to another set of engineers? To be fair to the IChemE (and their members) the complete fourth listing was 'Electricity generation from fossil fuels'. Even so, I thought it a little 'cheeky' and was surprised that chemists had rated it above plastics and fertilizer!

Now, I certainly don't want to get into a turf war with chemists – whatever the professional boundaries between chemical and electrical/electronic engineering, they are both professions that are vital to every aspect of our modern life. In fact, electronics owes a huge amount to the contribution of chemistry in the development of techniques to make all types of electronic components. Instead, it got me wondering what my top ten for the electronic and electrical industries should be.

I came up with the following technologies: 1) Batteries – the first reliable source of electric power. 2) Transformers – the cornerstone of the expansion of mass electric power distribution. 3) Light bulbs – just try living without them. 4) Induction motors – simple, rugged and reliable; Tesla's greatest invention has been keeping the wheels of industry moving for over a century. 5) Thermionic devices – without the diode and triode we'd have never dreamt up their solid-state equivalents. 6) Junction-based electronic devices – diodes, transistors and FETs are reliable, scaleable and irreplaceable. 7) Integration – from op amps to microprocessors, one simple but important idea has given us the ability to make cheaply multi-billion-component devices. 8) Computers – *the* disruptive technology of the last 70 years. 9) The world-wide web – for better, and sometimes worse, the web's spread of knowledge and data is unprecedented in history. 10) Qubit – or 'quantum bit', if engineers can get quantum computing to work, it will make a modern computer look like an abacus.

These are just my top ten, and they reflect my biases, interests and probably lack of knowledge in some areas. You can only fit so much into a list of 10; I haven't even mentioned optoelectroncs, radio and a host of other technologies. Let me know where you agree, disagree and what your list would look like.



NEWS

A roundup of the latest Everyday News from the world of electronics

Freeview fights its corner – report by Barry Fox

Freeview free-to-watch digital terrestrial TV – which grew out of the world's first full-scale but failed pay DTTV services ONDigital and ITV Digital – is one of the great British success stories. Now Freeview is under threat from mobile operators who would love to buy its 700MHz frequencies.

So Digital UK, the body funded by the BBC, ITV, Channel 4 and Arqiva, which organised the digital switch-over to clear 800MHz spectrum for 4G mobile broadband, is fighting back with a report that shows Freeview is worth £80 billion to the UK economy. This is greater value than mobile broadband – and greater value than the UK government has so far thought.

The DUK report, by media and telecoms consultancy Communications Chambers, bases the £80 billion figure on 15,000 jobs in broadcasting and independent production; a report by Analysis Mason for the UK government DCMS in November 2012 had estimated the benefit from DTTV at only £63.6 billion.

Value of DTTV

The DUK report estimates that the average value per MHz of spectrum for DTTV is far higher than that for mobile data; £0.19 billion for mobile compared to £0.47 billion for DTTV.

This write-down for mobile broadband value is based on the way Wi-Fi is becoming an increasingly viable substitute for mobile data. Three-quarters of tablets are Wi-Fi-only, with no mobile connectivity. Wi-Fi is widely available, often for free, for example from BT's hot spots. The result is that mobile networks only carry 18% of mobile device traffic.

DTTV is able to reach over 98% of the UK population, so is key to public service broadcasting (PSB) TV such as BBC commercial TV. Video is a key driver of mobile data growth, which has a much lower value per MB or MHz than web or email. Voice creates most of mobile's value yet uses little network capacity – Ericsson estimates 5% in Western Europe.

With a clear nod to Rupert Murdoch and Sky, the report notes that DTTV with PSB is less likely than any one platform owner to exercise undue influence over public opinion or the political agenda.

Freeview/Freesat rivalry

Showing the rivalry that exists between Freeview free-to-air DTTV and Freesat free-to-air satellite, the report argues that DTTV is 'a critical component of TV platform competition and is much better placed than Freesat to provide robust competition to pay operators, as well as providing critical competition within that segment of the market that will always seek free-to-Air TV'.

If free-to-air DTTV no longer existed, the report warns, it is highly likely that the established pay operators would be the prime beneficiary: 'Sky in particular would have both the incentive and the ability to attract former DTTV viewers to its satellite platform' and 'would have significant advantage over Freesat in doing so… with a marketing budget of £1.1 billion compared to Freesat's total operating expenditure of £12 million'.

What's more, the report sums up, Mobile already has a far greater allocation of spectrum than DTTV, occupying 560MHz vs 256MHz

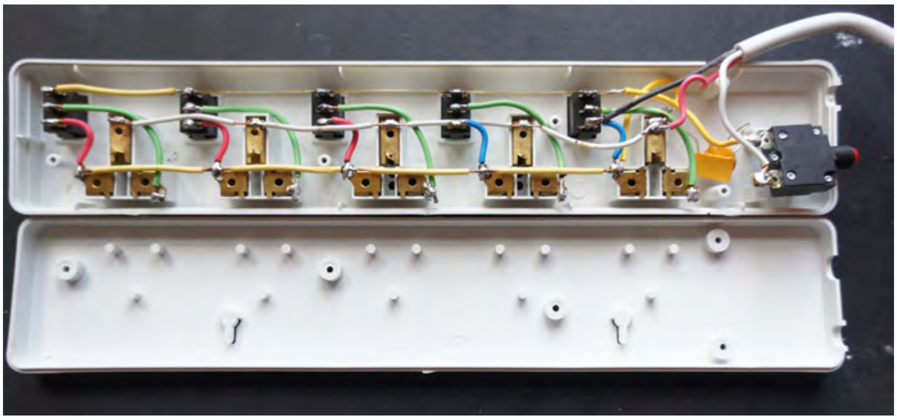
'Don't be greedy' is the clear message to the mobile operators and Treasury which may see the prospect of easy cash from a 700MHz spectrum sale.

A cautionary tale

A foreign multi-gang mains socket, came with a moulded-on 250V flat pin and *round* earth plug. The wires into the moulded plug were red, white and black.

So that's red for live, black for neutral and white for earth, right? Wrong. Opening up the socket revealed the soldered connections inside to be black for live, white for neutral and red for earth.

Moral for DIYers: *Never assume anything.*



Cheap, imported multi-gang mains sockets – not worth the risk

UK infrastructure at risk of cyber attack

Data available from mainstream online media, such as blogs, social networking websites, and specialist online publications, could be used to mount a cyber-attack on UK critical national infrastructure, according to an investigative report.

Key information regarding vulnerabilities in company systems is now openly available from a range of sources on the Internet, according to 'Using Open Source Intelligence to Improve ICS & SCADA Security', a report carried out by design and engineering consultancy Atkins.

The research was published at the Institution of Engineering and Technology's (IET) 'Cyber Security for Industrial Control Systems' seminar. It discovered that many industrial sector websites and academic papers also provide some information which identifies staff and their social media information used to corroborate control systems data.

Hugh Boyes from the IET said: 'The UK has been proclaimed as the most Internet-based major economy. While this provides a basis for industry to expand and grow, it is essential that any connections between the Internet and industrial control systems are adequately protected.'

Proton flow battery may challenge lithium

Researchers at RMIT University in Australia have developed a concept battery based on storing protons produced by splitting water.

The proton flow battery eliminates the need for the production, storage and recovery of hydrogen, which limits the efficiency of conventional hydrogen-based electrical energy storage systems.

Lead researcher Associate Professor John Andrews said the novel concept combined the best aspects of hydrogen fuel cells and battery-based electrical power.

'As only an inflow of water is needed in charge mode – and air in discharge mode – we have called our new system the "proton flow battery". Proton-powered batteries have the potential to be a much more economical device than using lithium ions, which have to be produced from relatively scarce sources.

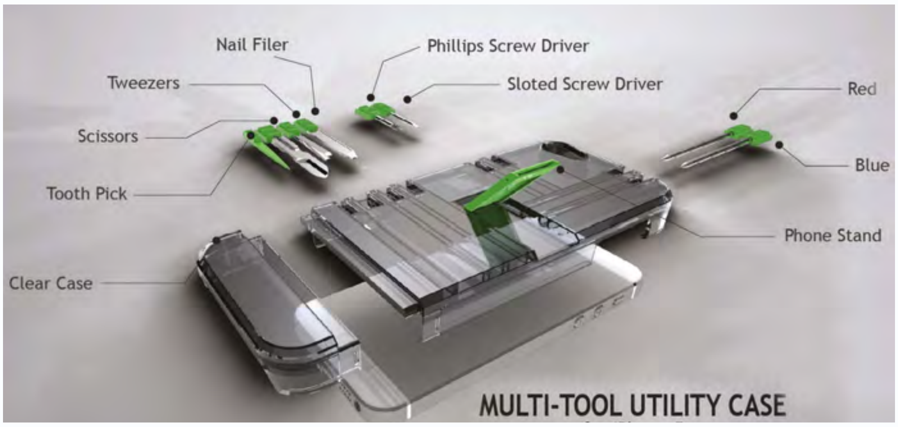
The concept integrates a metal hydride storage electrode into a reversible proton exchange membrane (PEM) fuel cell. During charging, protons produced from splitting water are directly combined with electrons and metal particles in one electrode of a fuel cell, forming a solid-state metal hydride as the energy storage. To resupply electricity, this process is reversed. The research found that, in principle, the energy efficiency of the proton flow battery could be as high as that of a lithium ion battery, while storing more energy per unit mass and volume.

Multi-tool utility iPhone case

Well, someone had to make it! – in this case that someone is 'IN1'. Combining the world-beating iPhone with something like a Swiss Army knife – although it isn't actually from that iconic knife manufacturer, this case is clearly in the spirit of their enduring classics. It has a variety of tools built in, including:

- 2 Precision screwdrivers
- 2 Ballpoint Pens (red and blue)
- A nail file
- Tweezers and scissors
- A phone kick stand

The tool collection is designed to help with everyday tasks. So, while the IN1 case won't help you remove and rebuild a gearbox, or slay, skin and cook a wildebeest, it will help with day-to-day tasks such as writing notes and cutting open packages. Available in white, black and clear from Amazon for £29.99.



XBee from Parallax

From Parallax, comes a pair of 'XBee Wi-Fi' embedded RF modules which provide simple serial-to-IEEE 802.11b/g/n connectivity. By bridging the low-power/low-cost requirements of wireless device networking with the proven infrastructure of 802.11, the XBee Wi-Fi creates new wireless opportunities for energy management, process and automation, wireless sensor networks, intelligent asset management and more. Key features: flexible SPI and UART interfaces provide flexible connection options, support for 802.11 b, g and n standards, 802.11n provides up to 72Mbps data. Available with or without antenna. Priced at $36.99, look for part numbers 32412/3 at: **www.Parallax.com**

iC0nstruX.com launches

Just launched, **iC0nstruX.com** is a website targeted at hackers and makers – people who 'build stuff'. The site offers lots of embedded systems, gizmos and gadgets, as well as complete kits, toys, and beginner products. Their video provides a handy overview: **www.youtube.com/watch?v=hYZ2QWfkQ7c**

Wearable UV exposure monitor

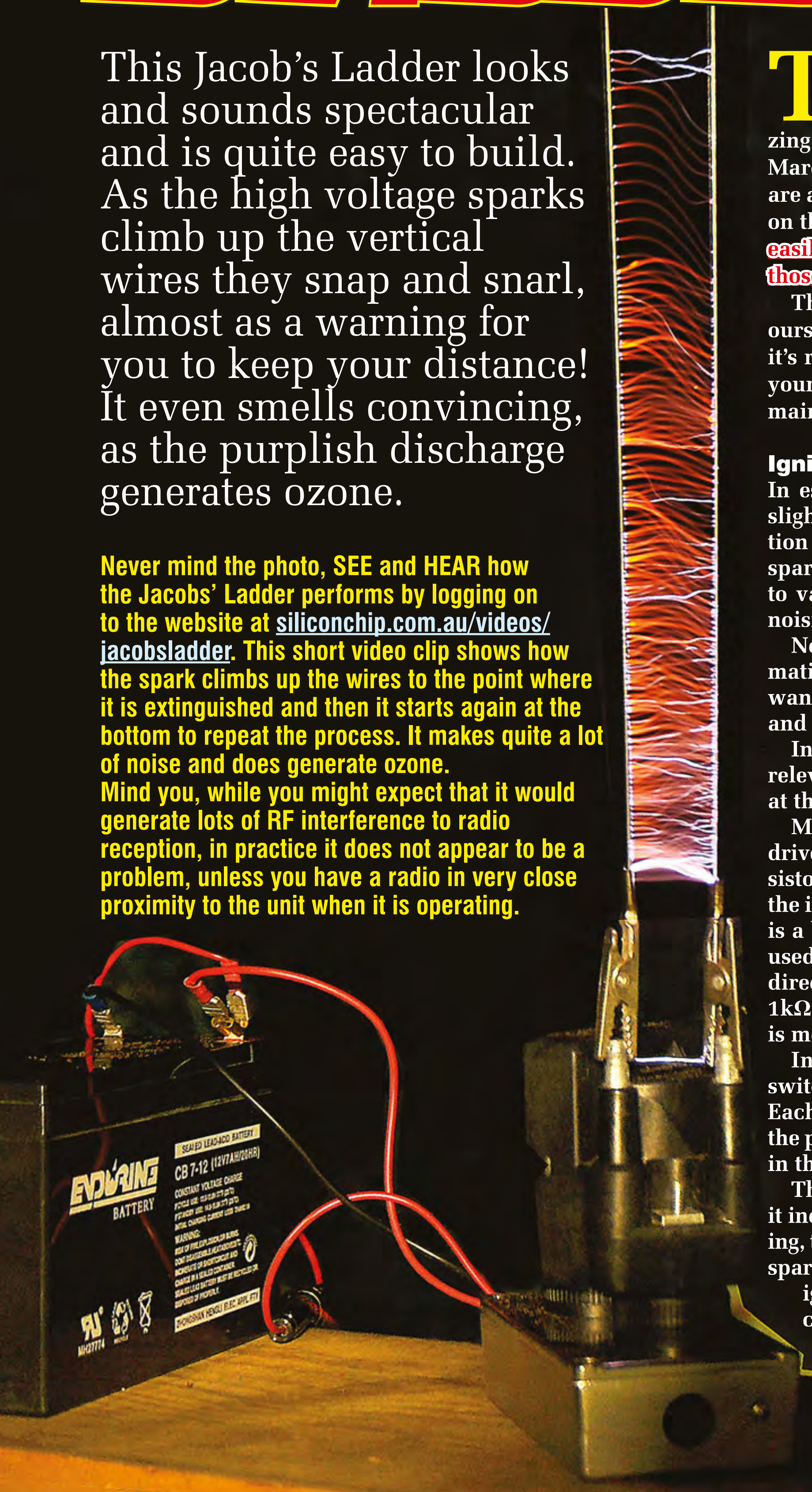
One of the hottest areas of electronics at the moment is 'wearable' electronics, particularly components and systems that monitor the wearer's health. Silicon Labs has launched a series of single-chip (Si1132/4x) optical sensors that can track UV exposure, ambient light and other biometrics to add health-monitoring functions to smartphones and wearable computing products.

New! Improved! More Zap for your bucks . . . Build this

JACOB'S LA

This Jacob's Ladder looks and sounds spectacular and is quite easy to build. As the high voltage sparks climb up the vertical wires they snap and snarl, almost as a warning for you to keep your distance! It even smells convincing, as the purplish discharge generates ozone.

Never mind the photo, SEE and HEAR how the Jacobs' Ladder performs by logging on to the website at siliconchip.com.au/videos/jacobsladder. This short video clip shows how the spark climbs up the wires to the point where it is extinguished and then it starts again at the bottom to repeat the process. It makes quite a lot of noise and does generate ozone.
Mind you, while you might expect that it would generate lots of RF interference to radio reception, in practice it does not appear to be a problem, unless you have a radio in very close proximity to the unit when it is operating.



This project is based on our design for a *High-Energy Ignition* module, which makes an ideal Jacob's Ladder driver – providing plenty of zing and zap for a stunning display (See Feruary and March 2014 *EPE*. (By the way, we are aware that there are a number of mains-power Jacob's Ladder circuits on the Internet. **These are very dangerous and could easily be lethal. Don't even consider building one of those – build ours!**

That's not to say you won't get a helluva belt off ours if you're silly enough to touch the bitey bits when it's running. But at least you'll be able to learn from your mistake – you may not get that chance with a mains-powered type.

Ignition module variant

In essence, the *Jacob's Ladder* presented here is a slight variation on designs for a self-contained ignition coil tester. Its frequency can be varied up to 75 sparks per second and the 'dwell' setting can be used to vary the timing to obtain the best sparks, ie, the noisiest and most nasty!

Now we are are not going to reproduce all the information on the *High Energy Ignition* module – if you want to read that you should refer to the February and March 2014 issues.

Instead, we will give all the information which is relevant to this particular variant. So let's have a look at the circuit of Fig.1.

Microcontroller IC1 is the heart of the circuit. It drives the gate of the IGBT (insulated gate bipolar transistor), Q1. These IGBTs are used by the squillion in the ignition system of modern cars. This type of IGBT is a big improvement on the high voltage transistors used in older ignition systems and it can be driven directly from the output of the microcontroller via a 1kΩ resistor from pin 9 (RB3). As a result, the circuit is more efficient and very little power is dissipated.

In operation, IC1's RB3 output is alternately switched high (to +5V) and low to turn Q1 on and off. Each time Q1 is turned on, the current builds up in the primary winding of the coil and this stores energy in the resulting magnetic field.

This magnetic field collapses when Q1 turns off and it induces a very large voltage in the secondary winding, to fire the spark plug, or in our case, to cause a big spark to jump across the high voltage terminals of the ignition coil.Incidentally, in the past, most ignition coils have been auto-transformers, meaning that the primary and secondary windings are connected

By LEO SIMPSON

together at one end. However, many modern ignition coils are true transformers, with completely separate primary and secondary windings.

The particular ignition coil we are using for the *Jacob's Ladder* is from a VS series Holden Commodore. These can usually be purchased from a wrecker or via ebay (which is where we got ours). We paid $27.50 including postage. **In the UK, eBay is also the best bet, but the car model that you need to search for is along the lines of: 'Camaro Firebird V6 Ignition Coil' – expect to pay around £30. It does need to look similar to the photo opposite and the penultimate page.** Apart from being a readily available high energy ignition coil, this unit has a further advantage in that it has two high voltage terminals and these normally drive two spark plugs in series when used in the Commodore V6 engine.

In our case, the two high-voltage terminals make it very suitable for a *Jacob's Ladder*. Just connect a stiff wire to each terminal and it's done. Now back to the circuit description.

In operation, IC1 monitors two separate voltages, at pin 1 and 18.

What is a Jacob's Ladder?

Jacob's Ladder has its origins in three major religions – Judaism, Christianity and Islam.

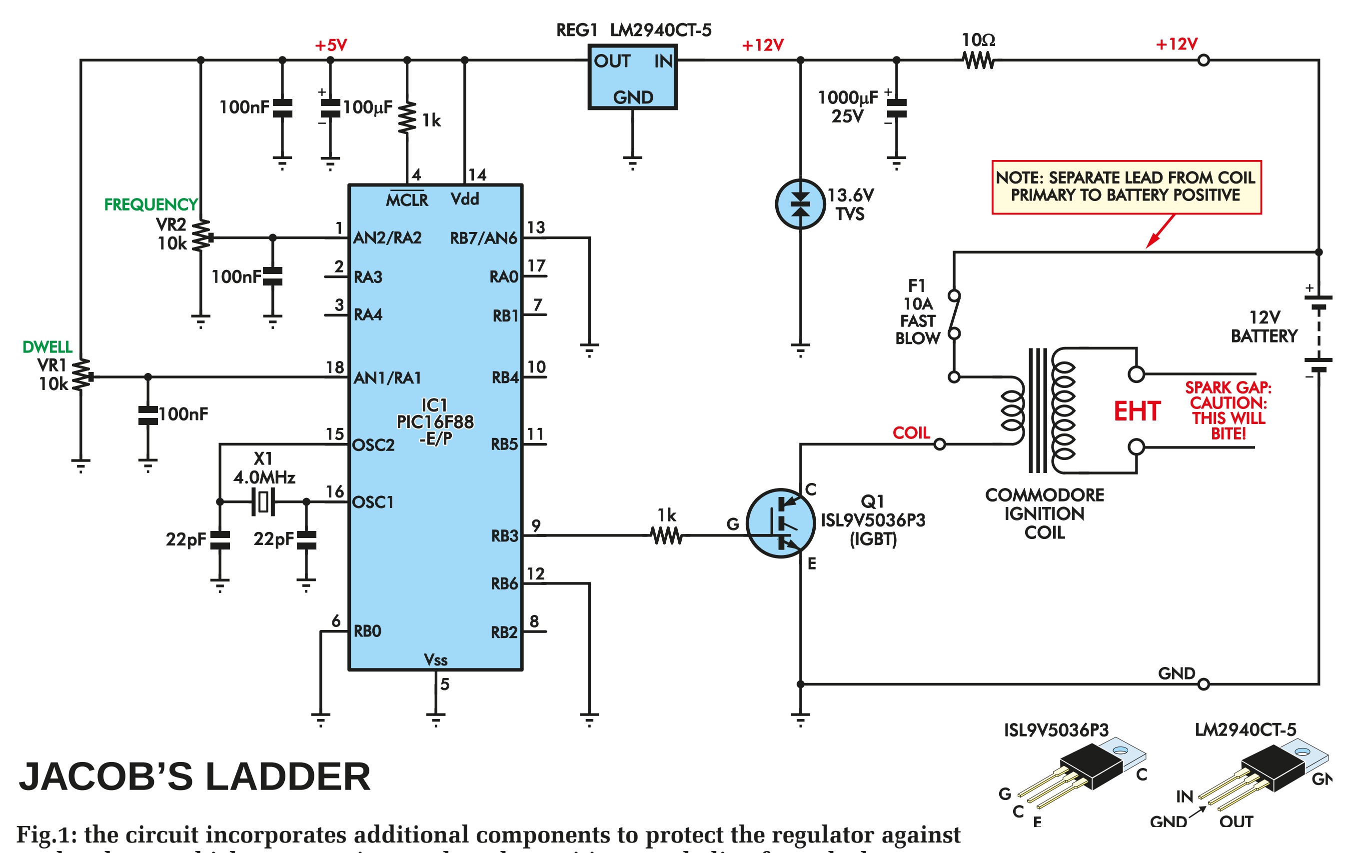
Jacob, the son of Abraham, dreamed about a 'ladder' between earth and heaven with angels climbing up and down. Some references have this ladder made from flames and sparks – hence the electronic version doing the same thing. OK, no flames – but plenty of mean-sounding sparks!

Physically, as our photos show, it has two parallel (or near-parallel) metal rods about 300mm long and about 30mm apart, which have such a high voltage between them that sparks jump from one to the other. As the spark is hot, the surrounding air is heated. Hot air rises, so the column of rising air pushes the sparks upward so that they appear to form the 'rungs' of a ladder.

The dwell period and spark rate are set by trimpots VR1 and VR2, each connected across the 5V supply. VR1 (dwell) is monitored by input AN1 (pin 18), while VR2 (frequency) is monitored by input AN2 (pin 1).

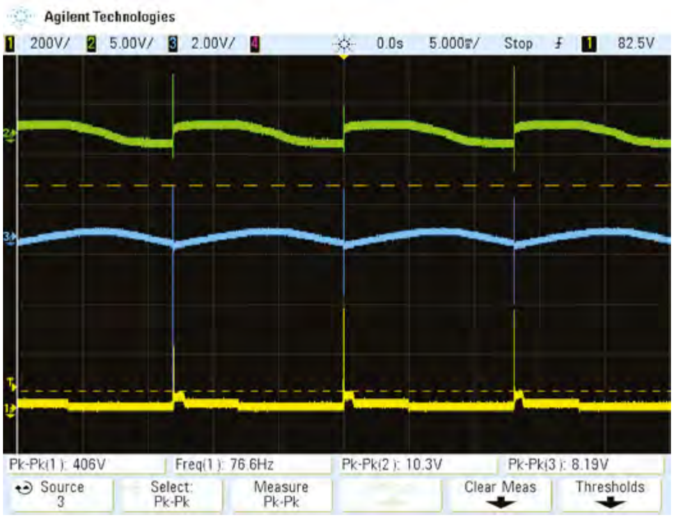
The dwell is adjustable from 129µs to 26ms and is set by monitoring the voltage at TP1. However, this is not necessary. In practice, you simply tweak VR1 to give the 'hottest' (ie, best looking!) spark discharge.

We are using the coil/spark test mode of the software for the *High-Energy Ignition* module. In the original circuit (February 2014) this was selected with LK2 (connected to

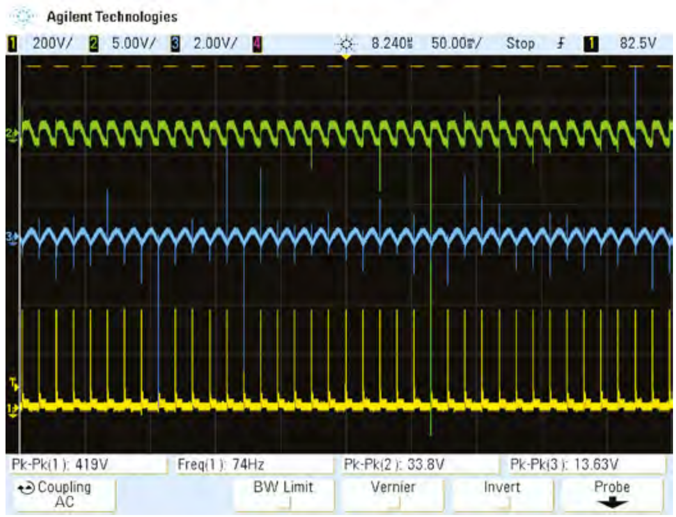


JACOB'S LADDER

Fig.1: the circuit incorporates additional components to protect the regulator against peak voltages which are superimposed on the positive supply line from the battery.



This scope grab shows the *Jacob's Ladder* circuit running at 76 sparks/second and a sweep speed of 5ms/div. The yellow trace shows the high voltages (around 400V peak) at the collector of the IGBT, while the green trace shows the fluctuation on the positive battery rail. The blue trace is the voltage across the transient voltage suppressor (TVS), showing that it is doing its job of protecting the regulator.



This shows the same waveforms as Scope1, but with the sweep speed slowed to 50ms/div. This captures more of the spike voltages on the supply lines. Without the input protection components, these spike voltages would be a great deal higher and would damage the regulator. Note that the spike voltages differ because each spark discharge takes a different path across the gap.

pin 12 of IC1). Since we don't need link options, the Jacob's Ladder version of the circuit merely has pin 12 connected to 0V, to achieve the same outcome.

Trimpot VR2 is used to set the spark rate, with a range of 15-75Hz (clockwise for increased frequency).

IC1 is powered from a regulated 5V supply derived using REG1, an LM2940CT-5 low-dropout regulator designed specifically for automotive use. It features both transient overvoltage and input polarity protection.

However, even though the LM2940CT-5 is a rugged regulator, it needs protection from the very high transient voltages which can be superimposed on the +12V line from the battery.

Hence, we have incorporated extra protection with the 10Ω series resistor and the 13.6V transient voltage suppressor (TVS). It clamps transient voltages to around 23V, a safe level for the regulator. Also, we have included a 1000µF 25V capacitor to provide further filtering for the input to the regulator.

Even so, it is most important that the +12V line to the module must be a separate wire to the battery positive terminal, as shown on the circuit. We have added these components after twice blowing the regulator and the microcontroller while having fun (um, doing important research) with our prototype *Jacob's Ladder*.

REG1 also has a 100µF filter capacitor at its output, required for stable operation.

By the way, note that word 'battery'. **Don't even think about running this from a mains supply**. A 12V SLA (sealed lead-acid) battery, as shown in the first photograph, is perfect for this project and will not only give a long operation time, it can be disconnected and recharged for the next zap session!

For longest life, you could run this from a 12V car battery, but they are rather heavy and difficult to lug around. And they can be messy.

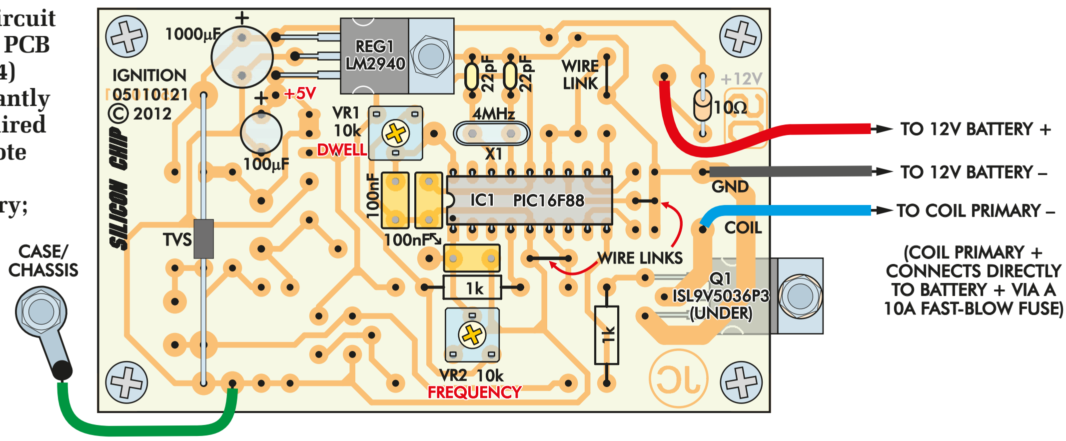
Building it

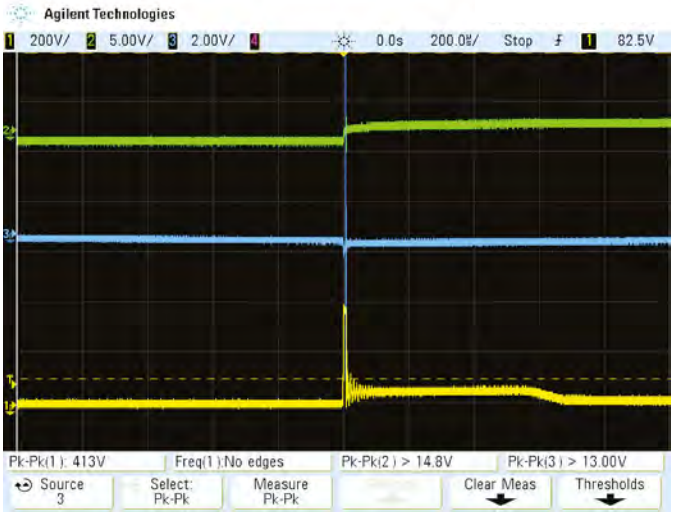
The *Jacob's Ladder* module is built on a PCB available from the *EPE PCB Service*, coded 05110121 and measuring 89mm × 53mm. This is housed in a 111mm × 60mm × 30mm diecast aluminium case to give a rugged assembly.

A cable gland at one end of the case provides the cable entry point for the positive and negative leads from the 12V battery and the lead from the IGBT's collector to one of the primary connections on the ignition coil.

There are significantly fewer components required for the *Jacob's Ladder*, so large areas of the PCB are unpopulated.

Fig.2: the *Jacob's Ladder* circuit uses the electronic ignition PCB (from February/March 2014) but as you can see, significantly fewer components are required (hence the empty holes). Note the wiring connection for the + side of the coil primary; it connects directly to the battery + terminal via a fuse. Don't be tempted to run the PCB wiring from this fuse. Keep the wiring separate.





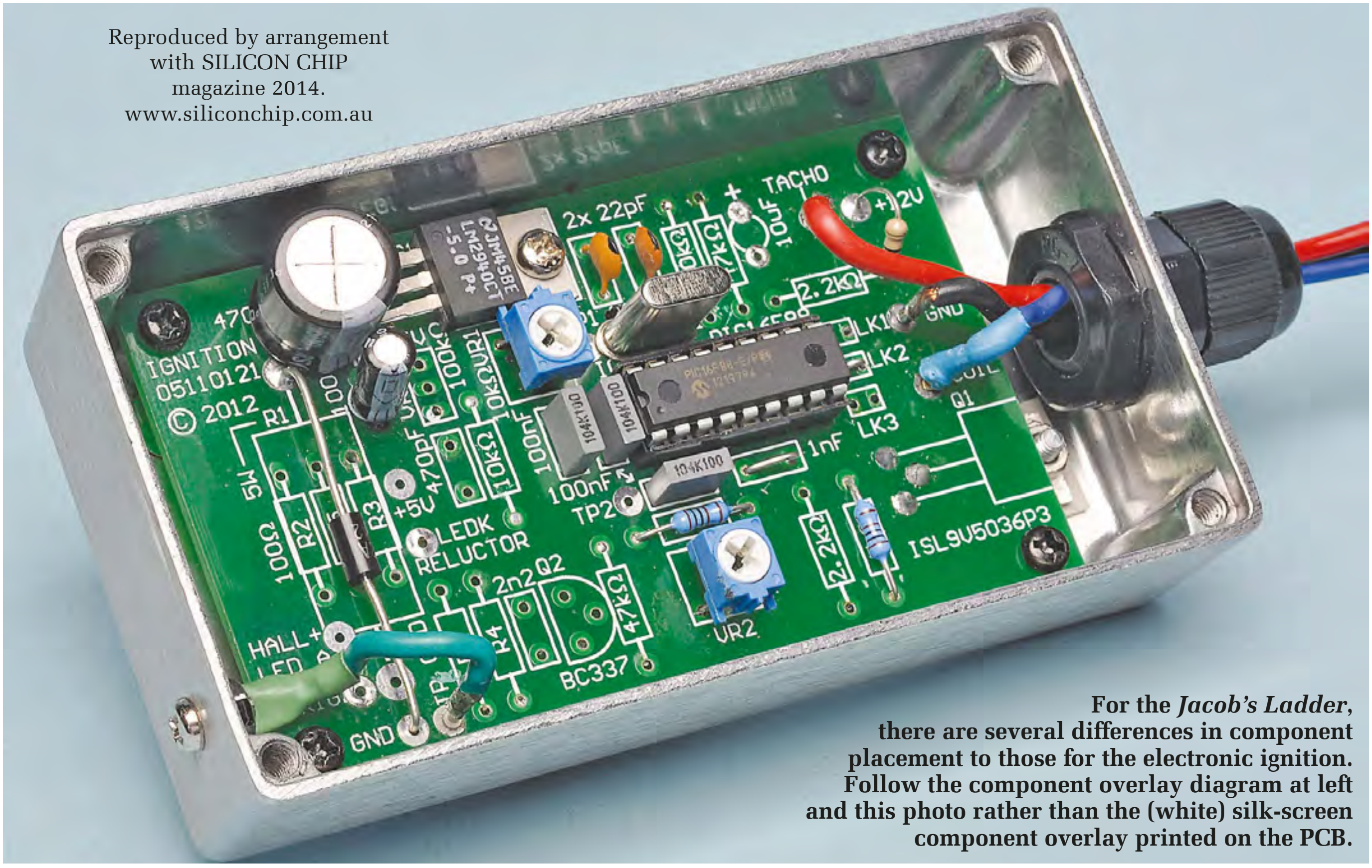
Again, same waveforms as before, but with sweep speed upped to 200µs/div. Here you can see the ringing of the coil primary after the main spike. The voltage is clipped to 413V by the protection limiting inside the IGBT.

Same conditions as the grab at left, but with sweep speed upped again to 20µs/div. Here we see that the spike voltages across the supply lines are very fast and both are actually clipped by the scope.

The first step is to check the PCB for any defects and in the unlikely case that there are any problems, fix them. Then install the components shown in the diagram of Fig.2. If you are using a PCB supplied by *EPE* you will find that some of the components to be installed are not as indicated on the silk-screened component layout on the top of the PCB itself.

For example, the red wire from the positive terminal of the battery does not connect to the +12V pin at the top right-hand corner of the PCB. Instead, it connects to the PC pin marked 'Tacho' which is not being used for its original function in this *Jacob's Ladder* version of the circuit. We will detail the other component variations as we go through the assembly procedure.

Begin the assembly by installing the four PC stakes at the external wiring points – ie, Tacho, GND, COIL, and TP GND. Then install three short wire links. One goes in the position labelled LK2 at one end of the microcontroller, another is wired in the position for the 1nF capacitor adjacent to pins 5, 6 and 7 (of the microcontroller) while the third replaces the 10µF capacitor near



For the *Jacob's Ladder*, there are several differences in component placement to those for the electronic ignition. Follow the component overlay diagram at left and this photo rather than the (white) silk-screen component overlay printed on the PCB.

The completed Jacob's Ladder in daylight, showing which bits connect to where! All the circuitry is inside the metal box, with the twin ignition coil mounted on top, spaced above the box by about 15mm with the aid of a pair of precision (Coke bottle cap) spacers. These are needed to allow the wires from the circuit to connect via spade lugs under the coil. Using crocodile clips on the coil terminals allows a great deal of flexibility when positioning the vertical (spark) wires, for best visual effect.

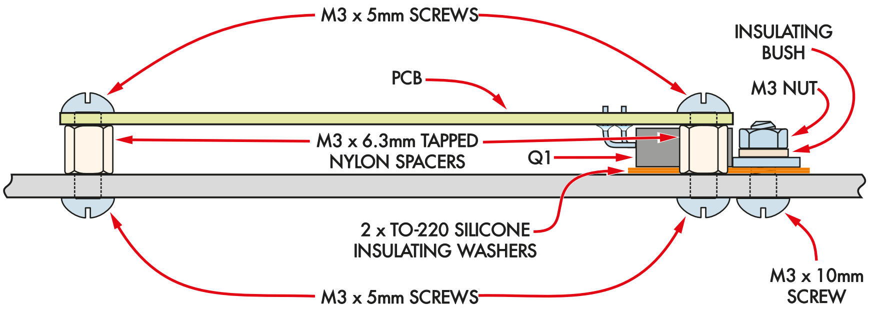


Fig.3: here's how the IGBT is mounted underneath the PCB. 6.3mm nylon spacers hold the PCB at the right height and also insulate it from the case. Two silicone insulating washers are used to insulate the IGBT from the case.

the original 'TACHO' PC stake. These are followed by the three resistors.

Follow with the IC socket, making sure it is oriented correctly – but don't install the PIC micro yet.

The capacitors can go in next. Orient the two electrolytics as shown) then install crystal X1 and trimpots VR1 and VR2. The transient voltage suppressor can be installed either way round as it is not a polarised device.

Regulator REG1 can then go in. Be sure to fasten REG1's tab to the PCB using an M3 × 10mm machine screw and nut before soldering its leads.

IGBT mounting details

Fig.3 shows the mounting details for IGBT transistor Q1. It's secured to the base of the case, with its leads bent at right angles and passing up through the underside of the PCB.

For the time being, simply bend Q1's leads upwards through 90° and test fit it to the PCB but don't solder its leads yet. Its tab mounting hole must be clear of the edge of the PCB, as shown in the diagrams.

Then fit the PCB assembly inside the case and slide it to the left as far it will go, to leave room for Q1. The mounting hole positions for the PCB and Q1's tab can then be marked inside the case, after which the PCB can be removed and the holes drilled to 3mm (hint: use a small pilot drill first).

Deburr these holes using an oversize drill. In particular, Q1's mounting hole must be slightly countersunk inside the case to completely remove any sharp edges.

The transistor's mounting area on the case should also be carefully smoothed using fine emery paper. These measures are necessary to prevent the insulating washers that go between Q1's metal tab and the case from being punctured by metal swarf or by a high-voltage arc during operation.

Having drilled the base, the next step is to mark out and drill the hole in the case for the cable gland. This hole is centrally located at the end of the case where the IGBT is mounted. It should be carefully reamed to size so that the cable gland is a close fit.

You will also have to drill a 3mm hole for the earth connection in the other end of the case – see photos.

Installing the PCB

Once the case has been drilled, fit 6.3mm tapped **nylon** stand-offs to the PCB's corner mounting holes using M3 × 5mm machine screws.

That done, the next step is to fasten Q1 in place. As shown in Fig.3, its metal tab is insulated from the case using two TO-220 silicone washers and an insulating bush, and it's secured using an M3 × 10mm screw and nut.

Do this screw up finger-tight, then install the PCB in the case with Q1's leads passing up through their respective mounting holes.

The PCB can now be secured in place using four more M3 × 5mm machine screws, after which you can firmly tighten Q1's mounting screw (make sure the tab remains centred on the insulating washers).

Finally, use your multimeter to confirm that Q1's tab is indeed isolated from the metal case (you must get an open-circuit reading), then solder its leads to the pads on top of the PCB.

External wiring

All that remains now is to run the external wiring. You will need to run three leads through the cable gland and solder them to the relevant PC stakes for the power, coil and input trigger connections.

Don't be tempted to use mains cable for the three leads – brown, blue and green/yellow should never be used for anything but mains.

The earth connection from the PCB goes to a solder lug that's secured to the case using an M3 × 10mm screw, nut and star washer.

Initial checks and adjustments

Now for an initial smoke test – apply power to the unit (between +12V and GND) and use your DMM to check the voltage between the +5V PC stake and GND. It should measure between 4.85V and 5.25V. If so, switch off and insert the programmed PIC (IC1) into its socket, making sure it goes in the right way around.

You can now do some more tests by connecting the car's ignition coil between the +12V battery terminal via a 10A in-line fuse. The unit should be powered from a 12V car or motorcycle battery or a sealed lead acid battery, **NOT from a mains power supply.**

The negative coil wire (shown in blue on the diagram) connects to the 'coil' terminal on the PCB.

Before connecting the +12V power, set the dwell trimpot (VR1) fully anticlockwise. Then apply power and slowly adjust VR1 clockwise. The sparks should start and gradually increase in energy with increased dwell. Stop adjusting VR1 when the spark energy reaches its maximum.

You can also set the spark frequency using VR2 but we found the best result was with it set to maximum, ie, clockwise.

Mounting the ignition coil

We mounted the ignition coil onto the lid of the case using two M3 bolts and nuts.

Since the two primary connections are recessed underneath the coil, we had to space it off the lid of the case and we used two soft drink bottle lids for this. Brand is unimportant – just make sure that you do not use metal caps!

We made the connections to the coil primary with red crimped male spade connectors.

Finally, we fitted a pair of crocodile clips with screws with stiff wire, about 250mm long.

You can dispense with the plastic finger grips since the sparks jump between the crocodile clips and then climb the wires.

Note how the clips fasten to the coil terminals in our photos – if you mount them the other way (ie, with the bodies closer together) you'll probably find that the sparks jump across the crocodile clips but don't climb up the wires.

In fact, you'll probably have to experiment somewhat with the wire positions to get the climbing action reliable.

We found that very close to parallel was right. We also bent the top 10mm or so of the wires away from the ladder, as you can also clearly see in the pic.

Want to use taller wires? Give it a go – but if they are too tall it becomes unwieldy.

Fitting a 'chimney'

We also experimented with a clear plastic (acrylic?) tube over the whole ladder. This has the added advantage of creating a vertical airflow as the air inside the tube heats up. This adds to the rising spark effect.

The biggest problem was finding a clear tube (a) big enough – it needs to be about 150mm inside diameter and (b) cheap enough to warrant its use. In the end, being somewhat tight in both the wallet and time departments, we gave the idea away!

However, if you can find such a tube it will add to the spectacle and should also assist the spark if there is any form of breeze. We found wind impedes the climbing effect. The tube needs to be open-ended top and bottom to create the draught.

An acrylic tube will also assist somewhat in keeping the zaps contained – but don't rely on it! A thick acrylic tube should have hundreds of kilovolts of insulation, but you can never be sure. The moral of the story is, keep your fingers (and anyone else's!) away from the vertical wires.

Before making any adjustments – moving the wires for a better display, for example – disconnect the battery and make sure gravity or any other force cannot accidentally make a connection when you don't want it to!

As we said earlier, accidentally touching the wires while in operation (why would anyone touch them deliberately?!!) will certainly give you some energy you didn't know you had – and may even (perish the thought!) cause you to issue forth with naughty words!

Parts list – Jacob's Ladder

1 VS Commodore/Camaro Firebird V6 ignition coil (source from wrecker or eBay)
1 PCB, available from the *EPE PCB Service*, code 05110121, 89mm × 53mm
1 diecast aluminium case, 111mm × 60mm × 30mm
1 cable gland to suit 3-6mm cable
1 transistor insulating bush
2 TO-220 3kV silicone insulating washers
1 4MHz HC-49 crystal (X1)
1 18-pin DIL IC socket
1 in-line 3AG fuse holder and 10A 3AG fuse (fast-blow)
1 solder lug
2 crocodile clips with screws
2 250mm lengths approx. 1.5mm diameter steel wire
2 red crimp spade lugs
4 6.3mm tapped nylon standoffs
8 M3 × 5mm screws
3 M3 × 10mm screws and nuts
2 M3 × 30mm screws and nuts
1 M3 star washer
4 PC stakes
1 500mm length of red automotive wire
1 200mm length of black automotive wire
1 200mm length of blue automotive wire

Semiconductors
1 PIC16F88-E/P microcontroller programmed with 0511012A.hex (IC1)
1 ISL9V5036P3 ignition IGBT (Q1)
1 LM2940CT-5 low drop out 5V regulator (REG1)
1 13.6V transient voltage suppressor (TVS)

Capacitors
1 1000µF 25V PC electrolytic
1 100µF 16V PC electrolytic
3 100nF MKT (code: 104)
2 22pF ceramic (code: 22)

Resistors (0.25W 1%)
2 1kΩ
(code: brown black black brown or brown black red brown)
1 10Ω
(code: brown black black gold brown or brown black black brown)
2 10kΩ mini horizontal trimpots (VR1,VR2)

By JIM ROWE

Deluxe GPS 1pps Timebase for frequency counters

Were you interested in the precision GPS timebase featured in the March 2014 issue? That was the 'no frills' version. Here we present a ***Deluxe GPS 1pps Timebase,*** which also suits our recently described ***12-Digit Frequency Counter***. It not only provides the same near-atomic-clock-accuracy 1pps pulses for the counter's timebase, but also extracts the NMEA 0183 data stream from the GPS satellites for processing on your PC.

MEASUREMENT ACCURACY IS the prime reason for building either the original no-frills timebase or this new *Deluxe GPS 1pps Timebase.* Each represents the simplest and most economical way to match accuracy to the resolution of the *12-Digit High-Resolution Frequency Counter* described in the January and February 2013 issues of *EPE*.

By using a GPS 1pps timebase, the counter can achieve a measurement accuracy approaching ±1 part in 10^{11}. That's nearing atomic clock territory.

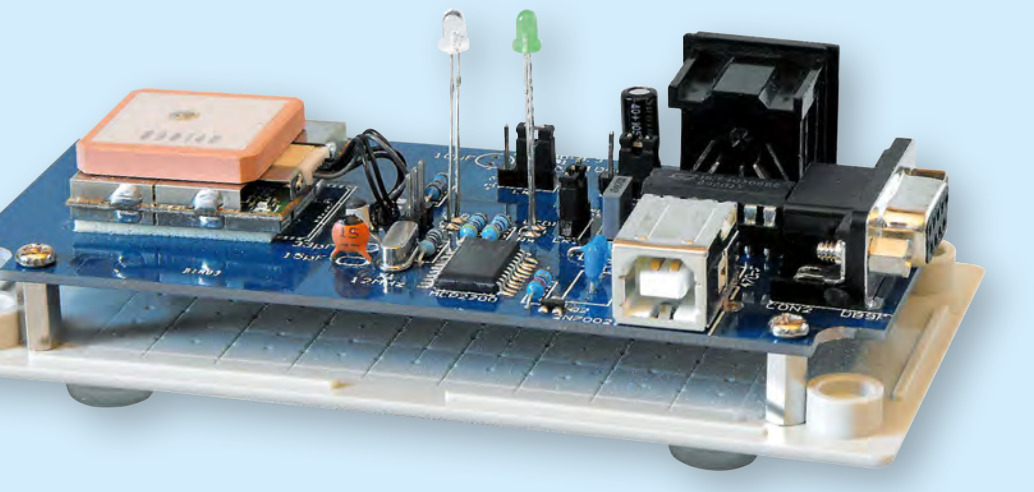
Our February no-frills design comprised little more than a cheap GPS receiver module with the all-important 1pps output, plus a handful of components to provide the module with power and to buffer the output pulses. Despite its simplicity, this first *GPS 1pps Timebase* works extremely well.

But while it was under development, we also had the intention of describing this deluxe version, which would also have the NMEA 0183 stream of navigation data. This data is provided by virtually all low-cost GPS modules, along with the 1pps pulses, but separated from them.

So that's the basis of the new *Deluxe GPS 1pps Timebase* described here. The NMEA data is fed out to a USB socket and it's relatively easy to analyse this data stream and extract the current UTC (Universal Time Co-ordinated) and date, along with such things as longitude, latitude, altitude and the number of GPS satellites in view. In addition, the PC can display the signal-to-noise ratio (SNR) of the signals from the satellites and even the quality of the 'fix' that the GPS module is currently able to achieve using them. This helps to confirm the accuracy and reliability of the 1pps pulses as a timebase.

GPS clock driver

Back in the May and June 2011 issues of *EPE*, we described a *GPS Clock Driver* module. This took the NMEA 0183 data stream from a low-cost GPS receiver module and made it available for driving our *6-Digit GPS Clock*. Alternatively, it could be fed to a PC via a 'legacy' serial port. There were a number of freeware and shareware software applications available at the time which could be used to analyse the data stream and display much of the useful information.



The parts are all installed on a small PCB which is then mounted on the lid of a UB3 jiffy box. The lid then acts as the base of the completed unit shown at left.

So one way of improving the March 2014 *GPS 1pps Timebase* would be to simply 'bolt on' the relevant parts of the June 2011 clock driver circuit, to make the NMEA 0183 data stream from the GPS receiver module available (as well as the 1pps pulses). This would allow the *GPS 1pps Timebase* unit to drive the 2011 clock or the serial port of a PC, as well as the timebase of the *12-Digit Frequency Counter*.

The problem with this approach is that most of today's PCs don't provide an RS232 serial port; they only have USB ports. Our deluxe unit features a USB port as well as an RS232 port, so it can be connected to a wide range of computers and laptops.

This makes it easy to monitor the receiver's 'fix' status by running a freeware application called GPS Diagnostics 1.05 (there are many others, but we have found this one to be excellent).

As shown in the accompanying photos, the *Deluxe GPS 1pps Timebase* is housed in a small plastic case. It can be powered via its USB port or from the *12-digit Frequency Counter*. The latter approach is appropriate when you are not using your computer to monitor the GPS signal status.

Circuit details

Fig.1 shows the full circuit details of the *Deluxe GPS 1pps Timebase*. It's still fairly simple, but again that's because all the complex circuitry needed to receive the signals from the GPS satellites and derive both the 1pps (1Hz) pulses and the NMEA 0183 data stream from them is buried deep inside the GPS receiver module.

We are again specifying either of two low-cost receiver modules which are currently available from various suppliers: the GlobalSat EM-406A module which is available from **amazon.co.uk** for around £40 or the Fastrax UP501 module from **uk.rs-online.com.** This is smaller and also priced at £40, but is becoming harder to buy. The project is also compatible with various other receiver modules, if you find the EM-406A or the UP501 out of stock.

The type of GPS receiver module required is one that incorporates its own ceramic 'patch' antenna for the UHF signals from the GPS satellites, while also providing an output for the 1pps (pulse per second) time pulses. It can operate from a DC supply of either 5.0V or 3.3V. A few currently available modules are listed in a panel elsewhere in this article.

The EM-406A has its own built-in GPS patch antenna and operates directly from 5V DC. It features the SiRF Star III high-performance GPS chip set, very high sensitivity and a relatively fast time to first fix (from a cold start).

The UP501 and other compatible GPS modules operate from 3.3V DC, so we have made provision for fitting a 3.3V LDO (low drop-out) regulator (REG1) to provide this lower voltage for modules that need it. In this case, we are using an LP2950-3.3 regulator, which comes in a TO-92 package.

Apart from the power supply arrangements, there is a 40106B hex CMOS Schmitt inverter (IC1), used for buffering both the 1pps timebase pulses for the counter and the NMEA 0183 data stream. IC1c is the buffer for the NMEA data, with its output going to pin 2 of CON2. The other five inverters in IC1 are used for the 1pps pulse buffer and as a level translator, with IC1a used as an optional inverter to restore pulse polarity if necessary. As shown, IC1b, IC1d, IC1e and IC1f are connected in parallel and drive pin 3 of CON1, which goes to the counter's external timebase input.

Parts List

1 PCB, available from the *EPE PCB Service,* code 04104131, 121mm × 57mm
1 UB3 jiffy box, 130mm × 68mm × 44mm
1 GPS receiver module with in-built patch antenna and 1pps output
4 3-pin SIL pin headers (LK1-LK4)
4 jumper shunts to match
1 12MHz crystal, HC-49US (X1)
1 5-pin DIN socket, PCB-mount (CON1)
1 DB9F socket, PCB-mount (CON2)
1 USB type B socket, PCB-mount (CON3)
1 14-pin DIL IC socket
4 M3 × 10mm tapped metal spacers
4 self-adhesive rubber feet
8 M3 × 6mm machine screws
25 × 25mm double-sided adhesive foam (to secure GPS module)

Semiconductors
1 40106B hex Schmitt inverter (IC1)
1 MCP2200 USB2.0 to serial converter (IC2)
1 LP2950-3.3 LDO regulator (REG1*)
1 NX2301P P-channel MOSFET (Q1)
1 2N7002 N-channel MOSFET (Q2)
1 3mm green LED (LED1)
1 3mm red LED (LED2)

Capacitors
2 10µF 16V RB electrolytic
1 470nF MMC
2 100nF MMC or MKT
1 33pF NP0 ceramic
1 15pF NP0 ceramic

Resistors (0.25W 1%)
1 47kΩ 3 470Ω
1 10kΩ 1 22Ω
1 1kΩ

*Only required if you are using a GPS module which requires a 3.3V supply

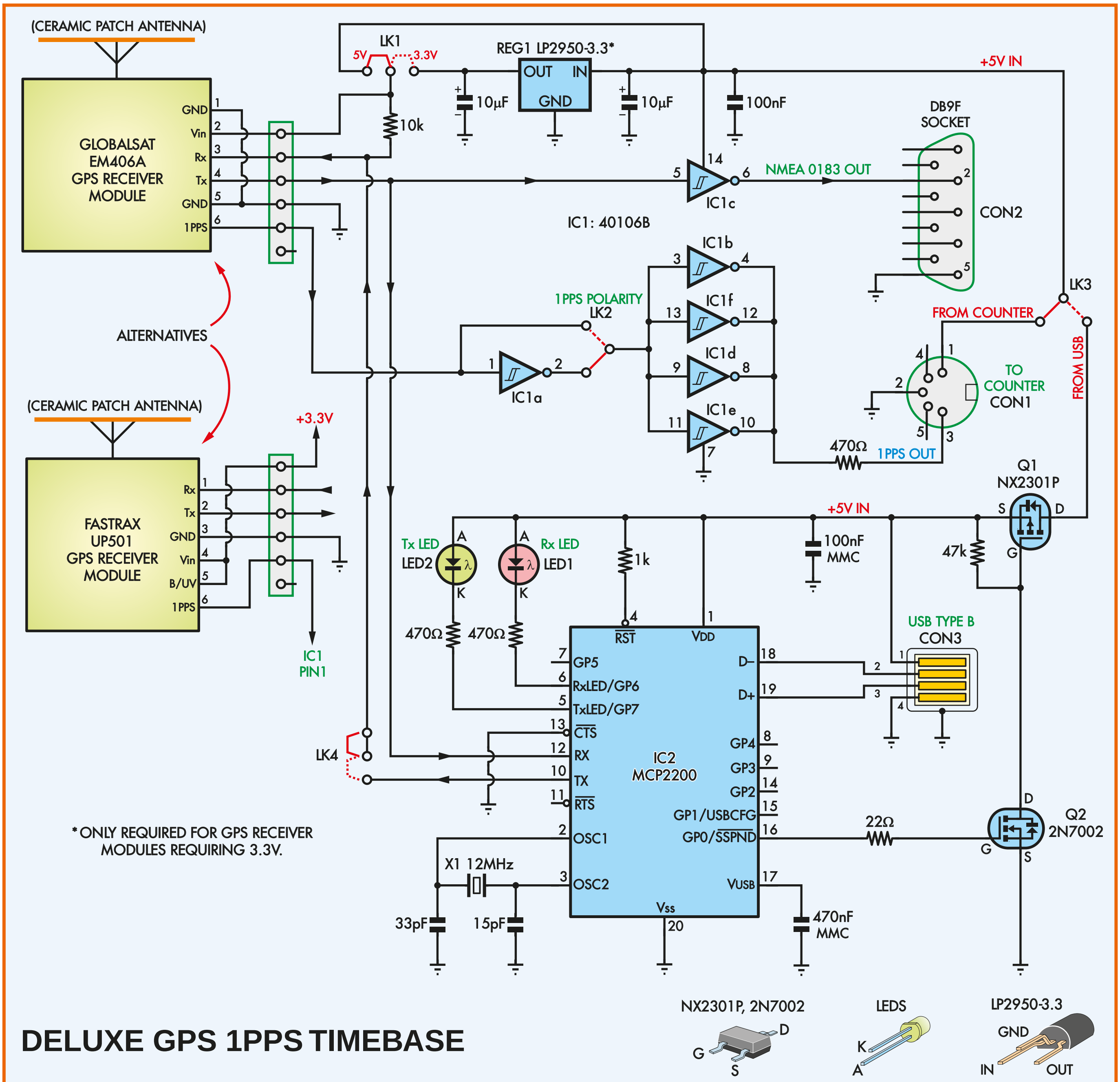


Fig.1: the circuit consists of the GPS receiver module plus a hex CMOS Schmitt trigger inverter to buffer the 1pps (1Hz) pulses and NMEA data from the module. The NMEA data is also fed to IC2 which drives the USB serial port.

As with the no-frills circuit, link LK2 is used to allow the 1pps pulses to be either inverted or not by the buffer, so that their leading edges are positive-going regardless of their polarity out of the GPS module (some modules may output them as inverted).

Basically, we need to ensure that the leading edges of the 1pps pulses fed to the *12-Digit Frequency Counter* are positive-going. That's because it's the leading edges of the pulses that are locked closely to the 'atomic time' provided by the GPS satellites.

The remaining circuitry in Fig.1 is used to provide the USB serial port. Here we are using a Microchip MCP-2200, a dedicated USB2.0-to-UART Protocol Converter device. It appears to be similar to a PIC18F14K50 microcontroller chip but is 'hard wired' to perform USB/serial and serial/USB conversion, so that when it's linked to the USB port of a PC it behaves as a 'virtual COM port device'.

As a result, Windows will communicate with the MCP2200 via a virtual COM port (VCP) driver. In addition, Microchip has a freeware 'Configuration Utility' program which can be used to configure the MCP2200 in terms of baud rate, data format and so on. We will describe this in greater detail later.

The MCP2200 (IC2) needs a 12MHz crystal (X1) for its clock oscillator. This crystal is connected between pins 2 and 3, along with two small NP0 ceramic capacitors. It also needs a 470nF MMC bypass capacitor connected between its V_{USB} pin (pin 17) and ground, together with a 100nF

MMC capacitor bypassing the +5V rail from the PC's USB port (ie, pin 1 of CON3).

The D– and D+ data lines from CON3 connect directly to pins 18 and 19 of IC2, while the NMEA data stream from the GPS receiver module is fed directly to pin 12 of IC2. IC2 converts this data stream into USB packets for transmission to the PC via CON3.

NMEA commands are also sent back from the PC via the USB cable and these emerge from pin 10 of IC2. These can be fed back to the Rx input of the GPS receiver module when link LK4 is used to complete the circuit. In this application, we don't need to send any commands to the GPS receiver module – we simply use its default operating configuration.

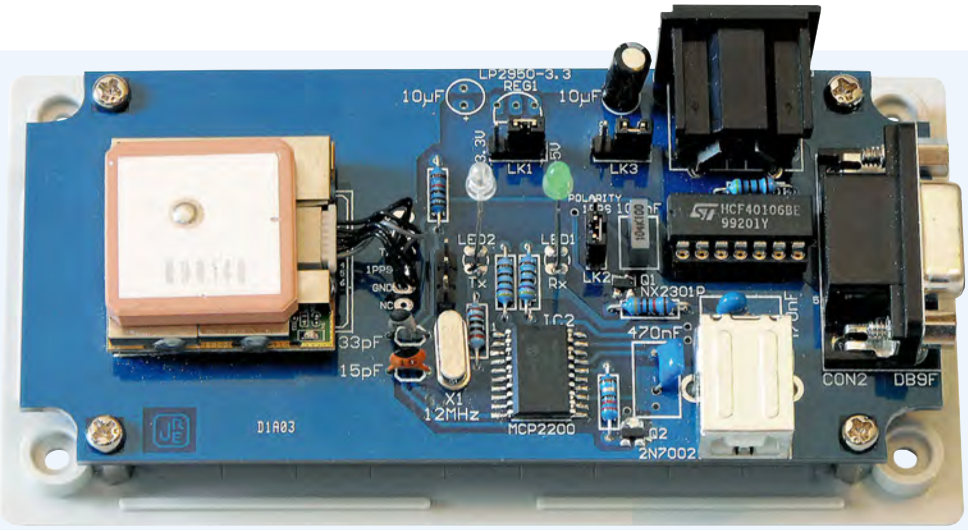
However, we found that when this connection was made in addition to the main Tx-to-Rx connection to pin 12 of IC2, there could be a conflict whereby IC2 could prevent the GPS receiver module from finding a 'fix'. In addition, the GPS receiver could prevent IC2 from configuring and enumerating correctly. So it seems best to leave LK4 in the 'open' position, as shown in Fig.1 (and Fig.2).

LED1 (receive) and LED2 (transmit) are driven from pins 6 and 5 of IC2. These LEDs flash when data is passing through IC2 in one direction or the other.

The remaining part of the circuit involves MOSFETs Q1 and Q2, which are used to allow IC2 to control the +5V power fed from USB socket CON3 to link LK3 (this link is used to select the power source for the GPS receiver module and IC1). This is done to conform to the USB 2.0 requirement that current drain from the PC's USB port drops to less than 2.5mA when the PC's USB host controller holds the device in 'suspended' mode.

IC2's SSPND output (pin 16) is connected to Q2's gate via a 22Ω suppressor resistor, so that Q2 is only turned on when IC2 receives a 'wake up from suspension' directive. Then, when Q2 turns on, it turns on Q1, which makes the connection between pin 1 of CON3 and LK3. So, if LK3 is in the power 'From USB' position, (rather than 'From Counter' position), the GPS receiver module will only receive power when (a) the project is connected to a USB port on a PC; (b) the PC is powered up; and (c) software is running on the PC and 'listening' to the GPS data stream, so that IC2 is not in suspended mode. Note that the GPS receiver module can take over a minute to get a 'fix' after power is applied.

Alternatively, by fitting LK3 to the 'From Counter' position, the upper part of the circuit can be powered from either the counter or an external plugpack supply (via CON1). This means that you don't have to connect the unit to a PC in order to simply derive 1pps pulses.



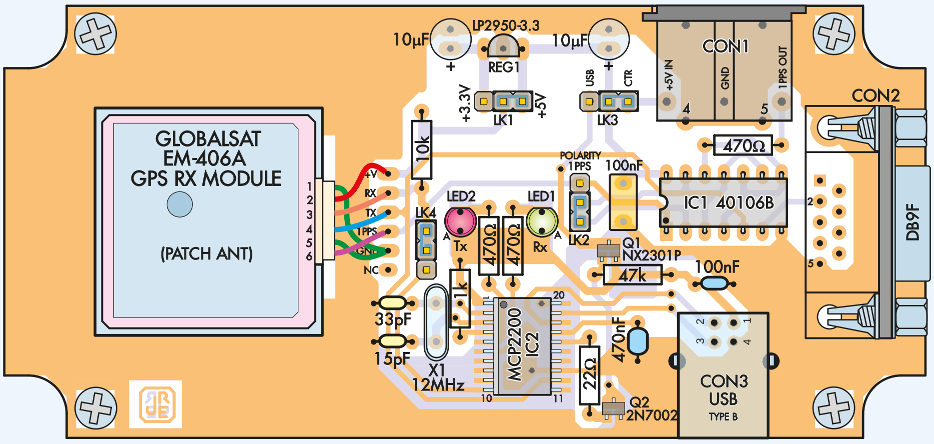


Fig.2: follow this layout diagram to build the unit. Omit REG1 and the 10µF capacitor to its left if you are using the Globalsat EM-406A module and install LK1 in the +5V position. Alternatively, install REG1 and the capacitor if your GPS module requires a 3.3V supply and fit LK1 to the +3.3V position.

Building it

All the parts for the *Deluxe GPS 1pps Timebase* fit on a PCB available from the *EPE PCB Service*, coded 04104131 and measuring 122mm × 57mm. Fig.2 shows the PCB parts layout diagram, while Fig.3 shows the pin connections for the GlobalSat EM-406A and Fastrax UP501 GPS receiver modules. Note that half of the PCB is for mounting the GPS module, which is held in place using double-sided adhesive foam.

Begin by fitting SMD components IC2, Q1 and Q2 to the PCB, as it is much easier to do this before any other parts are fitted. Take the usual precautions when soldering these parts, ie, use an earthed soldering iron with a fine-tipped bit. Tack-solder one or two device leads first, so that the device is held in position while you solder the rest of the leads. You then re-solder the original tacked leads to ensure reliable joints.

Don't worry if you accidentally bridge two or more SMD device leads with solder during this procedure. These bridges can subsequently be removed quite easily by pressing solder wick braid against the bridged leads using the tip of your soldering iron. This sucks up the excess solder while leaving the solder joining the leads to the PCB pads underneath in place.

Once the SMD parts have been installed, add the SIL pin headers for links LK1-LK4, followed by the

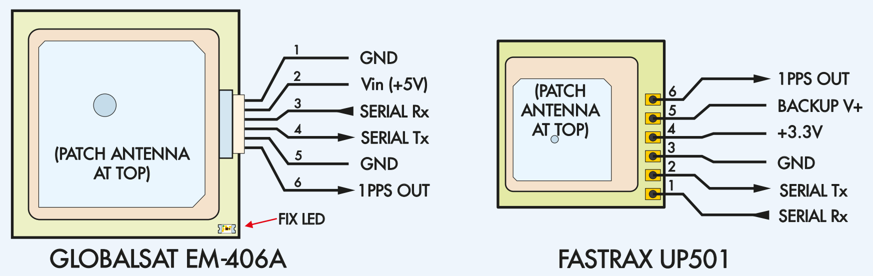


Fig.3: the pin connections for the GlobalSat EM-406A and Fastrax UP501 GPS modules. Check the pin connections if you use a different module.

resistors, capacitors and the 12MHz crystal. A 14-pin socket for IC1 can then be fitted – make sure it's oriented as shown.

Connectors CON1-CON3 can then go in, followed by LED1 and LED2. The latter are mounted vertically above the PCB, with their leads left at full length so that they later protrude through their matching holes in the case (see Fig.4).

Voltage regulator option

Regulator REG1 and the 10µF electrolytic capacitor to its left are installed only if the GPS receiver module you are using requires a 3.3V DC supply rather than a 5V supply. This means that if you are using the EM-406A module, you won't need to fit REG1 or that 10µF capacitor.

By contrast, the regulator and the capacitor must be installed if you are using the UP501 receiver module, since this runs off 3.3V. The same goes for the Digilent PmodGPS and RF Solutions GPS-622R GPS modules.

The GPS receiver module is installed last, but before doing this, you need to make the connections between its output pads (or lead wires) and the relevant pads on the PCB (ie, just to the left of LK4). Fig.3 shows the outputs for the Globalsat EM-406A and Fastrax UP501 modules. Be sure to connect these to their matching pads on the PCB.

The EM-406A module comes with a short 6-wire ribbon cable fitted with a sub-miniature 6-pin plug at each end. One of these plugs connects directly to the EM-406A's output socket. The plug at the other end of the cable is cut off and the six wires stripped and tinned before soldering them to their PCB pads.

By contrast, the UP-501 module just has a row of pads along one edge of its PCB. It's connected by first cutting six 25mm-lengths of light-duty hookup wire (eg, from a ribbon cable), then carefully stripping and tinning all the wire ends before soldering the leads into place.

Don't forget to match the output leads from the GPS module to the PCB pads (see Figs.2 and 3), as the connections are not 'straight through'.

Once all the connections have been made, the GPS receiver module can be secured to the top of the PCB using a 25mm-square piece of double-sided adhesive foam – see Fig.4. Make sure you attach the module with its patch antenna facing up – it won't work very well if it faces down!

Fitting the links

LK1's shunt position depends on the supply voltage (5V or 3.3V) required for the GPS receiver module you're using, while LK2's position depends on the polarity of the 1pps output pulses from the GPS receiver. In most cases, LK2 will need to be to the lower position (ie, nearest Q1).

LK3's position depends on just how you plan to power the GPS receiver module and IC1 (ie, the 1pps timebase section of the circuit). If you only intend using this part of the circuit when the unit is connected to a PC via a USB cable, then LK3 can be fitted in the USB (lefthand) position (ie, the circuit is powered from the PC's USB port).

Alternatively, if you want to use this part of the circuit continuously (eg, whenever the *12-Digit Frequency Counter* is on but without having to fire up the PC), you'll need to fit LK3 in the righthand CTR (From Counter) position and power the unit either from the counter or an external 5V plugpack via CON1.

Finally, LK4 should almost always be fitted to the upper position, to break the connection between pin 10 of IC2 and the Rx input of the GPS module.

Preparing the box

Fig.4 shows how the PCB assembly is fitted inside a standard UB3 jiffy box. The completed unit can be mounted near a window to get a good 'view' of the sky.

As shown, the PCB is mounted on the lid of the box, which then becomes the base. The main part of the box then fits down over the lid/board assembly, to act as a dust cover.

Fig.5 shows the drilling details for the box. Four mounting holes have to be drilled in the lid to accept the PCB, while two holes must be drilled through the top of the main box section for the LEDs. In addition, you have to drill a hole in the rear side of the box and make cut-outs in the front side and righthand end.

Use a small (eg, 1.5mm) pilot drill to start all the holes, then drill the 3mm holes out to the correct size. The hole in the rear side of the box can be enlarged to the correct size

Compatible GPS receiver modules

The following GPS receiver modules should be compatible with this project

- GlobalSat EM-406A: 30 x 30 x 10.5mm including patch antenna. Operates from 5V DC with a current drain of 44mA. Provides a 1pps output and a 'fix' indicator LED. Rated sensitivity –159dBm.
- Digilent PmodGPS: approximately 30 x 55 x 12mm including patch antenna. Operates from 3.3V DC with a current drain of 24/30mA. Provides a 1pps output and a 'fix' indicator LED. Rated sensitivity –165dBm.
- RF Solutions GPS-622R: 43 x 31 x 6mm including patch antenna. Operates from 3.3V DC with a current drain of 23/50mA. Provides a 1pps output and a 'fix' indicator LED. Rated sensitivity –148dBm/–165dBm.
- Fastrax UP501: 22 x 22 x 8mm including patch antenna. Operates from 3.3V DC with a current drain of 23mA. Provides a 1pps output. Rated sensitivity –165dBm.

Note that for use in this project, the GPS receiver module should have a built-in ceramic patch antenna and also provide an output for the GPS-derived 1pps pulses. Not all GPS modules currently available provide both of these features.

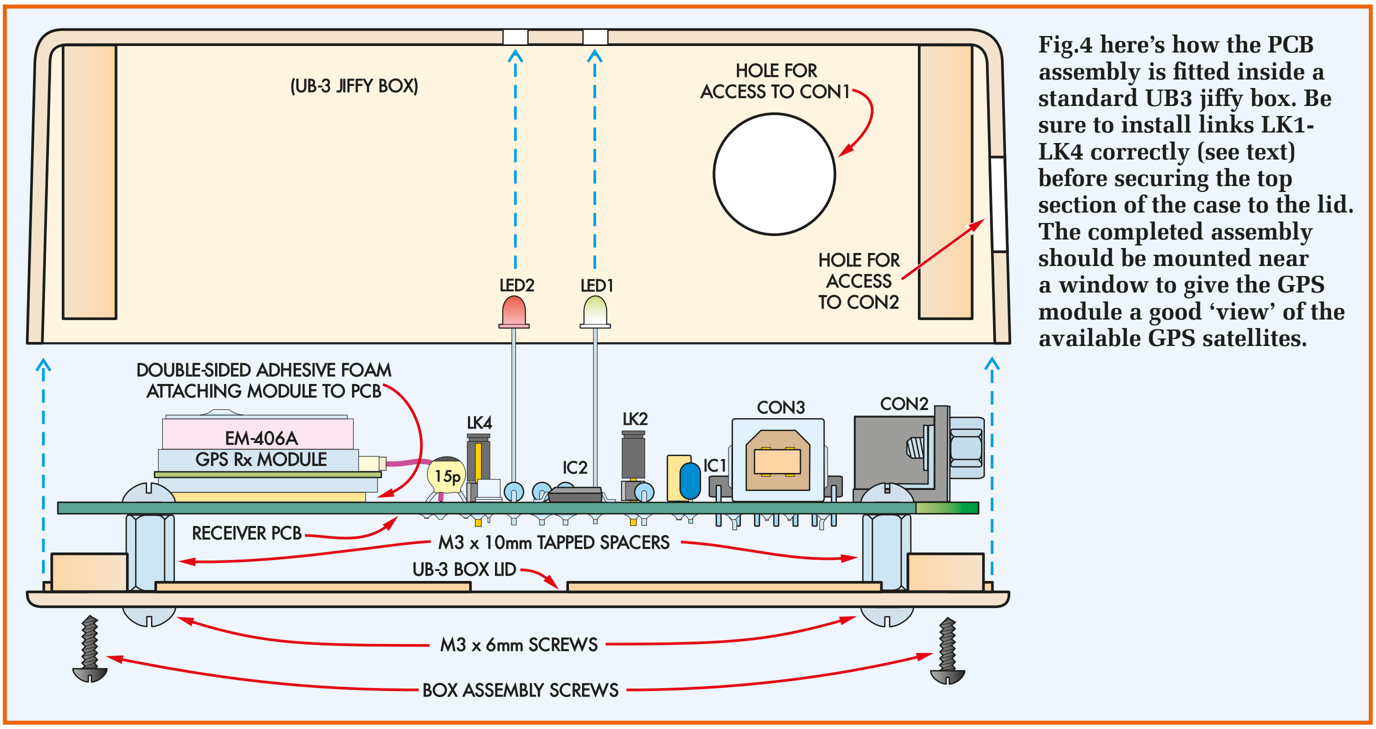


Fig.4 here's how the PCB assembly is fitted inside a standard UB3 jiffy box. Be sure to install links LK1-LK4 correctly (see text) before securing the top section of the case to the lid. The completed assembly should be mounted near a window to give the GPS module a good 'view' of the available GPS satellites.

(16mm) using a tapered reamer. The two square cut-outs can be made by drilling a series of small holes around the inside perimeter, then knocking out the centre piece and carefully filing the inside edges.

If you are using a GPS receiver module with a 'fix' indicator LED, you might want to drill an additional hole in the adjacent side of the box, so that you can view this LED to confirm that the receiver does indeed have a fix. The prototype shown in the photos uses an EM-406A module, which does have such an LED in the lower righthand corner – see Fig.3. That's the reason for the 5mm hole you can see in the front of the box, located 45mm from the lefthand end and 20mm up from the outer surface of the lid.

The UP501 module doesn't have a 'fix' LED, so there's no need to drill this hole. However, many other modules do have this LED and the hole location will depend on the LED's location on your particular module.

Once the box holes have been drilled, the PCB assembly can be mounted on the lid on four M3 × 10mm tapped spacers and secured using M3 × 6mm machine screws. That done, check that you've fitted the jumper shunts to each of the four SIL pin headers (for LK1-LK4) as required (see above). The box can then be lowered down onto the lid, taking care to ensure that LED1 and LED2 protrude through their respective holes at the top, and the assembly secured by fitting the four supplied self-tapping screws.

Finally, fit four small adhesive rubber feet to the lid (which now becomes the base) to prevent scratches due to the protruding screw heads. Your *Deluxe GPS 1pps Timebase* is now complete.

Other uses for this project

The NMEA output of this Deluxe GPS 1pps Timebase can be used with a range of navigation software and free Windows GPS-related software packages.

- For nautical chart and navigation software that works with NMEA-compatible GPS units see: **http://capcode.sourceforge.net/**
- For your position on Google Maps as you move (multiple options) see:
  http://mboffin.com/earthbridge/
  http://download.cnet.com/Google-Maps-with-GPS-Tracker/3000-12940_4-10494227.html?tag=keyword.feed&part=rss&subj=dl.gps
  http://blog.geoblogspot.com/2008/09/navigator-101.html
- For a GPS data logger: **https://github.com/javarobots/GpsDataLogger**
- Many more here: **www.maps-gps-info.com/fgpfw.html#Windows**

Counter connections

As with the simpler *GPS 1pps Timebase* unit, only three connections have to be run to the *12-Digit Frequency Counter*. These can all be made via a shielded stereo cable fitted with a 5-pin DIN plug which plugs into CON1 of the *Deluxe GPS Timebase*.

Fig.6 shows the wiring details. One of the inner conductors of the stereo cable connects to pin 3 of the 5-pin DIN plug, to carry the 1pps output pulses, while the other inner conductor connects to pin 1 of the DIN plug, to carry the +5V supply rail for the timebase. The shield braids are both connected to pin 2 of the plug, to link the two grounds.

At the other end of this cable, the 1pps signal lead and its shield braid should be fitted with a BNC plug, to connect to the counter's external timebase input (CON3). The +5V/GND power lead can either be connected to a 5V DC plugpack or fitted with a 2.5mm concentric DC plug, which mates with a matching DC power socket added to the rear of the frequency counter.

In the latter case, you will also have to connect the +5V and ground lines inside the counter to the added DC power socket – see Fig.6. Make sure that LK3 on the timebase PCB is in the CTR (righthand) position if you are powering the timebase section (ie, the GPS module and IC1) from the counter or an external plugpack.

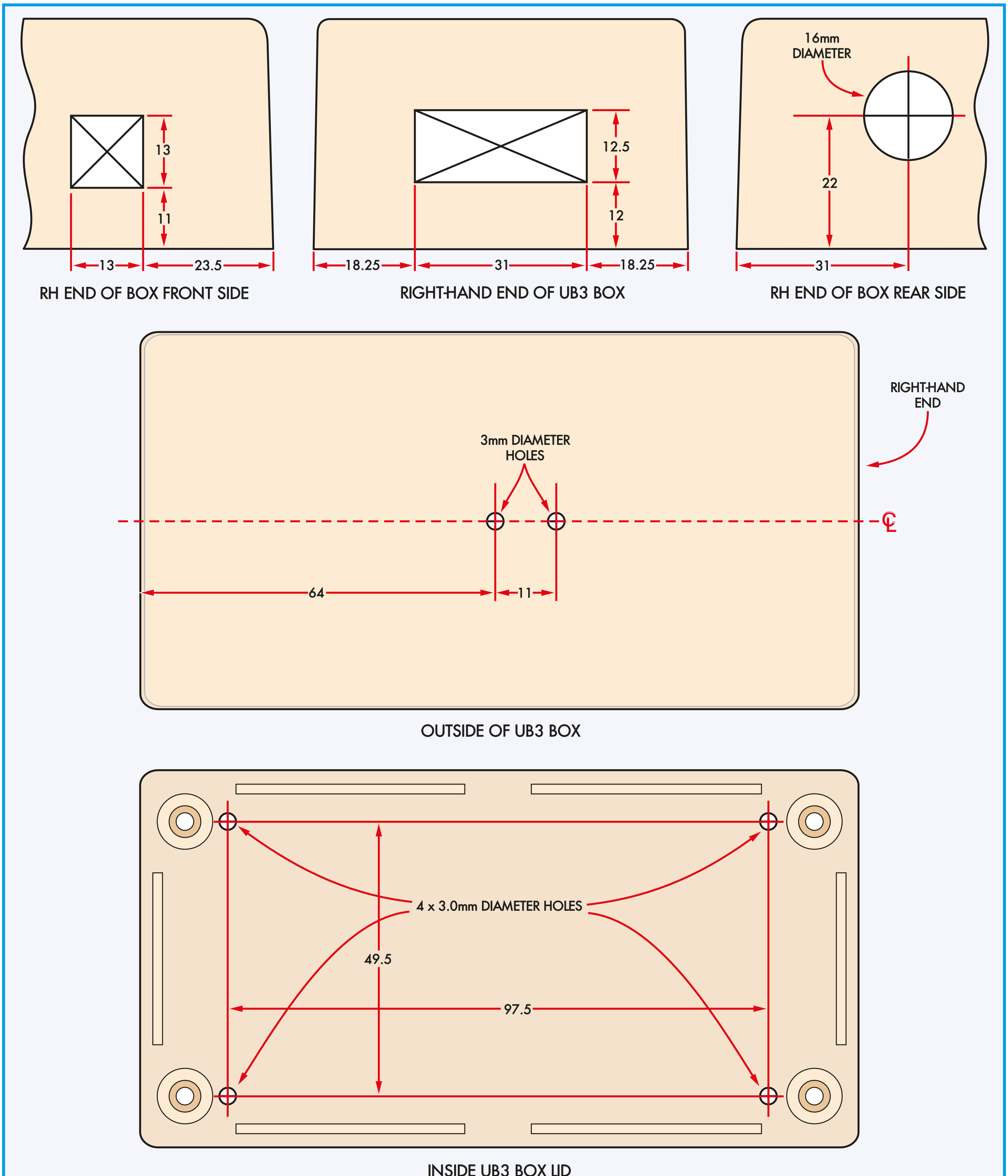


Fig.5: the drilling details for the UB3 jiffy box. The rectangular cutouts can each be made by drilling a series of small holes around the inside perimeter, then knocking out the centre piece and filing to shape.

Alternatively, if you intend running the entire unit exclusively from USB power, then you don't need to install this separate supply cable. Instead, it's simply a matter of connecting the *Deluxe GPS 1pps Timebase* to a USB port on a PC (or a downstream USB hub) using a standard USB cable. Don't forget to set LK3 to the USB position if powering the entire unit from a USB port.

Configuration

When you first connect the unit to a PC, Windows will respond by installing its standard 'virtual COM port' driver. Once it's done that, launch the

Device Manager (eg, via Control Panel) and look under 'Printers and Devices' to make sure that you now have a 'USB serial port'. You can then also check its Properties to discover the COM port number and check that it's working properly. You can also set the driver's baud rate to match the GPS module's rate, which is usually 4800bps.

Assuming this checks out so far, the next step is to download and install Microchip's custom MCP2200 Configuration Utility, available from: **ww1.microchip.com/downloads/en/DeviceDoc/MCP2200_Configuration_Utility_v1.3.zip** (5.13MB). Unzipping this provides a self-installing version of the MCP2200 Configuration Utility.

When you run this and then fire it up, you should see a dialog window as shown in Fig.7 – although you won't see any text as yet in the 'Output' box. This box will be blank initially, while some of the smaller boxes will have different contents.

Before clicking on the 'Configure' button at lower left, you'll need to ensure that the contents of all the smaller boxes are as shown in Fig.7. You probably won't need to change the contents of the Manufacturer, Product, Vendor ID or Product ID boxes, nor will you need to click on the 'Update VID/PID' button. However, you may need to click on the check box next to 'Enable TX/RX LEDs', to display the tick as shown.

Similarly you may need to click on the check box next to 'Enable Suspend Pin', to display its tick.

If the 'Baud Rate' box is not showing '4800', click on the down arrow to its right and then select '4800' from the drop-down list. Then, if the 'I/O Config' box is showing something other than '00000000', click inside the box so that you can type in the correct '00000000' text string.

Similarly, if the 'Output Default' box is not showing '11111111', enter in that text string yourself.

Now turn to the 'LED Function' section at lower right and click on the 'Blink LEDs' radio button if this isn't already selected (ie, displaying the central bullet). Similarly, click on the '200ms' radio button so that it too is selected.

At this stage, you should be seeing a display very much like that shown in Fig.7, except that the 'Output' window should be blank. If so, you can now click on the 'Configure' button at lower

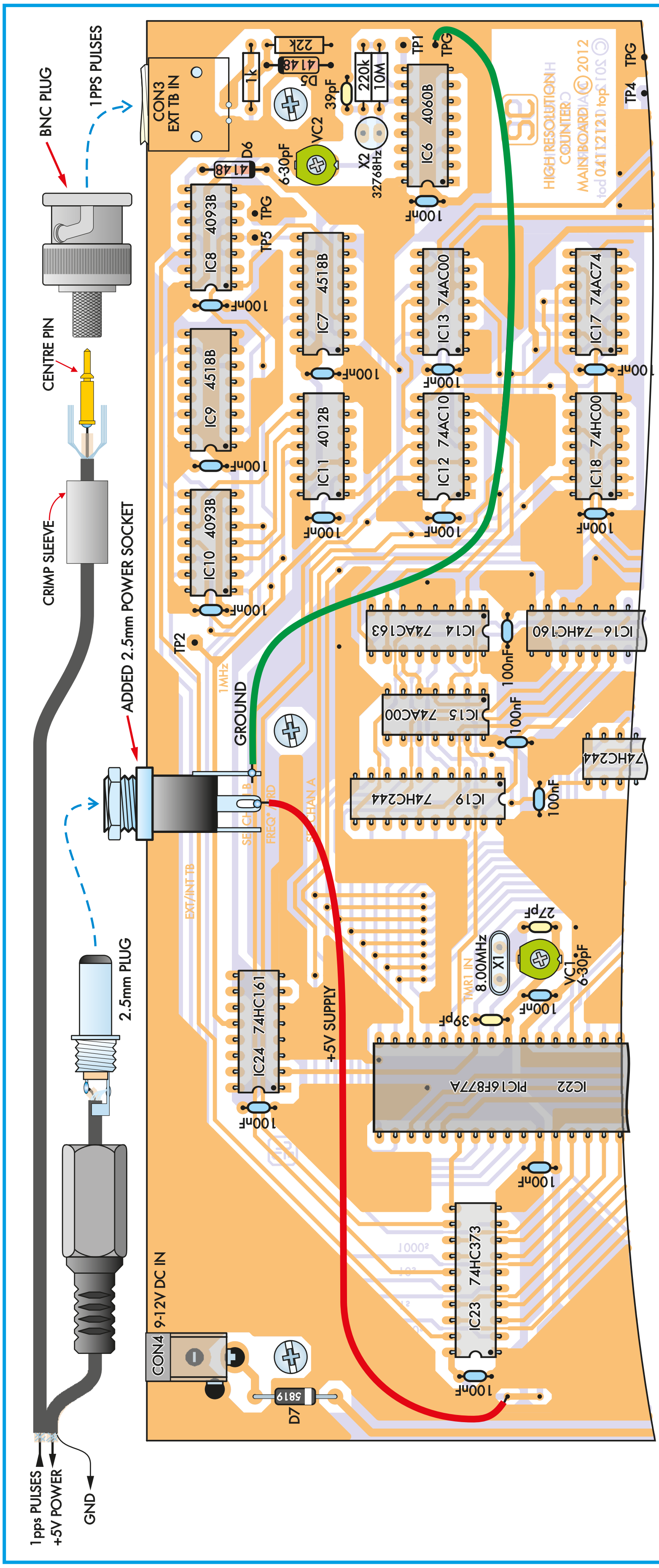


Fig.6: a shielded stereo cable can be used to make the connections between the *Deluxe 1pps GPS Timebase* and the frequency counter. The 1pps pulses are fed in via the counter's existing BNC socket on the rear panel, while a 2.5mm DC power socket can be added to accept a matching plug to pick up the counter's +5V and GND connections. You can omit this DC socket and the supply connections if you don't intend powering the timebase unit from the counter.

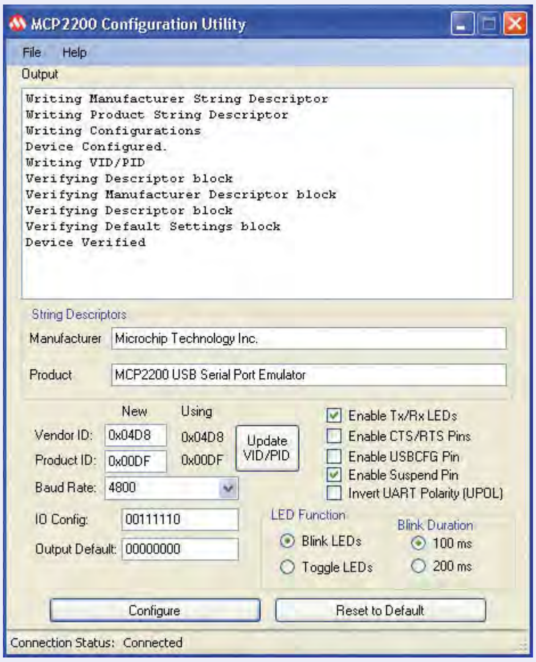


Fig.7: this is the dialog you will see when you launch Microchip's MCP2200 Configuration Utility (except that the Output box will be blank). Configure it as described in the text.

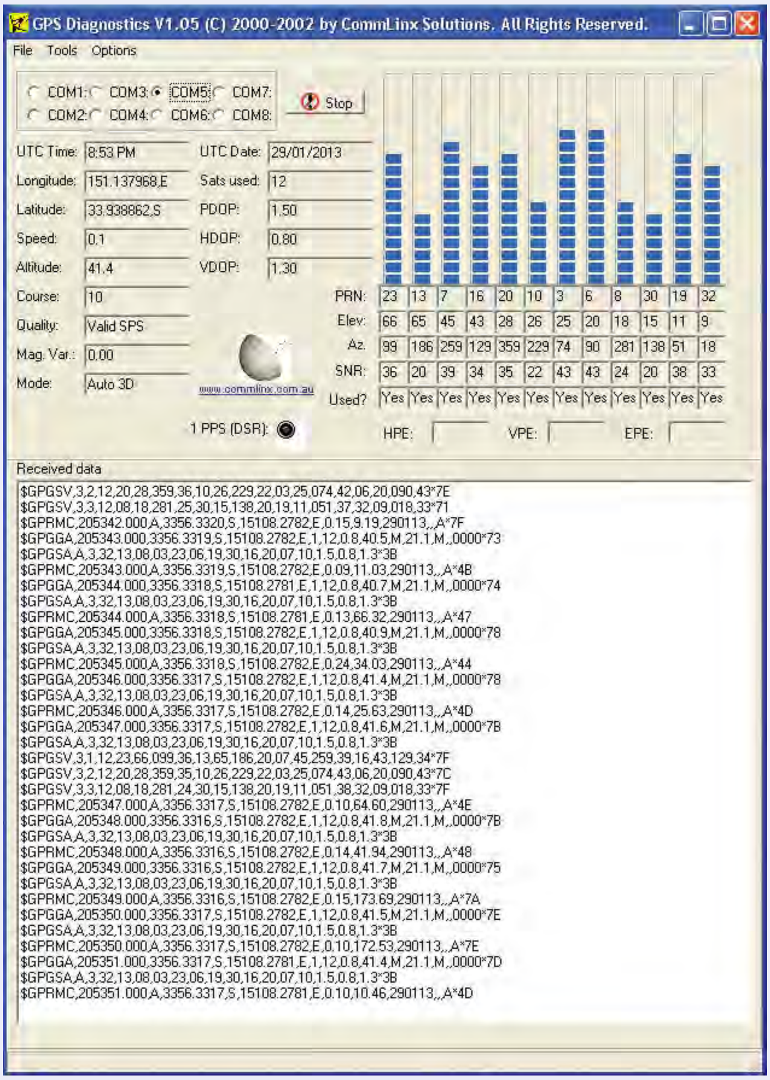


Fig.8: the GPS Diagnostics dialog displays a range of information from the analysed NMEA data, including UTC time, longitude, latitude, altitude, the number of satellites in 'view' and the signal strength from each one.

left. There should then be a brief pause while the configuration utility 'does its thing' with the MCP2200 chip in your *Deluxe GPS 1pps Timebase*, then the text shown in Fig.7 should appear in the 'Output' window. This indicates that the configuration routine has been completed and that the unit is now communicating with the the PC via the USB cable.

Once it's done that, you can then close the Configuration Utility.

Installing the PC software

The final step is to install a software application to allow your PC to analyse and display the useful information carried in the NMEA 0183 output data stream. There are many software apps capable of doing this, but one that we particularly recommend is called 'GPS Diagnostics V1.05'. Developed by CommLinx Solutions, this freeware program can be downloaded from **download.cnet.com/windows**

The quickest way to get to the download page is to search for it by typing its full name in the search box at top right.

Downloading the software is a two-step process. First, you have to download the customised installer program **cbsidlm-tr1_10a-GPSDiag-ORG-10055902.exe** (620kB). You then run this installer to download and install the GPS Diagnostics program itself.

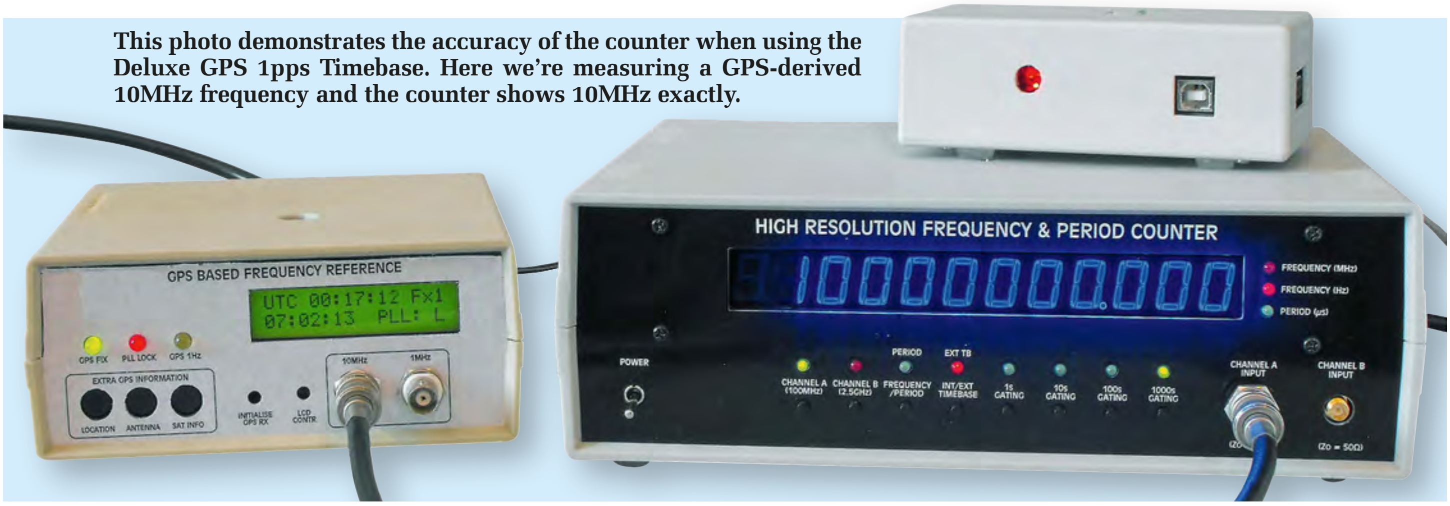
Once it's installed, launch the program to bring up a dialog window much like that shown in Fig.8. The only differences are that all of the text boxes and bargraphs will initially be blank – including the large box at the bottom labelled 'Received data'.

Earlier, when you first plugged the USB cable from the GPS Time Receiver into the PC's USB port, Windows installed it as a USB Serial COM port. The allocated port number could then be determined by going to Device Manager and checking under Ports (COM and LPT). Usually, this will be COM3, COM4 or COM5.

Once you've determined the allocated port number, the next step is to select the corresponding port number in the GPS Diagnostics window. That's done by selecting the appropriate radio button at upper left. This tells the program which COM port the incoming NMEA 0183 data stream from the *Deluxe GPS 1pps Timebase* will be on (in our case, it's COM5).

Analysing NMEA data

You should now find that GPS Diagnostics starts displaying all the information coming into the PC via that COM port. You'll see the NMEA sentences as they arrive in the large Received Data window at the bottom and within a few seconds, you'll also see the UTC time and date, the longitude and latitude, the altitude of your GPS receiver module and a great deal of other interesting information (see upper left of Fig.8).



This photo demonstrates the accuracy of the counter when using the Deluxe GPS 1pps Timebase. Here we're measuring a GPS-derived 10MHz frequency and the counter shows 10MHz exactly.

It will also show the number of GPS satellites currently in 'view', plus a bar chart for each one indicating the approximate signal strength. Under each chart, you'll also see its PRN number, its current elevation and azimuth, its signal-to-noise ratio (SNR) and whether or not it's currently being used. For example, when the screen grab of Fig.8 was captured, our prototype *Deluxe GPS 1pps Timebase* was able to view and use the signals from no fewer than 12 satellites.

That's a bit unusual though. Most of the time, it will use anywhere between five and nine satellites, while at odd times there may be only three or four in view and usable.

So how do you verify that the unit has a good 'fix' and is delivering usable GPS-locked 1pps pulses to your *12-Digit Frequency Counter*? That's done in GPS Diagnostics by examining the 'Mode' message box. This shows 'Auto 3D' in Fig.8, which means that it was able to achieve the highest level of fix when this screen grab was captured.

When you get this message, you can be satisfied that your counter is getting the best possible 1pps pulses.

When the GPS receiver is able to see only a small number of satellites (eg, two or three), the Mode box display can drop back to 'Manual 2D'. This still indicates that the receiver has achieved a 'fix', although some of the navigation information won't be of high quality. However, the 1pps pulses being fed to the counter should still be OK.

It's only time to worry if the Mode message box is blank or showing 'No fix', since that indicates that the unit will probably not be delivering any 1pps pulses at all. If that happens, the trick is to try moving the unit to a location where it can 'view' more of the sky and therefore 'see' more satellites so that it can get a good fix.

In short, GPS Diagnostics is an excellent tool for optimising the position of your *Deluxe GPS 1pps Timebase*. It also allows you to then monitor the reception conditions on a day-to-day basis.

Reproduced by arrangement
with SILICON CHIP
magazine 2014.
www.siliconchip.com.au

Frequency counter measurement accuracy

IN THE FIRST article describing our 12-Digit Frequency Counter (*EPE*, January 2013), we advised readers that by using a GPS-based external 1pps timebase, it should be possible to achieve measurement accuracy approaching that of an atomic clock. In the specifications panel, we also quoted measurement accuracy with a GPS 1Hz timebase of approximately ±1 part in 10^{11}.

Subsequent testing has quantified the accuracy that can be achieved. Over the last three months, Jim has made measurements using the set-up shown above, with the 12-Digit Frequency Counter fed with an external timebase (using the simpler March 2014 unit for the first five weeks and the deluxe unit described here for the remaining seven weeks). The counter was measuring the 10MHz output from our GPS-based Frequency Reference and was set for a gating time of 1000 seconds, so that each measurement took 16.66 minutes. This was done to provide the highest measurement resolution.

The results from this extended testing are: the GPS-locked 10MHz signal from the 10MHz Frequency Reference gave readings of 10,000,000.000 ± 0.003Hz – with a roughly Gaussian or 'bell shaped' distribution centred on 10,000,000.000Hz. In other words, a measurement accuracy of ±3 parts in 10^{10} can be achieved.

Note that with this measurement set-up there are three potential sources of measurement jitter:

1) The GPS module in the 1pps timebase(s)

2) The GPS module in the GPS-Based 10MHz Frequency Reference

3) The inevitable jitter in the PLL (phase-locked loop) inside the 10MHz Frequency Reference itself (used to lock the 10MHz output to the GPS 1pps pulses).

Clearly it isn't easy to separate these three sources of jitter, but with all three present they still allowed us to achieve a measurement accuracy of ±3 parts in 10^{10}. So the true measurement accuracy of the 12-digit frequency counter with the GPS 1Hz timebase is somewhere between ±3 parts in 10^{10} and ±1 part in 10^{11} – still very impressive.

Unless you are measuring an atomic frequency reference, your measurement accuracy is like to be far in excess of the drift and jitter of any source that is commonly available.

A Capacitor Discharge Unit for twin-coil points motors

by Jeff Monegal

Got a model railway? If it is not just a simple loop of track it is bound to have one, two or maybe dozens of sets of points. That means you need at least one Capacitor Discharge Unit (CDU) to power them. Most layouts can make do with just one CDU, but this unit is so cheap you might want to have several.

This Capacitor Discharge Unit, (CDU), is designed to drive twin-coil, snap-action points motors that are widely used on the majority of model railway layouts. These have the virtue of being cheap and easy to install under points.

In action, if one coil (more correctly a solenoid) is energised, the points move across to favour one direction for the on-coming loco. If the other coil is energised, the points move across in the other direction.

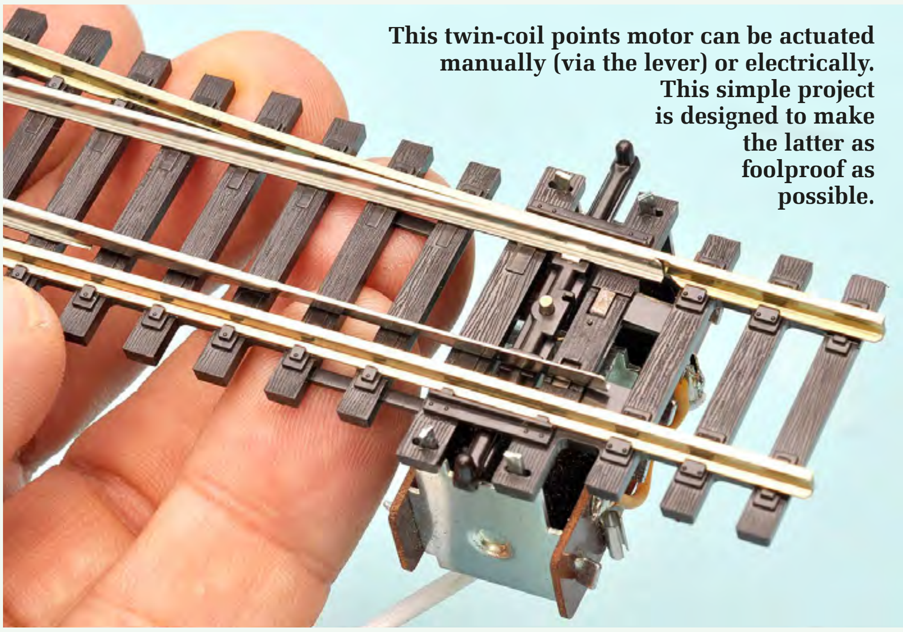
Many rail enthusiasts energise these point motors by simply connecting the two coils to a 15V (or thereabouts) DC or AC supply via momentary contact pushbuttons. Briefly pushing one or other of the buttons operates the points – simple.

The big disadvantage of that method is that if you press the button for too long or the button becomes jammed by something or someone leaning on it, then the respective coil will burn out.

Why? Because its resistance is only about 4.7Ω and it is wound with many turns of fine wire which simply cannot withstand the resultant dissipation of 40W or more for more than a second or two.

This is where the CDU comes in. It has a large capacitor which is charged from the 15V supply and then when one or other of the pushbuttons is pressed to energise one of the coils, it delivers a brief pulse to do the job and no damage can result if the pushbutton is pressed for longer than need be.

This CDU is being presented as a companion unit to the *Automatic Points Controller* in last month's issue, but it can be used independently on any model layout where points are being employed. The CDU is housed on a small PCB which can be located in a convenient position underneath the layout.



This twin-coil points motor can be actuated manually (via the lever) or electrically. This simple project is designed to make the latter as foolproof as possible.

The circuit

The circuit is shown in Fig.1. It consists of a small NPN power transistor, two 2200µF 25V capacitors and not much else. It works like this. Whenever the circuit is connected to the 15V supply (which may be DC or AC) current flows via diode D1 to the collector of Q1, an NPN transistor. Q1 is biased on by the 1kΩ resistor between its base and collector.

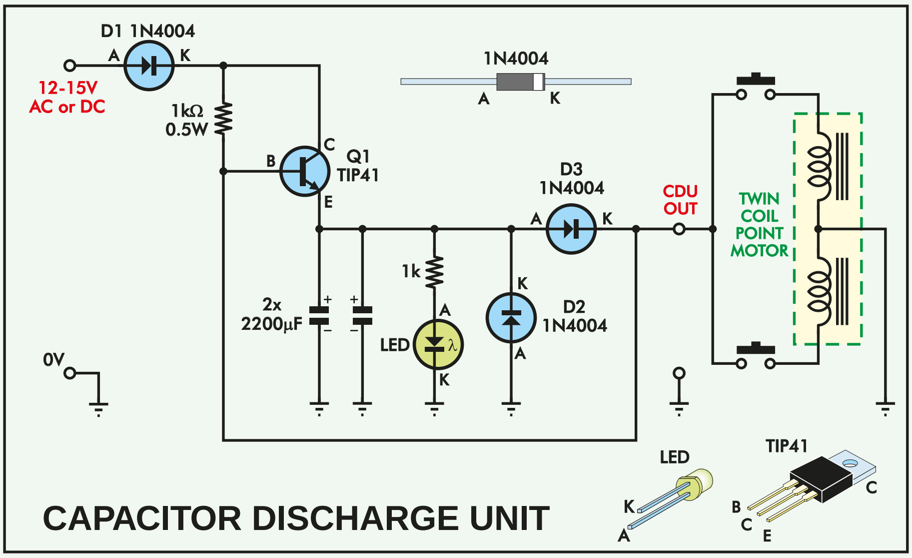
While Q1 is turned on, it charges the two 2200µF capacitors. Once they are charged, the current through Q1 is quite low, due to the leakage of the capacitors themselves and the current through LED1, which indicates that the unit is active.

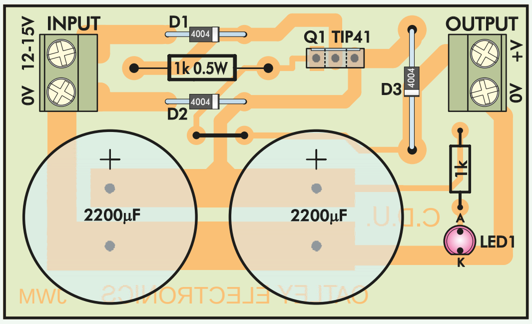
When one of the pushbuttons is pressed, the capacitor charge is dumped via diode D3 to its respective solenoid coil, energising the points motor in one direction or the other. D3 can easily withstand the brief pulse of current which is likely to be no more than 3A peak.

Diodes D2 and D3 act to suppress any back-EMF spikes which might occur if the pushbuttons have contact bounce. Normally of course, the pulse current will die away quickly while you hold the button down for a second or two and no back EMF spike should be generated

If the pushbutton stays depressed for longer, no damage can result since the base of Q1 is effectively grounded via the respective solenoid coil, keeping Q1 turned off.

Once the pushbutton is released, Q1 is biassed on again via the 1kΩ base





CAPACITOR DISCHARGE POINT MOTOR DRIVER

Fig.1 (left): the circuit diagram of the capacitor discharge unit shows it is basically a couple of capacitors and a switching transistor. Above (Fig.2) is the PCB component overlay. It's simple enough – but watch component polarity.

resistor and the capacitors are quickly recharged, ready for the next points operation.

Note that this CDU can power multiple sets of points. Each twin-coil points motor is wired to the CDU via a 3-way ribbon cable and two pushbuttons.

PCB assembly

The CDU circuit components fit on a small PCB measuring 69mm × 41mm, coded 09203131. This PCB is available from the *EPE PCB Service*. Assembly is straightforward, but remember that all components, except the two resistors, are polarised and must be installed as shown on the overlay diagram in Fig.2.

After double checking that you have all components in the correct position and the correct way round you can apply a DC power supply of around 12-15V DC or AC to the power in terminals. The project is polarity-protected by diode D1, so if you connect the supply the wrong way nothing will happen. But if all is well, the LED will come on shortly after power is connected.

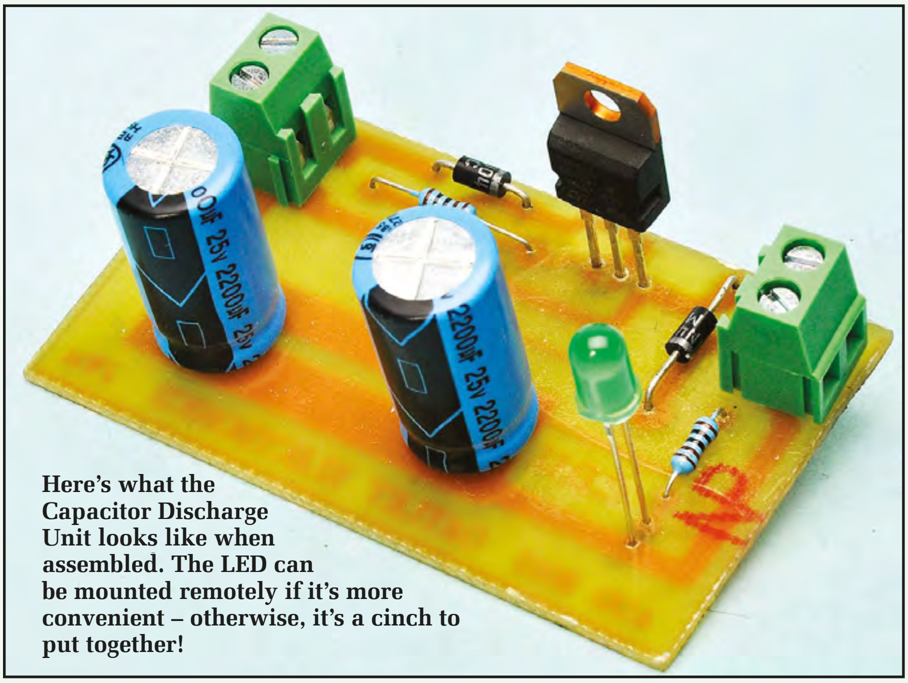
Using a twin-coil snap-action points motor and some hookup wire, join the centre terminal of the points motor to either output terminal.

Using another length of hookup wire with one end connected to the other output terminal, touch the free end to either of the other two terminals of the points motor. The motor should snap in one direction or the other.

At the same time, the LED should go out but then come back on within a few seconds.

Try again with the other points motor terminal, but this time leave the hookup wire connected. There should be very little (a few mA) load on the power supply.

Since the transistor is held off while the points motor is connected across the output, no current should flow. When the hookup wire is removed, current should briefly flow again to charge up the capacitors, ready for the next application.



Here's what the Capacitor Discharge Unit looks like when assembled. The LED can be mounted remotely if it's more convenient – otherwise, it's a cinch to put together!

Parts List – Model Railway Capacitor Discharge Unit

- 1 PCB available from the *EPE PCB Service,* measuring 69mm × 41mm, coded 09203131
- 3 1N4004 power diodes
- 1 TIP41 NPN power transistor
- 1 5mm LED (any colour)
- 2 PCB-mount 2-way connectors
- 1 1kΩ 1/2W carbon resistor
- 1 1kΩ 1/4W carbon resistor
- 2 2200µF 25V elect. capacitors

A complete kit of parts will be available from Oatley Electronics in Sydney. They can be contacted via their web site www.oatleyelectronics.com or by phoning (02) 9586 3564. Kit cost is $10 plus $7.00 P&P. All project enquires should be sent to the designer, Jeff Monegal. He can be contacted via email only. (jeffmon@optusnet.com.au) All emails will be replied to but please allow up to 48 hours for a reply.

Beta-Layout's Re-flow Oven Kit and Controller

Reviewed by Mike Hibbett

FOR hobbyists, the technique of soldering has changed little in the last 20 years. We use Veroboard, hand-etched boards and occasionally purchase a few professionally manufactured printed circuit boards when we want to treat ourselves. The more adventurous hobbyist might dabble with double-sided boards, and perhaps the odd surface-mount component. In the main though, designs use 'through hole' components, especially for low frequency microcontroller and audio projects.

With the advent of the Internet in the 1990s, niche suppliers such as board manufacturers and component suppliers were able to expand from their normal professional markets and connect with hobbyists around the world at very little additional cost. 20 years ago, few hobbyists would have known of the existence of 'solder paste'. Today, anyone can buy it on Amazon!

Surface-mount components

With increasing miniaturisation of consumer electronic devices and the huge numbers of products sold, tiny surface-mount devices are becoming common, cheap, and readily available. Components that feed the mobile phone market, such as GSM transceiver modules, digital compasses and 3-axis accelerometers are within the price range of most hobbyists – but in packages that are nearly impossible to hand solder.

Hobbyists are also becoming suppliers, selling custom-made printed circuit boards to other hobbyists through Internet channels such as eBay, Amazon and Kickstarter. With even modest production runs, hand assembly of boards can be very time consuming and quality difficult to maintain.

Solder reflow techniques

Professional circuit board assembly of surface-mount components is typically done with a re-flow oven, which heats the board (or a panel of boards) to a temperature where solder paste on the component pads melts, forming a joint between the component and the pad. Unlike hand soldering, where the solder is supplied in wire form, for automated re-flow soldering the solder is applied as a paste to the board (only on the exposed pads, or lands, as they are called) prior to the components being placed onto the board. The paste first acts like a weak glue, holding the components in place until the oven melts the paste and forms a normal solder joint.

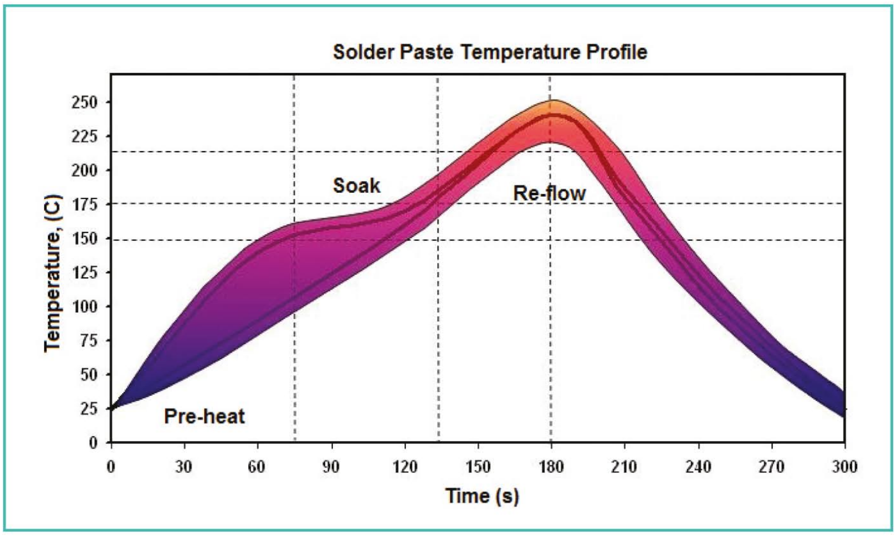


Fig.1. Solder paste temperature profile

There are two critical aspects to this process: the control of the heat within the oven (bringing the board to temperature without damaging the components, but ensuring the solder makes a good joint) and the design of the solder paste (which consists of tiny balls of solder held within a powerful flux.) If the oven heats too high, components will be damaged. Too quickly, and the flux may not have time to fully melt and act on the surfaces to be joined, resulting in a dry joint. The temperature profile for a typical solder paste is shown in Fig.1.

A significant amount of research has gone into this subject, resulting in machines capable of soldering millions of components without fault. And we lucky hobbyists can benefit from this, if we wish.

Professional re-flow ovens are truck-sized machines costing hundreds of thousands of pounds, but fortunately the solder paste is not so expensive, costing around £25 for 100g. As a small board only needs about 0.5g, that works out at about 12-pence-worth of solder per board.

Although solder paste can be applied to a board by hand using a syringe, a much simpler technique if you are getting your boards made professionally is to order a solder stencil with your boards. This is a thin sheet of stainless steel slightly larger than your board (or panel of boards) with precise holes cut to match your PCB design. The holes correspond with where solder paste should appear. The solder paste can be applied in a technique similar to silk screen printing, as we will see later. These stencils are relatively inexpensive, and are sometimes supplied free of charge when you purchase prototype PCBs.

The next issue to solve is the heating of the board. Some people have found success using a 'skillet' frying pan on a hob, but this can be hit and miss. Fortunately, the low cost domestic desktop oven (the type loved by students and bed-sit dwellers) has been found to be suitable, and virtually all are capable of reaching the temperatures required (240ºC). The only issue remaining is how to control the temperature, and this is where Beta-Layout comes in.

Temperature control

Beta-Layout is probably better known to hobbyists as PCB-Pool, the supplier of low volume and prototype PCBs. It has recognised the growing demand for low-cost surface-mount assembly solutions, and has started supplying a re-flow oven kit, controller module and various consumables such as solder paste. It's unsurprising to hear that PCB-Pool supplies a free stainless steel stencil with every prototype PCB order, helping to make the use of a re-flow oven an attractive option!

We will look at two products from Beta-Layout – the Re-flow Oven Kit and the Oven Controller module. Beta-Layout sells the solution as two kits for flexibility; not everyone will want to buy an oven from Beta-Layout, and on their website there are further options for buying solder paste, applicators and PCB frames.

Fig.2. The Re-flow Oven Kit contents

Re-flow Oven Kit

This is the basic kit, shown in Fig.2. It provides the bare minimum of parts required. There is nothing special about the convection oven, and if you have one already or are going to source one of your own choice, you can save 50 euros and buy the remaining parts as a 'Small Re-flow Kit'. Bear in mind that you really should not share the use of an oven used for cooking, as the flux fumes given off during soldering are toxic. And nobody wants to find an IC or capacitor in their Sunday roast. Likewise, the solder paste needs to be stored at a cool temperature (4 to 10°C) but keeping it in a domestic refrigerator can be dangerous, if some unsuspecting visitor thinks the solder paste is edible. Keep it somewhere else. We bought a small beer fridge for the task. Don't leave it somewhere that it might freeze either; the consistency of the mixture is very important, and will not like ice crystals forming on it.

The oven has the normal controls (timer, temperature, number of elements) which, when in use with the controller, are simply turned to maximum and left alone. Being a basic model, the timer is mechanical and gives an annoying tick as it counts down, and cannot be manually turned to zero. We quickly got the hang of setting all the controls to maximum – except the timer – which we set to 10 minutes. That is more than enough time to solder a board, and if not, you can just add a few minutes.

A cheap multimeter is provided with a thermocouple, so that in the absence of the controller you can manually monitor the oven temperature and adjust the temperature control to follow the temperature profile required by the solder paste. A word of warning – this is not easy, and although the soldering process only takes a few minutes, manual control is very hit and miss and a somewhat stressful activity. You really do need the controller module to accurately and automatically control the temperature.

Bits and pieces

A 100g tub of solder paste is provided, which itself would cost about £25 from some suppliers. The characteristics of solder paste are very different to the wire solder we hobbyists are used to, and deserves further attention. We will pick up on this shortly. The user manual shows how to apply the paste; it's quite easy. The paste sticks cleanly to the PCB and when the stencil is removed you can see the neat towers of solder paste. If you have made a bad job or accidentally smeared the paste you can wash it off with water and start again. Despite having never done this before we had perfect results the first time, which is probably more down to the careful design of the solder paste than skill on our part.

A small sheet of thin stainless steel is provided, which is the tool for applying the solder paste by smearing the paste across the board. It's a very simple tool, but also very effective. At about six inches in width, it will cope with most boards, but if you have a larger board (or, more likely, a panel of them) then you simply sweep the sheet around the board stencil dragging solder paste as you go.

A pair of plastic-tipped metal tweezers are provided for picking up and placing the surface-mount components. The plastic tip is helpful because it provides additional grip over a cheaper all-metal pair. A tip: if you plan to make a board, do it before having a coffee!

What look like scraps of 1.6mm and 1.0mm PCB are actually important components; these acts as guides that hold your PCB in position while you apply solder paste. A kind of simple 'jig', made from scrap PCB material. An A4-sized sheet of fibreboard is provided to act as a base for everything. Even masking tape is supplied, to secure the PCB jig, your PCB and the stencil in place. We will see how this all works shortly.

The remaining parts of the kit are two test printed circuit boards, a stencil for them and the components to fit onto the boards. The boards don't do anything; the components are not connected, but they give you the opportunity to try the techniques out, twice, before attempting your own board. This is a welcome addition; we, like many hobbyists, had never seen a solder stencil before, so it was good to even handle the thing and become familiar with it before trying to design and order a real one. Last, but not least, a CD is supplied with the user manual and a video tutorial demonstrating the machine. The video is available for download from their website.

It's a very comprehensive kit, and we only needed to supplement it with a magnifying eyeglass to check the distribution of the solder deposited and some Kapton tape to hold the temperature sensor on the board (more on that later.)

The instructions for the re-flow oven appear to indicate that you can use the oven without an automatic controller, but doing so is a bit too hit and miss for our liking. If your boards are simple, with a few components that are not too sensitive to excessive temperatures, then you can probably work without a controller. In this case, your initial costs can be quite low; get yourself a second-hand oven from somewhere, buy the 84 euros 'Small Re-flow Kit', and go off and have some fun. If you are interested in repeatable, quality soldering on complex or sensitive boards, then read on.

Controller

The Oven Controller is supplied as a separate item for 130 euros, excluding VAT and delivery charge. What you get for your money is shown in Fig.3. It's a simple, tidy solution. The box is inserted 'in-line' with the mains supply to the oven (an IEC-to-European mains lead adaptor is supplied for this purpose.) The user interface consists of three buttons and six LEDs. Plus a socket for the thermocouple probe, and a 9-pin RS232 interface.

First things first – let's take it apart to find out what makes it tick. There are no screws; the case opens by squeezing the sides of the box at the join between the top and bottom halves. This is rather surprising for a device that handles mains voltages, but we assume fully legal. Inside there are two circuit boards neatly laid out with solid interconnects; this is a bit of kit that is going to last. You can see the contents for yourself in Fig.4 and 5.

The top board, which connects to the LEDs and buttons has a small microcontroller, an ATMEGA32A. There is a TTL-to-RS232 converter IC, which presumably provides a serial comms link to the PC (we haven't looked at the user manual yet to see what that does) and a buzzer for audible feedback – all

Fig.3. The Oven Controller

standard fare. Underneath some wires we also found an IC that interfaces the thermocouple to the processor.

Things get a bit more interesting on the lower board. A small sealed transformer provides the power supply to the microcontroller and the essential isolation for the thermocouple and RS232 interface. There is a fuse, and a single four-pin IC connected to a heatsink. This IC turns out to be a solid-state relay that provides the power control to the oven. It's a neat device, a combination of high power triac and opto-isolated switch to provide 16A current switching capability with 4kV isolation. From the microcontroller's perspective, the device looks like an infrared LED. It's an interesting IC and makes power control rather simple, and we'd love to use it in some future projects. But we digress – how does the controller actually work?

How does it work?

By use of the built in processor, the controller is able to monitor the temperature of the board, and pulse the power to the oven off and on to cause the temperature to rise and fall in line with the requirements of the solder paste we are using. (The controller obviously has been pre-programmed with the 'curve' of the supplied solder paste.)

In theory, all one has to do is place the thermocouple on the PCB, put the board in the oven, press the 'Solder' button and wait a few minutes for the solder to flow and set.

There is, however, a very nice feature of the controller that makes it worth its price – it has a 'Learn' facility, where the physical characteristics of the oven you are using can be determined, and the oven will change its stored calibration data appropriately. There is a lot of work going on under the hood, the controller compensates for the oven 'lag', recognising that the temperature will continue to rise after the power has been turned off. It's a fairy complex algorithm.

The first time the controller is powered on it needs to run this learning algorithm. This was a simple case of securing the thermocouple to a piece of PCB, placing it in the oven and hitting the 'Learn' button. The controller heats the oven to 100˚C, switches the power off and then monitors the further rise in temperature for a minute or two.

The controller is now ready for use – or at least it will be, once the temperature has fallen below 50˚C. This step is necessary to ensure that the PCB goes through a complete temperature cycle from a reasonably cool point. The time you have to wait can be decreased by opening the door, naturally, and this is not an issue in hobbyist situations. The controller, unsurprisingly, enforces this 'non-operational' state until it detects the temperature has fallen sufficiently.

So, we are ready to go. Rather than leap ahead and try soldering a board, however, we did a 'dummy run' while monitoring the change in temperature – both on the multimeter and on the debug output text printed through the serial port. The output is very useful – it prints the current profile 'phase' (such as pre-heat, re-flow) the current time and the temperature.

We quickly noticed an oddity; the controller was displaying a temperature 6˚C higher than the multimeter. This could cause problems during solder re-flow, as the paste would not reach its minimum melting temperature. A quick search on the Internet indicated other had seen this too, but there was a simple fix – a serial port command to adjust (well, offset) the temperature value, just entering:

```
tempoffset -6
```

fixed the problem. It looks as though the controllers are shipped from the factory in a default state, and this is a second calibration routine that needs to be performed. No problem, but you will need a serial connection to a PC.

Buried within the controller is a large number of control variables, all of which may be accessed via the serial port. The user has complete control over the shape of the temperature profile, which will be handy in the future when dealing with different solder pastes or complex PCBs. For now, however, having completed the calibration of the

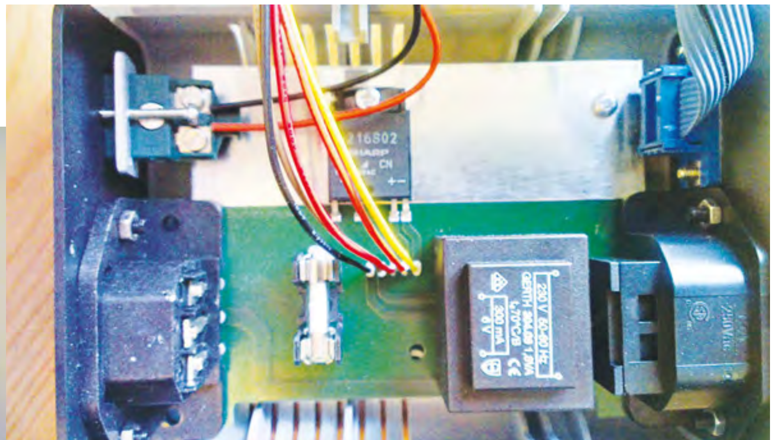


Fig.5. Inside the Oven Controller – power control board

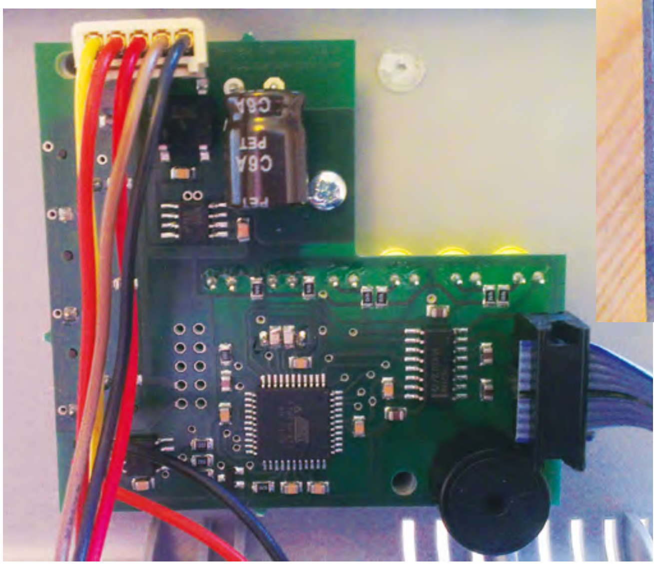


Fig.4. Inside the Oven Controller – the CPU board

Fig.6. Preparing the PCB 'jig'

controller, the serial port can be disconnected and the controller used standalone. Calibration constants are held in non-volatile memory.

The LEDs on the controller show the progress of the soldering operation, but as this is quite quick (taking only a few minutes) it's not necessary to monitor progress – simply press the 'Solder' button and wait for the buzzer to indicate that it has finished. At this point, open the oven door and wait for the board to cool, which takes about five minutes.

Solder paste

As you will have guessed, the solder paste is the important ingredient in surface-mount soldering. The paste has the consistency of fine peanut butter, and consists of tiny balls of solder about 0.05mm in diameter mixed with a viscous but fluid flux. The flux both cleans the two surfaces to be soldered and provides a weak adhesive to hold components in place prior to re-flowing.

The viscosity of the solder is very important – too 'runny' and it wont stay on the pads, causing shorts between pins. Too thick and it wont stick to the PCB. To maintain the ideal viscosity, the solder paste must be kept in a sealed container at a controlled temperature (between 4 and 10˚C) and it must be allowed to slowly warm to room temperature for a few hours before use, and should be stirred well immediately before use. Also, it has a limited shelf life of six months under the conditions mentioned above (although the reduction in performance may not be noticed for another six months or so with hobbyist PCB designs.)

Given that it has a relatively short shelf life, it should be purchased in small quantities. The 100g tub sold by Beta-

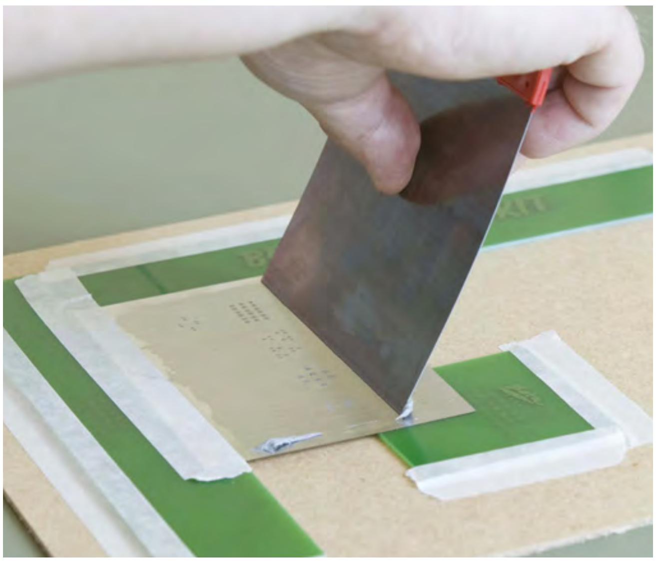


Fig.8. Applying the solder

Fig.7. Adding solder to the stencil

Layout costs about 20 euros and will be enough for roughly 200 small PCBs.

In use

A very clear video tutorial is supplied with the oven, but the process is just four steps: mount the board in the jig, apply paste, spread the paste and place the components.

The placing of the PCB in the jig is shown in Fig.10. Note that the PCB material used for the jig must be of the same thickness as your PCB, otherwise the stencil will not sit flat. There is nothing special in the design of the jig, it's simply a frame to hold your PCB still. The stencil is carefully aligned with the PCB and stuck to the jig, then small blobs of solder paste are applied at various points around the board, as shown in Fig. 7. Don't worry about putting too much solder down, you will be able to scrape any unused solder off at the end and put it back in the tub.

Now comes the critical stage: using the steel application sheet to spread the solder over the stencil, as shown in Fig.8. It took us two attempts to get this right, but the results, as shown in Fig.9, were perfect. Note how the solder paste forms a small 'tower' – this is exactly how it should look.

Finally, place the components. The kit supplies five different component types, so you can practise picking them up and placing them without smudging the solder paste. We deliberately placed a few of the components 'badly', to demonstrate another benefit of using an oven – as the solder melts, the components should re-align to the centre of the pads. You can see our efforts in Fig.10 and Fig.11.

We deviated from the instructions at this point, connecting the thermocouple to our board directly, using a small square of Kapton tape. The instructions recommend that you attach the thermocouple to a separate PCB with a loop of wire, but Kapton tape seems more effective. The temperature sensor can be attached closer to the parts that will

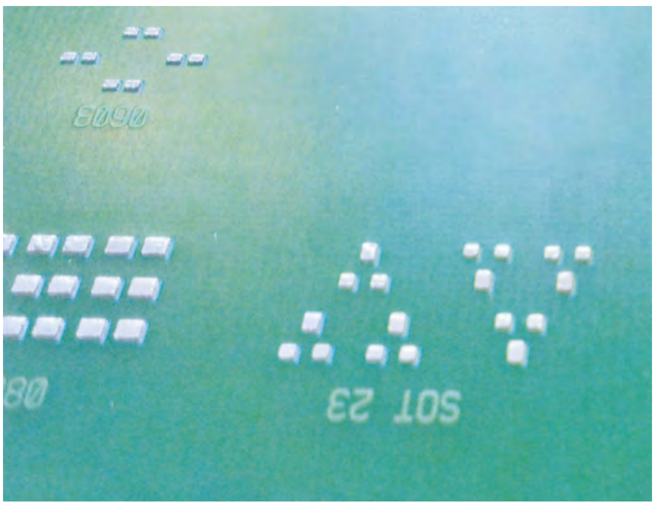


Fig.9. Solder paste on the PCB

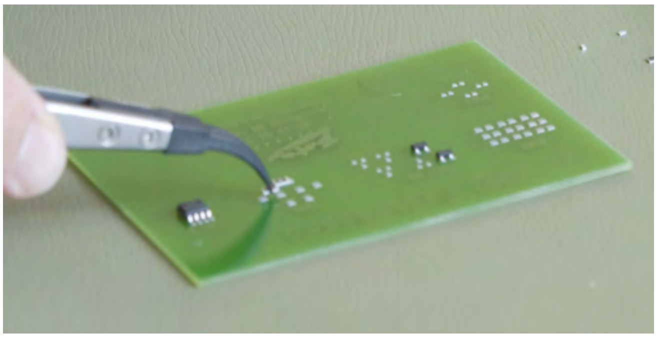


Fig.10. Placing the components

actually be soldered. A £4 roll of Kapton tape will be good for about 3000 boards.

Watching through the oven window with the aid of a torch, we could see the solder change suddenly from a matt paste to a shiny, smooth surface. The components visibly re-align on the pads, correcting any minor placement errors.

The finished result can be seen in Fig.12. As you can see, our badly aligned components have rotated round into the desired orientation. It isn't particularly clear from the photograph, but the quality of the solder finish was excellent.

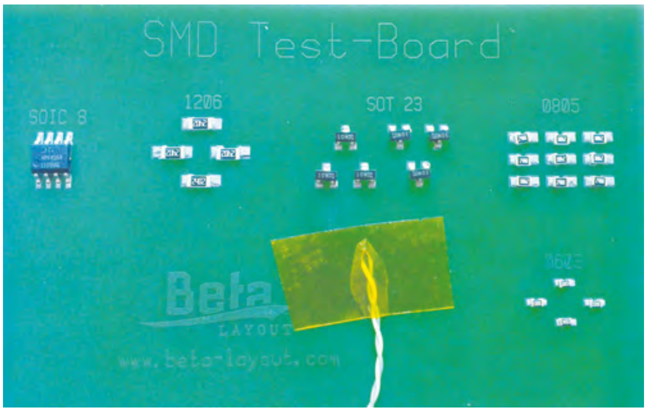


Fig.11. The board before re-flow (note the badly positioned components)

The real proof of how useful the Oven and Controller kit will come when we can demonstrate it being used with a real circuit board, made up as a panel of boards. In the summer, we will take a look at designing a PCB with surface-mount components and walk through the process from entering the design in a CAD system to soldering the board in the oven. And we shall use some small and exotic components.

Until then, we are off to warm something up in the oven!

References

Ref. 1 Solder Paste Datasheet: www.aimsolder.com/sites/default/files/nc254_sac305_solder_paste_rev_17_0.pdf

Ref. 2 Beta-Layout website: www.beta-estore.com/rkuk/index.html

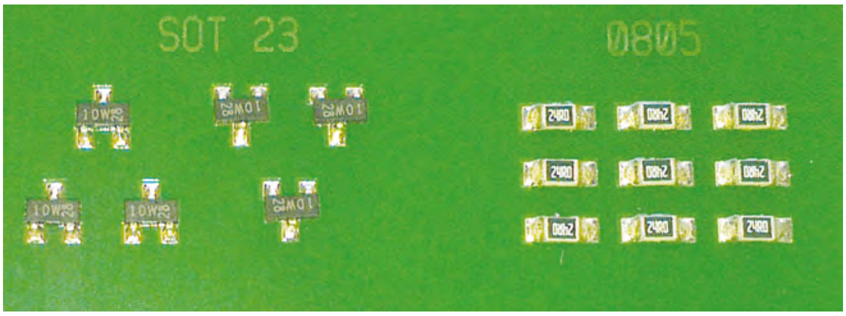


Fig.12. The board after re-flow (note how the components have re-aligned themselves)

Why might I want one?

This isn't a question we ask often when doing a review, but the question is reasonable – why would I want one of these? It's not the most obvious hobbyist accessory.

Building circuits with surface-mount components has not been common practice in the past, but we can expect more components to become available only in surface-mount format. While the days of the DIL package are not numbered, don't expected the latest cool IC, module or memory socket to be available in through-hole technology. For those of us who design and build a large number of boards, an SMA oven would come in very handy. And it would fit in well with the 'hacker spaces' that are springing up in major towns, where groups may make several dozen boards.

So, not for everyone – but for those who do make low-volume production runs of PCBs, it does the job, and does it well.

The oven and controller together costs 285 euros (inclusive of VAT and shipping charges) or 233 euros if ordered without the oven.

Blessed broadband, corrupt copper

TechnoTalk

Mark Nelson

This month's sermon ranges from the sublime to the scandalous – Mark Nelson looks at an inspired solution to rural broadband deserts and warns against cutting corners when it comes to choosing network cable.

EVEN though I don't see myself as particularly lucky, I am nevertheless fortunate to have a choice of superfast broadband providers where I live, In England's largest town that is not a city. Just across the road from me, BT Internet had the foresight to install a cabinet for its Infinity fibre broadband service, while the Virgin Media cabinet just down the road claims to beat BT's speed – but in practice could not manage anything faster more than 5Mbit/s here.

Broadband desert

This kind of talk will merely rub salt in the wounds of people who, through no fault of their own, live in more far-flung areas where not even one operator provides a high-speed broadband service. Parts of eastern England fit that category, and people living there have found a variety of differing solutions to the problem. Suffolk County Council's approach was to subsidise British Telecom to bring speeds of over 24Mbit/s to 85 per cent of premises in the county. This sounds promising until you see the map at: **www.betterbroadbandsuffolk.com/LineCheck.aspx** and observe the large areas of white space where there are no plans to introduce fibre broadband. **JayCee58**, a user in the broadband desert states: 'I'm in an area of Suffolk that appears to have no plans. I have a download speed of 1.2Mbit/s and I only get that by using a router that allows me to adjust the signal-to-noise ratio margin. Previous to that, I was only getting 0.5Mbit/s.'

Holy alliance

In nearby Norfolk, Steve Batson and Pete Freeman had an inspired solution to the dearth of high-speed broadband in their county. Recognising the 'big sky' nature of Norfolk – a relatively flat expanse dotted with churches erected mainly on high locations – they set about forming a holy alliance with the Anglican Diocese of Norfolk to install wireless hotspots on the spires of rural churches. Their high-capacity WiSpire wireless broadband (**www.wispire.co.uk**) offering serves many areas that have no current or planned fibre-optic coverage. At the start of this year, the company had over 30 hotspots already in service, with installation in progress at another 21 sites (check out the coverage map at **http://recombu.com/digital/news/wispire-rural-broadband_M10952.html**). Even better, prices are not significantly higher than what national broadband providers charge in more densely populated areas.

Other counties may follow this lead. A plan has been drawn up by the Rev Dr Julie Nelson, rural officer at the Diocese of Chelmsford, to deliver wireless broadband from church towers in Essex. She told local paper the *Saffron Walden Reporter*: 'The Church is interested in community health and community resilience, and today an essential requirement for any community is broadband – it's the fourth utility. Young families are choosing not to live in areas where broadband service is poor and local businesses are considering moving out – trends that threaten the long-term sustainability of rural communities.' Other parts of Essex already have independent wireless broadband services operated by local firm County Broadband (**www.countybroadband.co.uk/get-broadband/coverage**).

The relatively flat terrain in other parts of East Anglia makes wireless (radio) distribution a practical proposition there too. In Lincolnshire, the county and district councils have paid wireless broadband provider Quickline (**www.quickline.co.uk/coverage**) to provide wireless broadband in around 40 villages (and the towns of Caistor and Market Rasen). Residents there connect to hotspots installed at village halls to get speeds of up to 20Mbit/s.

Sinful deception

The more 'hands-on' type of users (such as readers of this magazine) tend to favour Category 5e (Cat 5e for short) cabling over Wi-Fi for distributing broadband around the home. Cable is more secure and if you lay it carefully in accordance with the guidelines (no sharp bends please!), it often delivers a more robust signal. If your installation fails to satisfy, the cause might be the cable you use and you need to be aware that some Cat 5e products on the market are decidedly dodgy.

'Many of the cables masquerading as Category 5e actually have a cheaper aluminium core hidden beneath an external covering of copper. We know the price of copper, we know how much it should cost to manufacture fully compliant Cat 5e cable and what we found was truly shocking,' warns Neil Mabbott of cable supplier Draka UK. 'A genuine Category 5e cable has a high quality copper conductor. Installers tempted by these low-cost, counterfeit products should bear in mind that networks installed with copper-covered aluminium cables cannot comply with Category 5e standards. More importantly, they could be unsafe.'

Strong words, and if you are getting adequate performance from your network, you have no need to worry. But why is copper-covered aluminium cable bad for you? There are three reasons. Aluminium is a less efficient conductor than copper, so the cable will invariably deliver reduced performance, which means that it will also fail to meet some of the near-end crosstalk or return loss requirements of Category 5e. Aluminium is less malleable than copper, which means it becomes 'work hardened' and extremely brittle. The cable ends break off easily and because the conductor is now too short, you end up installing the entire cable afresh.

More misery

Aluminium also starts to oxidise as soon as it is exposed, which is the case when you use insulation displacement punch-down connectors. This means that there is always an area of the core exposed, another factor that could make re-terminating the connection tricky. BT learnt both of these lessons the hard way when they installed aluminium telephone wires about 40 years ago, when the price of copper rocketed following the Rhodesia crisis and they used cheaper aluminium to save money (a Google search for 'aluminium telephone wires' indicates pages of grief about the problems with aluminium cabling). Many, many miles of aluminium cable have had to be replaced with copper at great cost. The bottom line is that you should stick to all-copper Cat 5e cable and be suspicious of any bargain offers!

EPE EVERYDAY PRACTICAL ELECTRONICS BACK ISSUES

We can supply back issues of *EPE* by post, most issues from the past five years are available. An *EPE* index for the last five years is also available at **www.epemag.com**. Where we are unable to provide a back issue a photocopy of any one article (or one part of a series) can be purchased for the same price. Issues from Jan. 99 are available on CD-ROM or DVD-ROM – and back issues from recent years are also available to download from **www.epemag.com**. Please make sure all components are still available before commencing any project from a back-dated issue.

DID YOU MISS THESE?

BACK ISSUES ONLY £5.00 (each inc. UK p&p.)

Overseas prices £5.80 each surface mail, £6.80 each airmail.

We can also supply issues from years: 2006 (**except** Jan., Feb., Mar., Apr., May, July). 2007 (**except** June, July, Aug), 2008 (**except** Aug, Nov, Dec) 2009 (**except** Jan, Mar, Apr.). 2010 (**except** May, June, July, August, Oct, Nov, Dec) 2011 (**except** Jan).

FEB '13

PROJECTS • Semtest – Part 1 • Crystal DAC • 10W LED Floodlight • Built-In Speakers • Universal USB Data Logger – Part 3

FEATURES • Jump Start – Logic Probe • Techno Talk • PIC N' Mix • Raspberry Pi – Software Investigation • Circuit Surgery • Interface • Max's Cool Beans • Net Work

MAR '13

PROJECTS • Lightning Detector • SemTest – Part 2 • Digital Spirit Level • Interplanetary Voice • Ingenuity Unlimited

FEATURES • Jump Start – DC Motor Controller • Techno Talk • PIC N' Mix • Raspberry Pi – Further Investigation • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work

APR '13

PROJECTS • SoftStarter • 6-Decade Resistance Substitution Box • SemTest – Part 3

FEATURES • Jump Start – Egg Timer • Techno Talk • PIC N' Mix • Circuit Surgery • Interface • Max's Cool Beans • Net Work

MAY '13

PROJECTS • Electronic Stethoscope • PIC/AVR Programming Adaptor Board – Part 1 • Cheap, High-Current Bench Supply • Ingenuity Unlimited

FEATURES • Jump Start – Signal Injector Probe • Techno Talk • Raspberry Pi • PIC N' Mix • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work

JUNE '13

PROJECTS • Mix-It • PIC/AVR Programming Adaptor Board – Part 2 • A Handy USB Breakout Box • Converter For Neon Lamp Experiments • Ingenuity Unlimited

FEATURES • Jump Start – Simple Radio Receiver • Techno Talk • PIC N' Mix • Circuit Surgery • Interface • Max's Cool Beans • Net Work

JULY '13

PROJECTS • 6-Decade Capacitance Substitution Box • Soft Starter For Power Tools • High Power Brushless Motors From Old CD/DVD Drives • High-Current Adaptor For Scopes And DMMs

FEATURES • Jump Start – Temperature Alarm • Techno Talk • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work

AUG '13

PROJECTS • Driveway Sentry • Milliohm Meter Adaptor For DMMs • Build A Vox • Superb Four-Channel Amplifier – On The Cheap

FEATURES • Techno Talk • Circuit Surgery • Interface • Max's Cool Beans • Net Work • Computer Error: Reliable Digital Processing – Part 1

SEPT '13

PROJECTS • Digital Sound Effects Module • USB Stereo Recording & Playback Interface • Vacuum Pump From Junk • Minireg 1.3-22V Adjustable Regulator • Ingenuity Unlimited

FEATURES • Techno Talk • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work • Computer Error: Reliable Digital Processing – Part 2

OCT '13

PROJECTS • LED Musicolour – Part 1 • High-Temperature Thermometer/Thermostat • Ingenuity Unlimited

FEATURES • Teach-In 2014 – Part 1 • Techno Talk • Circuit Surgery • Interface • Max's Cool Beans • Net Work • Computer Error: Reliable Digital Processing – Part 3

NOV '13

PROJECTS • CLASSiC-D Amplifier – Part 1 • LED Musicolour – Part 2 • Mains Timer For Fans Or Lights • Ingenuity Unlimited

FEATURES • Teach-In 2014 – Part 2 • Techno Talk • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work

DEC '13

PROJECTS • Six Test Instruments In One Tiny Box • Virtins Technology Multi-Instrument 3.2 • CLASSiC-D Amplifier – Part 2 •

FEATURES • Teach-In 2014 – Part 3 • Techno Talk • Circuit Surgery • Interface • Max's Cool Beans • Net Work

JAN '14

PROJECTS • 2.5GHz 12-Digit Frequency Counter With Add-on GPS Accuracy – Part 1 • The Champion Amplifier • Simple 1.5A Switching Regulator •

FEATURES • Teach-In 2014 – Part 4 • Techno Talk • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work • PIC N' Mix • Net Work

FEB '14

PROJECTS • High-energy Electronic Ignition System – Part 1 • Mobile Phone Loud Ringer! • 2.5GHz 12-Digit Frequency Counter With Add-on GPS Accuracy – Part 2

FEATURES • Teach-In 2014 – Part 5 • Techno Talk • Circuit Surgery • Interface • Max's Cool Beans • Net Work • PIC N' Mix • Net Work

MAR '14

PROJECTS • Infrasound Detector • Extremely Accurate GPS 1pps Timebase For A Frequency Counter • High-energy Electronic Ignition System – Part 2 • Automatic Points Controller For Your Model Railway Layout

FEATURES • Teach-In 2014 – Part 6 • Techno Talk • Circuit Surgery • Practically Speaking • Max's Cool Beans • Net Work • PIC N' Mix • Net Work

Where we do not have an issue a photocopy of any *one article* or *one part* of a series can be provided at the same price.

ORDER FORM – BACK ISSUES – PHOTOCOPIES– INDEXES

☐ Send back issues dated . . . .

☐ Send photocopies of (article title and issue date) . . . .

Name . . . .

Address. . . . .

. . . . Tel: . . . .

☐ I enclose cheque/P.O./bank draft to the value of £ . . . .

☐ Please charge my Visa/Mastercard/Maestro £ . . . .

Card No . . . . Maestro Issue No . . . .

Valid From . . . . Card Expiry Date . . . . Card Security Code . . . .
(The last 3 digits on or just under the signature strip)

SEND TO: **Everyday Practical Electronics, Wimborne Publishing Ltd., 113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU.**
Tel: 01202 880299. Fax: 01202 843233.
E-mail: orders@epemag.wimborne.co.uk **On-line Shop**: www.epemag.com

Payments must be in £ sterling – cheque or bank draft drawn on a UK bank. Normally posted within seven days of receipt of order. Send a copy of this form, or order by letter if you do not wish to cut your issue. M04/14

EPE EXCLUSIVE

Win a PICDEM Lab Development Kit

EVERYDAY PRACTICAL ELECTRONICS is offering its readers the chance to win a PICDEM Lab Development Kit. The PICDEM Lab Development Kit (#DM163045) is designed to provide a comprehensive development and learning platform for Microchip's FLASH-based 6-, 8-, 14-, 18- and 20-pin 8-bit PIC microcontrollers. Geared toward first-time PIC microcontroller users and students, the development kit is supplied with five of Microchip's most popular 8-bit PIC microcontrollers and a host of discrete components used to create a number of commonly used circuits. Expansion headers provide complete access/connectivity to all pins on the connected PIC microcontrollers and all mounted components.

A solderless prototyping block is included for quick exploration of the application examples described in the 'hands-on' labs included in the user's guide. These labs provide an intuitive introduction to using common peripherals and include useful application examples, from lighting an LED to some basic *mixed signal* applications using the free HI-TECH C PRO for the PIC10/12/16 MCU Family Lite Mode Compiler.

Alternately, a companion guide featuring the free version of Matrix Multimedia's Flowcode V3 Visual Programming Environment (VPE) provides a flowchart-based method of implementing a series of introductory labs. A free version of Flowcode V3 can be downloaded from the Microchip website.

Completing the kit are Microchip's PICkit 3 Programmer/Debugger, which should be used for development purposes only, and a suite of free software tools that enable original applications to be developed quickly.

HOW TO ENTER

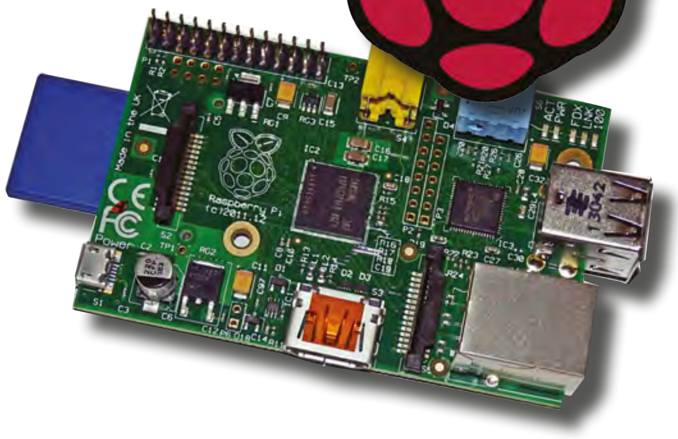
For your chance to win a PICDEM Lab Development Kit, please visit:
http://www.microchip-comps.com/epe-piclabdev and enter your details in the entry form.

CLOSING DATE

The closing date for this offer is 30 April 2014

Teach-In 2014
Raspberry Pi – Part 7

by Mike and Richard Tooley



Welcome to *Teach-In 2014* with Raspberry Pi. This exciting series has been designed for electronics enthusiasts wanting to get to grips with the immensely popular Raspberry Pi, as well as computer buffs eager to explore hardware and interfacing. So, whether you are considering what to do with your Pi, or maybe have an idea for a project but don't know how to turn it into reality, our new *Teach-In* series will provide you with a one-stop source of ideas and practical information.

The Raspberry Pi offers you a remarkably effective platform for developing a huge variety of projects; from operating a few lights to remotely controlling a robotic vehicle through the Internet. *Teach-In 2014* is based around a series of practical exercises with plenty of information for you to customise each project to meet your own requirements.

The Raspberry Pi is no mean performer; it can offer you very similar performance to that which you might expect from a larger and much more expensive computer system, so don't be fooled by the relatively small price tag. By shopping around you can build a very effective computer system based on a Raspberry Pi for less than £100. However, if you are looking for something more modest and just want to take advantage of the Raspberry Pi as a single-board computer for a particular control application then you can be up and running for a very reasonable outlay.

This series will teach you about:

- **Programming** – introducing you to the powerful Python programming language and allowing you to develop your programming skills
- **Hardware** – learning about the components and circuits that are used to interface microcomputers to the real world
- **Computers** – letting you get to grips with computer hardware and software and helping you understand how they work together
- **Communications** – showing you how to connect your Raspberry Pi to a network and control a remote device using Wi-Fi and the Internet.

So, what's coming up? Regular features of Teach-In 2014 with Raspberry Pi will include:

- **Pi Project** – the main topic for each part will be a project that explores a particular use or application of the Raspberry Pi in the real word. Projects will include shopping for your Pi, set up, environmental monitoring, data logging, automation and remote control.
- **Pi Class** – each of our Pi Projects will be linked to one or more specific learning aims. Examples will include methods of representing and handling data, serial versus parallel data transmission and architecture of a microprocessor system.
- **Python Quickstart** – a short feature devoted to specific programming topics, such as data types and structures, processing user input, creating graphical dialogues and buttons and importing Python modules. We will help you get up and running with Python in the shortest time!
- **Pi World** – this is where we take a look at a wide range of Raspberry Pi accessories, including breadboards, prototype cards, bus extenders and Wi-Fi adapters. We will also help you build your Raspberry Pi bookshelf with a selection of recommended books and other publications.
- **Home baking –** suggested follow-up and extension activities such as 'check this out', a simple quiz, things to try and websites to visit.
- **Special features –** an occasional 'special feature'. For example, how to laser cut your own mounting plate – with additional downloadable resources such as templates and diagrams.

What will I need?

To get the best out of our series you will, of course, need access to a Raspberry Pi. If you don't already have one, don't worry – we will be explaining what you need and why you need it (we will also be showing you how you can emulate a Raspberry Pi using a Windows PC).

This month

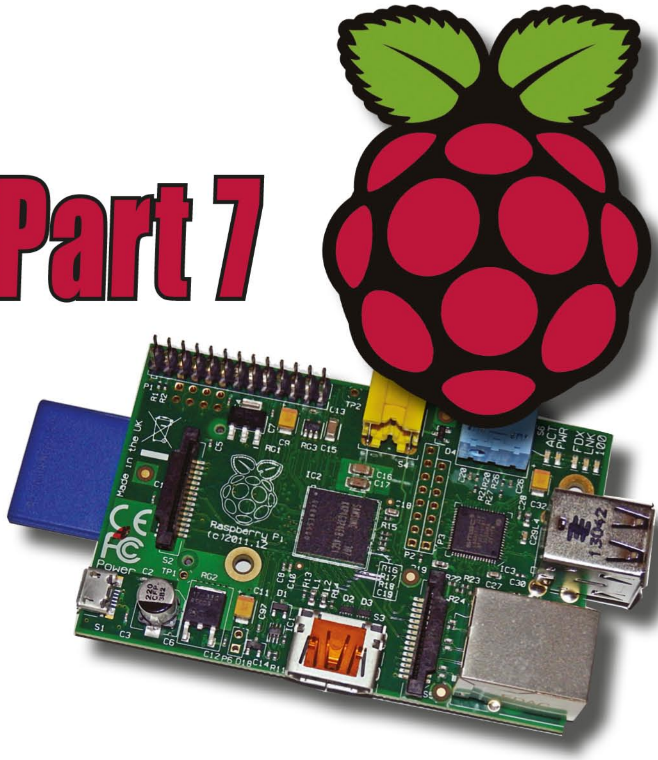
In this month's *Teach-In 2014*, our *Pi Project* features the construction of a 16-channel port expander for the Raspberry Pi. This handy interface uses the System Management Bus (SMBus) as an alternative means of accessing a wide-range of I^2C devices. Our *Pi Class* introduces binary, octal and hexadecimal numbers; and *Python Quickstart* deals with methods of handling and manipulating binary, octal and hexadecimal data. Finally, in *Home Baking* we will show you how to configure your Raspberry Pi for use as a web server that can be accessed anywhere in the world.

Pi Class

In this month's *Pi Class* we shall be looking at alternative ways of representing numbers. This can be particularly useful when we need to send or receive data from an I/O port. We will start with a quick review of the denary (base 10) number system before introducing binary (base 2) representation, octal (base 8) and hexadecimal (base 16). Later, in *Python Quickstart*, we will be putting this into good use with some handy routines that will help you quickly convert numerical data from one form to another.

Teach-In 2014
Raspberry Pi – Part 7

by Mike and Richard Tooley



Welcome to *Teach-In 2014* with Raspberry Pi. This exciting series has been designed for electronics enthusiasts wanting to get to grips with the immensely popular Raspberry Pi, as well as computer buffs eager to explore hardware and interfacing. So, whether you are considering what to do with your Pi, or maybe have an idea for a project but don't know how to turn it into reality, our new *Teach-In* series will provide you with a one-stop source of ideas and practical information.

The Raspberry Pi offers you a remarkably effective platform for developing a huge variety of projects; from operating a few lights to remotely controlling a robotic vehicle through the Internet. *Teach-In 2014* is based around a series of practical exercises with plenty of information for you to customise each project to meet your own requirements.

The Raspberry Pi is no mean performer; it can offer you very similar performance to that which you might expect from a larger and much more expensive computer system, so don't be fooled by the relatively small price tag. By shopping around you can build a very effective computer system based on a Raspberry Pi for less than £100. However, if you are looking for something more modest and just want to take advantage of the Raspberry Pi as a single-board computer for a particular control application then you can be up and running for a very reasonable outlay.

This series will teach you about:

- **Programming** – introducing you to the powerful Python programming language and allowing you to develop your programming skills
- **Hardware** – learning about the components and circuits that are used to interface microcomputers to the real world
- **Computers** – letting you get to grips with computer hardware and software and helping you understand how they work together
- **Communications** – showing you how to connect your Raspberry Pi to a network and control a remote device using Wi-Fi and the Internet.

So, what's coming up? Regular features of Teach-In 2014 with Raspberry Pi will include:

- **Pi Project** – the main topic for each part will be a project that explores a particular use or application of the Raspberry Pi in the real word. Projects will include shopping for your Pi, set up, environmental monitoring, data logging, automation and remote control.
- **Pi Class** – each of our Pi Projects will be linked to one or more specific learning aims. Examples will include methods of representing and handling data, serial versus parallel data transmission and architecture of a microprocessor system.
- **Python Quickstart** – a short feature devoted to specific programming topics, such as data types and structures, processing user input, creating graphical dialogues and buttons and importing Python modules. We will help you get up and running with Python in the shortest time!
- **Pi World** – this is where we take a look at a wide range of Raspberry Pi accessories, including breadboards, prototype cards, bus extenders and Wi-Fi adapters. We will also help you build your Raspberry Pi bookshelf with a selection of recommended books and other publications.
- **Home baking –** suggested follow-up and extension activities such as 'check this out', a simple quiz, things to try and websites to visit.
- **Special features –** an occasional 'special feature'. For example, how to laser cut your own mounting plate – with additional downloadable resources such as templates and diagrams.

What will I need?

To get the best out of our series you will, of course, need access to a Raspberry Pi. If you don't already have one, don't worry – we will be explaining what you need and why you need it (we will also be showing you how you can emulate a Raspberry Pi using a Windows PC).

This month

In this month's *Teach-In 2014*, our *Pi Project* features the construction of a 16-channel port expander for the Raspberry Pi. This handy interface uses the System Management Bus (SMBus) as an alternative means of accessing a wide-range of I^2C devices. Our *Pi Class* introduces binary, octal and hexadecimal numbers; and *Python Quickstart* deals with methods of handling and manipulating binary, octal and hexadecimal data. Finally, in *Home Baking* we will show you how to configure your Raspberry Pi for use as a web server that can be accessed anywhere in the world.

Pi Class

In this month's *Pi Class* we shall be looking at alternative ways of representing numbers. This can be particularly useful when we need to send or receive data from an I/O port. We will start with a quick review of the denary (base 10) number system before introducing binary (base 2) representation, octal (base 8) and hexadecimal (base 16). Later, in *Python Quickstart*, we will be putting this into good use with some handy routines that will help you quickly convert numerical data from one form to another.

Table 7.1. Equivalence between the first sixteen denary, binary, octal and hexadecimal numbers

Denary (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Python Quickstart

Last month, we looked at ways of validating user input, checking that it contains only valid characters and, where appropriate, converting it to the correct numeric format. This month, we will be showing you how to handle and manipulate binary, octal and hexadecimal data.

Changing base

Python has a number of useful built-in functions that allow you to represent numbers in a base that is different from the default base 10. Consider the following fragment of code:

```
n = 123
# Print the value of n using the default base 10
print(n)
```

Not surprisingly the printed result is: `123`

There are many occasions that we might need the result to be printed in a different base, for example base 2 (binary), or base 16 (hexadecimal). Fortunately, Python has some built-in functions that can take care of it.

Binary (base 2)

First, let's assume that we need to display the value of a variable, n, in binary (base 2). Here we can make use of Python's `bin()` function, as follows:

```
n = 123
# Print the value of n in binary
print(bin(n))
```

When this code is executed the result is: `0b1111011`

Notice how Python has preceded the number with 0b in order to indicate that the number is displayed in binary (base 2). The binary number (without the leading 0b) is just 1111011. You can check that this is correct by simply adding the denary (base 10) values of each of the set bits, as follows:

1111011_2 (base 2) $= 64_{10} + 32_{10} + 16_{10} + 8_{10} + 0_{10} + 2_{10} + 1_{10} = 123_{10}$

In this example we have used the subscript (either 2 or 10) as another way of indicating the number base (note that 1111011_2 is the same as Python's 0b1111011).

Hexadecimal (base 16)

If we need to display the value of n in hexadecimal (base 16). We can simply use Python's `hex()` function, as follows:

```
n = 123
# Print the value of n in hexadecimal
print(hex(n))
```

When this code is executed the result is: `0x7b`

This time Python has preceded the number with 0x to remind us that the number is displayed in hexadecimal (base 16). The hexadecimal number (without the leading 0x) is just 7b. You can check that this is correct by simply adding the denary (base 10) values of each of the hexadecimal digits, as follows:

$7b_{16} = (7 \times 16)_{10} + (11 \times 1)_{10} = 112_{10} + 11_{10} = 123_{10}$

In this example we have used the subscript (either 16 or 10) to indicate the number base (note that 7b16 is the same as Python's 0x7b.

Octal (base 8)

Not content with just being able to convert numbers to their binary and hexadecimal equivalents, Python also provides you with a means of displaying the octal (base 8) equivalent of a number. You can do this using the `oct()` function, as follows:

```
n = 123
# Print the value of n in octal
print(oct(n))
```

When this code is executed the result is: `0o173`

This time Python has preceded the number with 0o to remind us that the number is displayed in octal (base 8). The hexadecimal number (without the leading 0o) is just 173. You can check that this is correct by simply adding the denary (base 10) values of each of the octal digits, as follows:

$173_8 = (1 \times 64)_{10} + (7 \times 8)_{10} + (3 \times 1)_{10} = 64_{10} + 56_{10} + 3_{10} = 123_{10}$

In this example we have used a subscript (either 8 or 10) to indicate the number base (note that 173_8 is the same as Python's 0o173).

Better formatting

There are some occasions you might just want to display the number in binary, octal or hexadecimal without the 0b, 0o or 0x prefix. You might also want to display a specific number of digits (for example, 8 or 16-bits for a binary number or four or eight characters for a hexadecimal value). Python offers you a very neat way of doing this using formatted printing.

To bring this to a conclusion, here's a block of code that will allow you to enter a number and display its neatly formatted binary, octal and hexadecimal equivalents:

```
# Python 3.3

value_input = input("Value to convert: ")
n = int(value_input)

# Print eight bit binary number without 0b
# and pad with leading zeros if less than 8-bits

print('Binary = {:08b}'.format(n))

# Print three digit octal value without 0o
# and pad with leading zeros if less than 3-digits

print('Octal = {:03o}'.format(n))

# Print four digit hexadecimal without 0x
# and pad with leading zeros if less than 4-characters

print('Hex = {:04x}'.format(n))
```

Pi Project

Last month, we described a simple digital-to-analogue converter (DAC) based on the MCP4725 chip which uses an I^{2}C interface to the host controller. This month, our Pi Project deals with the construction of a port expander that will provide you with a further 16 programmable I/O channels.

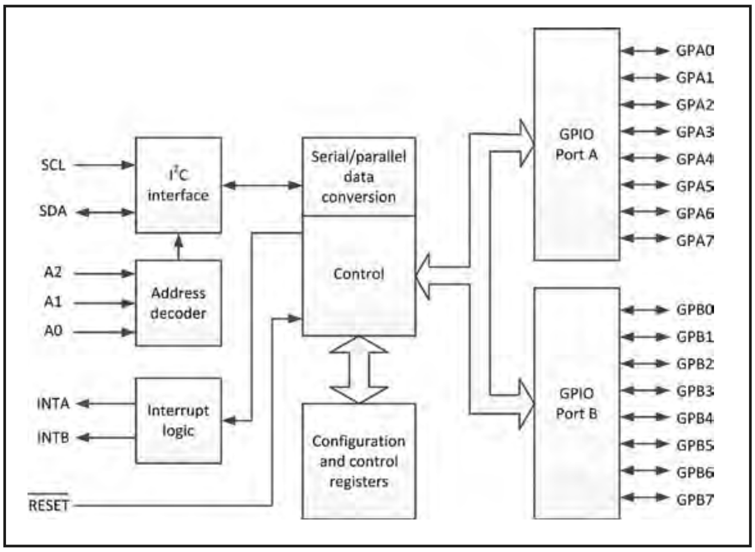


Fig.7.6. Simplified block schematic of the MCP23017 I/O device

Table 7.2. MCP23017 signals

Signal	Pin no.	Direction	Function
GPB0	1	I/O	Port B bit-0 (note 1)
GPB1	2	I/O	Port B bit-1 (note 1)
GPB2	3	I/O	Port B bit-2 (note 1)
GPB3	4	I/O	Port B bit-3 (note 1)
GPB4	5	I/O	Port B bit-4 (note 1)
GPB5	6	I/O	Port B bit-5 (note 1)
GPB6	7	I/O	Port B bit-6 (note 1)
GPB7	8	I/O	Port B bit-7 (note 1)
V_{DD}	9	n/a	Positive supply
V_{SS}	10	n/a	Ground
NC	11	n/a	Not connected
SCL	12	I	Serial clock
SDA	13	I/O	Serial data
NC	14	n/a	Not connected
A0	15	I	Device address bit-0 (note 2)
A1	16	I	Device address bit-1 (note 2)
A2	17	I	Device address bit-2 (note 2)
/RESET	18	I	Hardware reset (active low)
INTB	19	O	Port B interrupt output (note 3)
INTA	20	O	Port A interrupt output (note 3)
GPA0	21	I/O	Port A bit-0 (note 1)
GPA1	22	I/O	Port A bit-1 (note 1)
GPA2	23	I/O	Port A bit-2 (note 1)
GPA3	24	I/O	Port A bit-3 (note 1)
GPA4	25	I/O	Port A bit-4 (note 1)
GPA5	26	I/O	Port A bit-5 (note 1)
GPA6	27	I/O	Port A bit-6 (note 1)
GPA7	28	I/O	Port A bit-7 (note 1)

Table notes

1. Can be enabled for interrupt-on-change and/or configured with an internal weak pull-up resistor
2. Logic level must be externally biased
3. Can be configured as active-high, active-low or open-drain

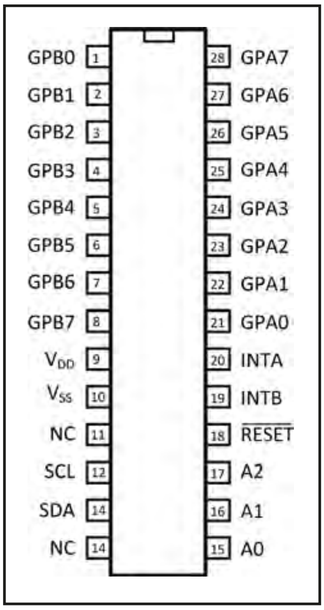


Fig.7.7. Pin connections for the MCP23017

MCP23017 I/O device

Our port expander makes use of a Microchip MCP23017 general purpose parallel I/O device. This device is fitted with an I^{2}C interface (the MCP23S17 is similar, but is fitted with an SPI interface). The MCP23017 contains an array of 8-bit registers that are used for configuring input, output and polarity selection, and the device is configured by writing I/O configuration bits to the appropriate registers. If necessary, the polarity of each input port can be inverted by writing data to the respective polarity inversion register.

The 16-bit I/O functionality is provided through two separate 8-bit I/O ports, PORTA and PORTB – but, where necessary, the chip can be configured to operate in a 16-bit (rather than an 8-bit) mode. An active-low Power-on Reset (POR) signal is available to set the MCP23017's registers to an initial default value.

The simplified internal arrangement of the MCP23017 is shown in Fig.7.6, and the signals are summarised in Table 7.2. The MCP23017 is supplied in a 28-pin plastic dual-in-line (DIL) package, as shown in Fig.7.7.

Interrupts

The MCP23017 can be used for applications that require an interrupt signal to be generated. This is a signal that can be used to alert the host controller to the need to respond to an external condition that might need immediate attention. The device has two interrupt pins, INTA and INTB, associated with their respective ports (see Table 7.2). Alternatively, they can be logically ORed together so that both pins will activate if either port causes an interrupt. For more information on interrupt operation you should consult Microchip's documentation.

Addressing

Three hardware address pins (A0, A1 and A2) can be used to determine the address of the MCP23017 chip. Using these three bits, a total of eight individual addresses can be configured by placing appropriate logic levels on the address lines.

Prototype port expander

The complete circuit of the 16-channel port expander is shown in Fig.7.8. The eight I/O lines for Port A are taken to an 18-way connector, SK2. Those for Port B are taken to a second 18-way connector, SK2. Note that these two connectors are not wired identically and they are therefore not interchangeable. When a particular application needs eight (or less) I/O lines it is only necessary to use one of the 8-bit ports. For more than eight I/O lines, or for a full 16-bit I/O application, both ports will be required. Note that, in addition to the I/O signal lines, each I/O connector provides nine ground (0V) connections and a single +3.3V line.

In our prototype, the three address lines, A0 to A2, were held low (via R2, R3 and R4) so that, within the MCP23017's address space, the chip is given a hexadecimal address of 00 (see earlier). The active-low

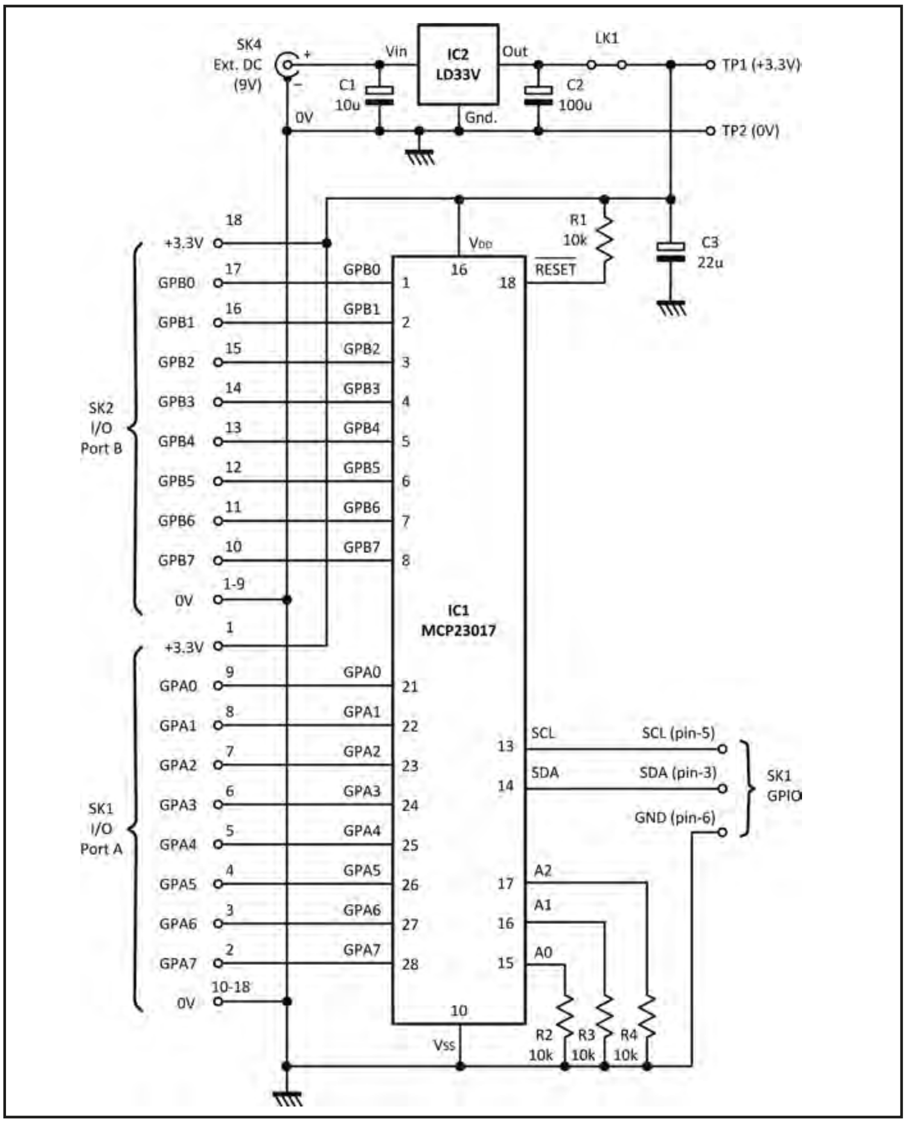


Fig.7.8. Complete circuit of the 16-channel port expander

RESET line is held high via R1. However, if it should ever become necessary to reset the chip, pin-18 can be taken low (this facility was not needed on the prototype board).

A 3.3V regulator, IC2, provides the supply for the MCP23017 (IC1). An external (unregulated) supply of nominally 9V at up to 500mA provides the input to IC2. For applications that only have modest current requirements (up to 50mA, or so) the regulator is not required and IC2 can derive its positive supply from the Raspberry Pi (pin-1 on P1). Despite this, and to ensure maximum protection for your Pi (avoiding the risk of inadvertent supply misconnection) we would strongly recommend the use of a suitably rated external power supply and an additional voltage regulator for all I/O applications. For test purposes, test points TP1 and TP2 can be fitted in order to provide access to the +3.3V line and, if necessary, LK1 can be removed in order to isolate the MCP23017 from its local +3.3V supply.

The only connections required to the Raspberry Pi via the GPIO connector (P1 on the Raspberry Pi) are made via the ground (common 0V), serial clock (SCL) and serial data (SDA) lines. These connections are made via pins 6, 5 and 3 (respectively) on the GPIO header (SK1).

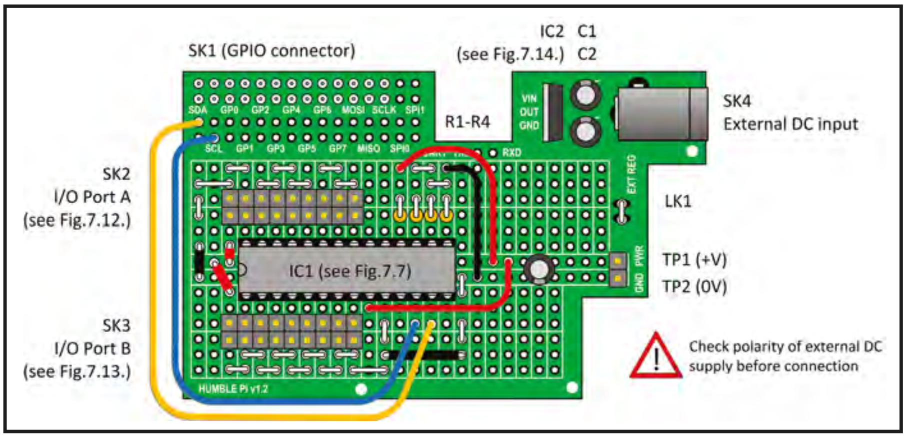


Fig.7.9. Humble Pi prototype board layout

The Humble Pi prototyping board

We built our 16-channel I/O port expander using a Humble Pi prototyping board and the wiring layout shown in Fig.7.9. The Humble Pi prototyping board should be fitted with the 26-way connector (SK1) supplied with the Humble Pi. This connector mates with the male connector (P1) on the Raspberry Pi and it ensures that the Humble Pi board sits neatly piggy-back style above the Raspberry Pi. The completed prototype board is shown in Fig.7.10.

Do notice that there have been some recent changes to the pin labelling convention used on Humble Pi prototyping boards. The latest Version 1.3 boards use pin labelling based on the Broadcom (BCM) chip signals, while earlier versions were based on the earlier Raspberry Pi GPIO pin convention. We've summarised the changes in Table 7.3 and in Fig.7.11.

The pin connections for the Port A I/O connector (SK2) are shown in Fig.7.12, while those for Port B I/O (SK3) are shown in Fig.7.13. The power supply (and associated components) is based on the kit supplied as an optional extra for use with the Humble Pi prototyping board. The kit consists of a low-dropout voltage regulator (LD33V, LD50V, or equivalent), two capacitors, and a standard DC power connector. Note that, for this application, you will require a 3.3V regulator (LD33V) and not a 5V regulator (LD50). The four additional power supply components can be quickly and easily fitted to the Humble Pi prototype board (see Fig.7.10).

Table 7.3. Pin labelling conventions used on Humble Pi prototyping boards

Early versions (Raspberry Pi GPIO)	Later versions (BCM signals)
GP0	G17
GP1	G18
GP2	G27
GP3	G22
GP4	G23
GP5	G24
GP6	G25
GP7	G04

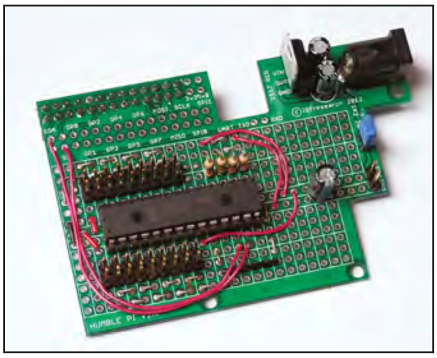


Fig.7.10. Humble Pi prototype 16-channel port expander

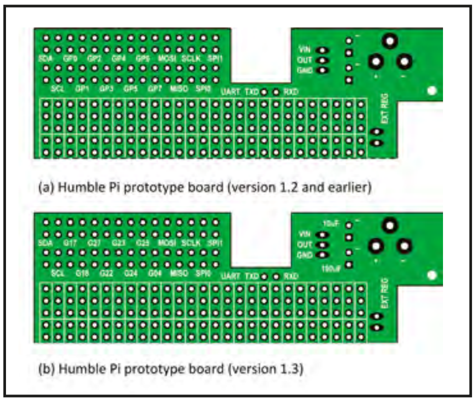


Fig.7.11. Changes to the Humble Pi pin labelling

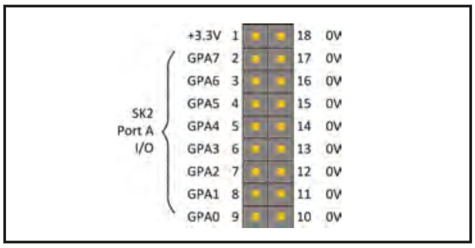


Fig.7.12. Port A I/O connector SK2 pin connections

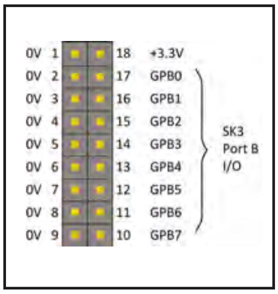


Fig.7.13. Port B I/O connector SK3 pin connections

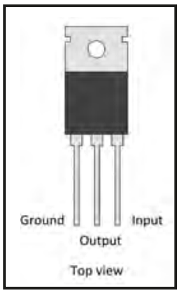


Fig.7.14. IC2 pin connections

Connections to 3.3V regulator (IC2) are shown in Fig.7.14.

Testing the 16-channel I/O expander

The 16-channel I/O expander can be easily tested using LEDs and switches connected to the I/O ports. The LED (as output) connections for Port A and Port B are shown in Fig.7.15 and 7.16. While a suitable arrangement for connecting switches (as input) to Port A are

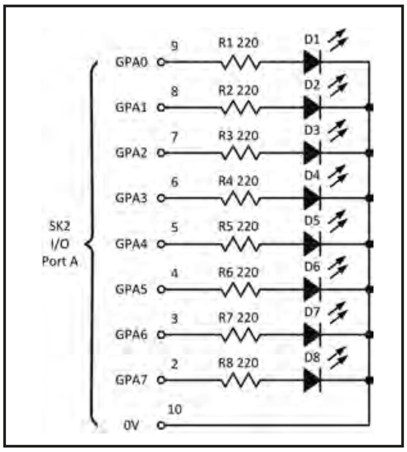


Fig.7.15. LED connection to Port A

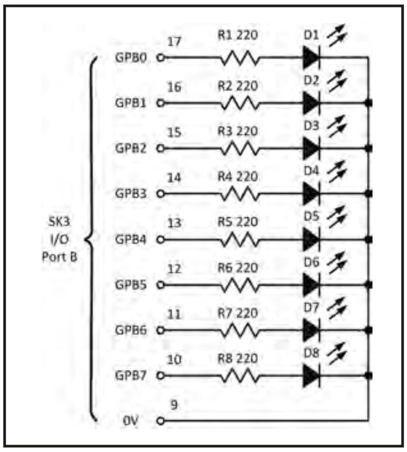


Fig.7.16. LED connection to Port B

shown in Fig.7.17. Suitable software is discussed in the next section.

Accessing the SMBus interface

In order to make use of the SMBus interface you will need to have the i2c-tools package installed in your Raspberry Pi. In most current Debian distributions the package will already be installed, but if you do need to install the package you can use `sudo apt-get install i2c-tools` from the system prompt. Then, following installation you should add a new user to the i2c group using `sudo adduser pi i2c` before rebooting the system using `sudo shutdown –r now`. Note that you will not need to do any of this if you already have the i2c-tools package installed.

Now, to check for I^{2}C devices connected to your Raspberry Pi you will need to use `i2cdetect` from within LXTerminal (or equivalent) as follows:

```
sudo i2c detect -y 0
```

As explained in last month's boxed feature, a Rev.2 board needs the following command:

```
sudo i2c detect -y 1
```

A typical display produced by this command was shown in Fig.6.9 (last month). The display shows the mapping of 128 addresses within the i2c space (but note that a few addresses are reserved for special purposes, so not all of the 128 possible addresses can be mapped to i2c devices). If you look carefully at the display you should see that address 20 (hex.) has been mapped to the MCP23017 device and so we are now ready to start using it.

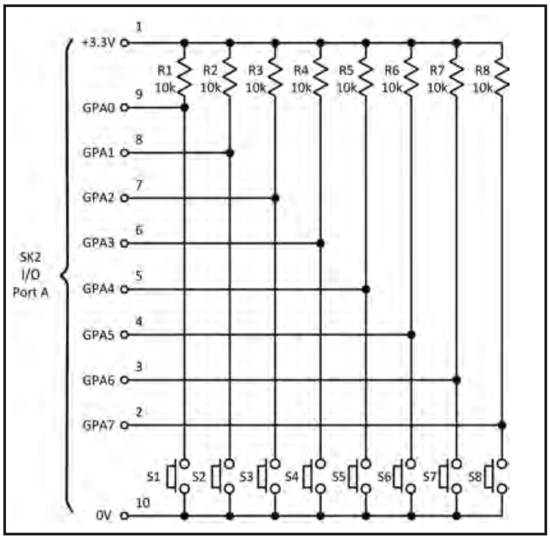


Fig.7.17. Switch input connections to Port A

System Management Bus (SMBus)

In Teach-In Part 6 we introduced you to the popular I^{2}C bus. This bus uses a single line (SDA) for bidirectional data and a second signal line for a clock signal (SCL). Originally developed by Intel in 1995, SMBus provides a stricter definition of I^{2}C that helps to improve the interoperability of I^{2}C devices from a wide range of different manufacturers.

Like I^{2}C, SMBus is a two-wire bus designed primarily to provide a simple and effective means of communication between a host controller and a number of peripheral devices. Unlike, I^{2}C (which originally focused on standard 5V logic), SMBus employs different logic levels to represent its high (logic 1) and low (logic 0) states.

The latest I^{2}C specification defines the fixed logic levels as less than 1.5V for logic 0 and more than 3.0V for logic 1. The logic levels can also be defined in terms of the positive supply voltage, V_{DD}. These are $0.7V_{DD}$ for the high state and $0.3V_{DD}$ for the low state (note how this definition is clearly based on the use of a +5V supply as it agrees exactly with the fixed logic level voltages of 3.0V and 1.5V respectively. In the case of the SMBus, logic high and low levels correspond to 2.1V and 0.8V respectively. This helps to ensure that the SMBus specification caters effectively for bus implementations where the positive supply voltage ranges from as little as 3.0V to as much as 5.0V.

It is also worth noting that the difference in logic levels also has an effect on the range of resistance values that we can use for pull-up and pull-down resistances. SMBus specifies a minimum sink current of 100μA, and a maximum of 350μA. This contrasts with 3mA for the I^{2}C bus and it implies that the minimum pull-up resistance value for I^{2}C bus running on a 3V supply should be 1kΩ, while for a 5V supply it should be 1.6kΩ. For the SMBus the respective values would be 8.5kΩ and 14kΩ. Note also that, by virtue of a small amount of stray capacitance present on the bus, the value of any pull-up resistance will need to be reduced when the bus is to be operated at high speed. This, in turn, will impact on power consumption which will usually increase significantly as the bus speed increases.

Happily, despite these obvious differences in logic-level and pull-up specifications, devices can usually be mixed and matched between I^{2}C and SMBus without any real difficulty.

To import the `smbus` library module into your code, include the following line near the beginning of your Python code:

```
import smbus
```

To simplify your code you might find it useful to define your bus at the start of your program. If you are using a Rev.1 Raspberry Pi you will need the following line of code:

```
bus = smbus.SMBus(0)
```

For a Rev.2 Raspberry Pi your code will need to be as follows:

```
bus = smbus.SMBus(1)
```

You might also want to add the following a few definitions at the start of your code. These definitions will help make the code a little more understandable later on:

```
DEVICE = 0x20 # Device address when A0-A2 are reset
GPIOA = 0x12  # Port A data
GPIOB = 0x13  # Port B data
IODIRA = 0x00 # Port A direction
IODIRB = 0x01 # Port B direction
```

Notice how we've used hexadecimal numbers (see earlier) in all of these definitions.

Programming the slave address

If you have several MCP23017 bus expanders attached to your Raspberry Pi you will need to ensure that each device has its own unique 'slave' address. The slave address forms the seven most significant bits of the control byte that is sent to configure the MCP23017 (see Fig.7.18). The slave address can be determined from the data shown in Table 7.4.

Let's assume you want to configure all of the Port A lines inputs and all of the Port B lines as outputs. To do this we need to write hexadecimal values of FF to the Port A direction register (IODIRA) and 00 into the Port B direction register (IODIRB) as follows:

```
# MCP23017 for Python 2.x
# Import the required libraries
import time
import smbus

bus = smbus.SMBus(0) # change 0 to 1 for Rev.2 boards

# Define constants to use later on
DEVICE = 0x20 # Device address when A0-A2 are reset
GPIOA = 0x12  # Port A data
GPIOB = 0x13  # Port B data
IODIRA = 0x00 # Port A direction
IODIRB = 0x01 # Port B direction
# Initialise Port A as an input
bus.write_byte_data(DEVICE,IODIRA,0xff)

# Initialise Port B as an output
bus.write_byte_data(DEVICE,IODIRB,0x00)
```

Table 7.4. Slave address format (bits 1 to 7 of the control byte – see Fig.7.18.)

A5	A4	A3	A2	A1	A0	Hex. address
0	1	0	0	0	0	20
0	1	0	0	0	1	21
0	1	0	0	1	0	22
0	1	0	0	1	1	23
0	1	0	1	0	0	24
0	1	0	1	0	1	25
0	1	0	1	1	0	26
0	1	0	1	1	1	27

Table notes

1. Dark shaded cells are permanently programmed within the MCP23017 and cannot be changed.
2. Lines A2, A1 and A0 must be programmed by external pull-up/pull-down resistors.

Now suppose that you want to alternately flash a bank of LEDs connected to Port B. We could do this by setting up a loop and outputting data first of 55 (01010101_2) for a short period followed by data of AA (10101010_2) for a short period. The code to do this would be as follows:

```
# MCP23017 for Python 2.x
# Import the required libraries
import time
import smbus

bus = smbus.SMBus(0) # change 0 to 1 for Rev.2
boards

# Define constants to use later on
DEVICE = 0x20 # Port expander's I2C bus address
IODIRB = 0x01 # Direction register for Port B
OLATB = 0x15 # Output latch for Port B
GPIOB = 0x13 # Data register for Port B

# Initialise Port B as an output
bus.write_byte_data(DEVICE,IODIRB,0x00)

# Initialise Port B with all latches in the
high state
bus.write_byte_data(DEVICE,OLATB,0xff)

while 1:
   # Send 10101010 bit pattern to port B
   bus.write_byte_data(DEVICE,GPIOB,0xaa)
   # Wait 0.5 second
   time.sleep(0.5)
   # Send 01010101 bit pattern to port B
   bus.write_byte_data(DEVICE,GPIOB,0x55)
   # Wait 0.5 second
   time.sleep(0.5)
```

Notice that we've used the `time.sleep()` function to hold the LED display for a time interval of 0.5s.

Next, here's a simple LED sequencer that moves a single LED across the display. The program works by setting up a loop with an initial data value of 00000001_2, sending this value to the Port B I/O register (configured as an output) and then successively shifting (ie, multiplying by 2) this data value before sending it again to the Port B I/O register:

```
# MCP23017 for Python 2.x
# Import the required libraries
import time
import smbus

bus = smbus.SMBus(0) # change 0 to 1 for Rev.2
boards

# Define constants to use later on
DEVICE = 0x20 # Port expander's I2C bus address
IODIRB = 0x01 # Direction register for Port B
```

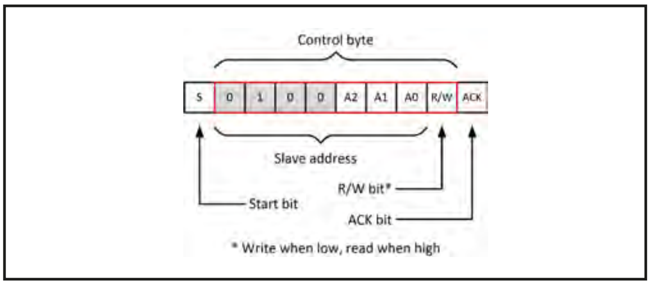


Fig.7.18. Format of the MCP23017's I^2C control byte

```
OLATB = 0x15 # Output latch for Port B
GPIOB = 0x13 # Data register for Port B

# Initialise Port B as an output
bus.write_byte_data(DEVICE,IODIRB,0x00)

# Initialise Port B with all latches in the high state
bus.write_byte_data(DEVICE,OLATB,0xff)

count = 0

while 1:
   # Shift bit pattern right
   pattern=0b00000001 # Single bit set
   count = 0 # Reset the count
   while count < 8:
      # Send bit pattern to port B
      bus.write_byte_data(DEVICE,GPIOB,pattern)
      pattern = 2 * pattern
      # Wait 0.1 second
      time.sleep(0.1)
      count = count + 1
   count = 0 # Reset the count
   pattern = pattern / 2
   # Shift bit pattern left
   while count < 8:
      # Send bit pattern to port B
      bus.write_byte_data(DEVICE,GPIOB,pattern)
      pattern = pattern / 2
      # Wait 0.1 second
      time.sleep(0.1)
      count = count + 1
```

Next, here's some code that continuously reads the status of a bank of switches connected to Port A and then outputs the value as a binary display on the screen. By examining the display and noting the position of any bits that are set (ie, 1) you can determine which of the switches (if any) have been pressed:

```
# MCP23017 for Python 2.x
# Import the required libraries
import time
import smbus

bus = smbus.SMBus(0) # change 0 to 1 for Rev.2 boards

# Define constants for later use
DEVICE = 0x20 # Port expander's I2C bus address
IODIRA = 0x00 # Direction register for Port A
GPIOA = 0x12 # Data register for Port A

# Initialise Port A as an input
bus.write_byte_data(DEVICE,IODIRA,0xff)

while 1:
   # Get input state of Port A
   input_state = bus.read_byte_data(DEVICE,GPIOA)
   print('{0:08b}'.format(input_state))
   # Wait 1 second
   time.sleep(1)
```

Finally, here's some code that continuously reads a bank of switches connected to Port A before outputting the state of the switches to a bank of LEDs connected to Port B:

```
# MCP23017 for Python 2.x
# Import the required libraries
import time
import smbus

bus = smbus.SMBus(0) # change 0 to 1 for Rev.2 boards

# Define constants to use later on
DEVICE = 0x20 # Port expander's I2C bus address
IODIRA = 0x00 # Direction register for Port A
GPIOA = 0x12 # Data register for Port A
IODIRB = 0x01 # Direction register for
Port B
GPIOB = 0x13 # Data register for Port B
OLATB = 0x15 # Output latch for Port B

# Initialise Port A as an input
bus.write_byte_data(DEVICE,IODIRA,0xff)

# Initialise Port B as an output
bus.write_byte_data(DEVICE,IODIRB,0x00)

# Initialise Port B with all latches in
the high state
bus.write_byte_data(DEVICE,OLATB,0xff)

while 1:
   # Get switch pattern from Port A
 pattern=bus.read_byte_data(DEVICE,GPIOA)
   # Write switch pattern to Port B
 bus.write_byte_data(DEVICE,GPIOB,pattern)
   # Wait 0.5 second
   time.sleep(0.5)
```

Home Baking

One of the best things about the Pi compared with simpler embedded prototyping systems is that being essentially a fully-fledged Linux system we've got network connectivity 'straight out of the box'. This of course opens up incredible possibilities in terms of exciting web connected projects. This massively broadens the opportunities for both fledgling and experienced electronics enthusiasts to take their ideas to the next level.

In this month's *Home Baking* we're going to look at the first step of this process; setting up your Pi as a web server that's accessible externally from the web. We will describe installing web server software, a scripting language and the settings you'll need to allow you to communicate with the rest of the world. The process is actually pretty easy, although there are several steps and things that all need to be setup and working harmoniously in order to achieve the end result. We should note here that the article describes *general steps* – it's obviously not possible to be specific about everyone's hardware setup. Therefore, you should use our instructions alongside any relevant documentation for your own equipment, as well as the plentiful Internet resources and tutorials to help you get up and running with your own particular setup.

It should also be noted that in order to connect your Pi to the outside world we will need to make changes to the settings in your router that are designed to protect or 'firewall' your network. You should ensure that you fully understand and are happy with the security implications beforehand. Without due care you could be providing a route for undesirable visitors into your network, so do take care. This article will not explain the detail of creating web pages or sites and their management. However, this information is widely available.

As we've just mentioned, there are several steps to setting up your Raspberry Pi as a web server. Since this is a multi-stage process we'll be describing each step in turn:

1. Installing web server software on to your Pi
2. Installing a scripting language (optional)
3. Setting up IP addresses on your local network
4. Setting up routing to/from the web
5. Setting up dynamic DNS.

We'll describe each of these steps in turn.

1. Installing web server software

If you're planning a web-connected project you may well want to provide a web-based front end that's accessible via a web browser. For example, to allow users to view data from sensors or a camera, or to control output devices remotely. Alternatively, you may just want to use the Pi to help you experiment with web design/scripting, using the Pi merely as a test server. In either case, we are going to need to install a web server to do this job for us.

When you view a web page your browser sends a request for the page to the designated web server, which upon receiving the request, 'serves' the web page along with any page content such as graphics or media. To do this on our Pi we are going to be installing Apache HTTP Server. Apache is a very well known, free and open-source web server and is in fact the most popular web server on the Internet. To install Apache, simply open up a terminal window (or via SSH) execute the following commands and follow any on-screen prompts, as required.

```
sudo apt-get update
sudo apt-get install apache2
```

This will install the web server, start it and set it to run automatically in the background upon boot. Once complete, you can check if it's worked by opening a browser locally on your Pi itself and entering the address `http://127.0.0.1` If all has gone well, an 'It Worked!' test page should appear, served from your very own Raspberry Pi web server (Fig.7.19).

The IP address `127.0.0.1` is sometimes referred to as 'local loopback'; it effectively allows you to send a request back to yourself – in this case a request to your Pi web server from your Pi web server. Now jump on a different computer or device on your local network, open a browser and this time enter the IP address of your Raspberry Pi into the address bar eg, `http://192.168.1.99`. If you don't know your Pi's current IP address you can use the following command on your Raspberry Pi:

```
ip addr show
```

We discussed IP addresses in Part 4 of *Teach-In* and we'll be looking at setting IP addresses in Step 3. If you are able to see the same Apache test page as you did on your Pi then congratulations – you now have a Pi web server running on your local network.

Fig.7.19. Apache test page on the Midori web browser demonstrating a successful installation

If you have previous experience of web design you'll know that web pages are essentially just plain text documents written in a structured coding language known as HTML. Web page files have an extension of `.htm` or `.html`. Web pages are stored as files in a folder that is made accessible via the web through the web server. This folder is known as the web root.

The default web root for Apache on your Pi is located at `/var/www`. If you navigate to this folder you'll be able to see the test page called `index.html` (Fig.7.20). It is this directory that would contain your website pages and files. It's beyond the scope of this *Home Baking* to explain any further detail on how web pages and sites are created and managed, although there's a huge amount of resources available on these topics freely available on the Internet. If, however, you are already experienced in this respect then you should be able to go ahead and start playing around with your new web server. In next month's *Teach-In* we will be showing you how to output data from your Python code in the form of an HTML file which can be made visible to the outside world using your Pi as a web server.

2. Installing a scripting language

Having your own web server is pretty cool, but with our projects we want to do a little more than just have static web pages. We want to be able to run scripts that can, for example, run a process or read/write to the GPIO. In order to do this we'll need to install a scripting language. Hypertext Preprocessor, or PHP as it is commonly known, is a server-side scripting language that has been around for almost 20 years and used on almost a quarter of a billion websites around the globe. It's one of the most popular web scripting languages, and best of all its completely free. Installation is really easy too, just issue the following commands in terminal or SSH and follow any prompts:

```
sudo apt-get install php5
```

Once installed, we can check that it's all up and running properly using a neat in-built test function within PHP. What we are going to do here is create a new PHP script on our web server that we can run. Using File Manager,

Fig.7.20. Apache web root folder

navigate to `/var/www`. To create the file we need to have elevated privileges. To open the folder with root privileges click 'Tools' on the top menu bar and select 'Open Current Folder as Root'. A new File Manager window will appear with root access (note the key icon on the top left). You may be asked to enter root user credentials here depending on whether you normally run your Pi as a root user or not. Now create a new blank file (right-click > Create New… > Blank File) and name the file `phptest.php`. Next, edit this file in Leafpad by right-clicking and selecting 'Leafpad' from the context menu. Enter the text as follows before saving your file and exiting Leafpad (see Fig.7.21).

```
<?php
phpinfo()
?>
```

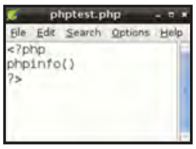


Fig.7.21. The phptest.php file being written using the Leafpad text editor

The simple script instructs PHP to run its inbuilt test function `phpinfo()` that will dynamically create a web page with all of its settings and information. To run the script, open your preferred web browser on the Pi and enter the URL `http://127.0.0.1/phptest.php`. Alternatively, you can test this from another computer on your network by replacing 127.0.0.1 with your Pi's own IP address.

When the PHP script is run, you should be presented with a PHP test page similar

Fig.7.22. The phpinfo() output page

to our example in Fig.7.22. This page is a really useful way of checking that everything is working. It's also worth printing this file out as you may need some of this information in the future. However, it is not recommended that you leave this page accessible (especially if/when you open up your server to the whole of the web) as this provides a great deal of information about your system. Therefore, once you're happy that it's all working, we'd recommend deleting the phptest.php file.

If you're a more advanced user or you have experience of web scripting you may also want to install additional server features at this point, such as MySQL for using databases. However, we will not be describing this process in this instalment of *Teach-In*.

3. Setting up IP addresses in your local network

In Part 4 of *Teach-In* we covered remotely accessing your Pi from within a home network and we discussed in some detail finding out and configuring your Pi's IP address. Essentially, each computer on your network has a unique number assigned to it known as the IP address. In order to route requests for webpages/scripts to our Pi web server we need to know the Pi's IP address at all times.

On most home networks your router will automatically assign the next available IP address to any device that you connect to it. This is called Dynamic Host Configuration Protocol or DHCP. DHCP is great because it allows you to quickly connect new devices without having to worry about setting up any network configuration. However, this does mean that the IP address of the Pi can change – and this is obviously a problem. Therefore, we need to make sure that it always gets the same IP address. There are several ways to do this, but the easiest way is to make sure that your Pi is always given the same IP address is to instruct your router to 'reserve' an IP address for it and always give this to the Pi when it requests to join the network. Most home routers have this facility built-in. You should follow the instructions for your specific device to do this as all routers have a slightly different setup process. However, this will generally involve logging in to the router through a web-based front end using a web browser, then finding the settings to 'reserve' or 'permanently lease' an IP address to the Pi.

It is also possible to setup your Pi's IP address manually and tell it not to request one from your router's DHCP server at all. Generally, you would set an IP address that is just outside of the range of addresses that the DHCP issues to avoid an IP address conflict. You would also need to know and set some of the other network settings manually. If you are an advanced user you may be used to this process and be aware of the information that you need. However, we will not cover this in detail here as we would recommend the first approach. There are many tutorials available for this on the Internet should you wish to follow this method.

4. Setting up routing to/from the web

Now we have our web server up and running and our Pi has a specific, static IP address on our network. If you are just looking to run scripts from within your local network then you can stop at this point. However, if you wish to make your web server available outside of your network on the Internet then we need to make a few tweaks to your router's settings to make sure that connections can be made right through to your Pi.

Your home router is really quite an advanced piece of kit. As well as managing your Internet connection with your ISP and managing your home network with a DHCP server (as we utilised earlier) it also protects your network from the outside world by providing a 'firewall' between the two. It does this by limiting what data can go in/out through various channels or ports. In this way the ports that are used for browsing the net, reading email or streaming media are open, but those that might be used by hackers to get in to your network for no good are blocked. In order for our Pi to communicate with the outside world we need to 'open up' one of these channels. The normal port for web page traffic is port 80. When we browse the Internet we make connections to web servers from our home network with this port. However, if we want to allow an external computer to request pages from our Pi server we need to make an incoming connection and we therefore need to ensure that our router does not block this.

We also need to tell the router that when it receives a request it knows where to route it within our network, and this is why we needed to make sure that our Pi always has the same IP address. The process of setting this up on a router is known as 'port forwarding' or 'port mapping'. As with setting up the IP addresses, this process is achieved by logging in to the admin console of your home router and you should follow the manufacturer's instructions to achieve this. The exact details and terminology used will depend on your router. You will need to setup a port mapping as follows:

External/incoming port	**80**
Internal/outgoing port	**80**
Internal host/IP Address	**[Pi's IP address]**
Protocol	**TCP**
Name	**Pi Web server***

** You can name the mapping as you wish in order to identify it.*

You may be asked to provide a port range or a start/finish port; if this is the case, set both to 80.

An example of the setup screen is shown in Fig.7.23 (this example is taken from a Huawei TalkTalk router). At this point we now have everything set up so that an external computer can request a page, but what's our web address? We'll look at this in the next step.

5. Setting up dynamic DNS

When we viewed pages on our web server from within the network we could just use the IP address of the Pi as the URL. However, that IP address only works within our own network, so we need to know how to get to our web server from the outside world.

Your home router is your connection to the wider network that is the Internet. Just as the computers on our home network need to have a unique IP address, so our router must have one on the Internet. When it connects to the Internet it is given an IP address by a DHCP server in just the same way as our router gives an IP address to our local computers. And just as (without the changes we made earlier) our devices get different IP addresses at different times, so our router's external internet IP address changes. Commercial web servers on the other hand have fixed, static IP addresses that never change so that they can always be found at the same place. This, of course, gives us a problem. We could simply find out what our external IP address is and then use it until it changes (and typically they do stay the same for some time) but potentially if the IP address changes then we would be without access until we could find out the new one – not an ideal solution.

Another option is to see if our ISP can provide us with a static IP address that never changes. This would solve our problem, but not all ISPs provide this service and those that do often only offer it at a

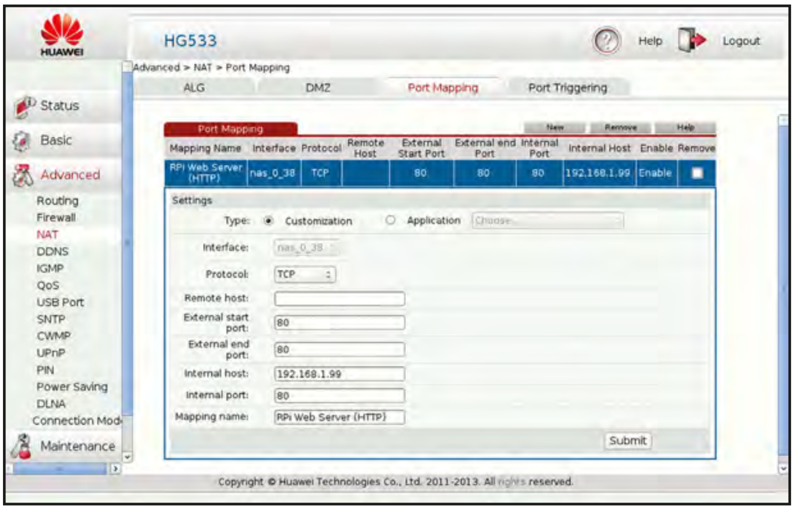


Fig.7.23. Example port mapping setting screen from a Huawei TalkTalk router

Fig.7.24. A test page being served to the world from our Raspberry Pi as viewed using Google Chrome on a Windows PC

premium cost and/or for business users only. Even if we did arrange to have a static IP address we'd still have to remember a horrible 12-digit number. What we need then is a way of getting a nice looking, easier-to-remember web address, and this is where dynamic DNS comes in.

When we browse the Internet we use text-based web addresses or URLs (universal resource locators). These are great because they're easy to remember and often relevant to what they are about; for example, the name of the company, product or service. They also have a suffix that can tell us where the site is located geographically. For example to go to Google's UK site we would go to **www.google.co.uk** – 'google' is the domain name (obviously linked to the name of the company) and '.co.uk' is the domain suffix for the UK.

Could you imaging browsing the Internet and having to remember the IP address for the server of each website that you like to visit? This is all possible because of Domain Name System or DNS for short. Put very simply, DNS is a massive phone book for the Internet, containing a database of all of the IP addresses and the domain names that link to them. A world-wide network of domain name servers keep these records and are constantly being updated. When we go to a website we actually look up the entry in this 'phonebook' and we are directed appropriately to the web server behind the scenes.

As we mentioned earlier, commercial web servers have an IP address that stays the same, but we know that ours can change. So if we had a conventional DNS entry this would go 'out of date' quickly. It would be a bit like changing your mobile phone number and not telling anyone. What we need then is to use a *dynamic* DNS service or DDNS. DDNS services allow you to have a domain name for which the IP address can be updated straight away as soon as your IP address changes. That way it's always up-to-date and you'll always be able to access your web server from the same easy-to-remember address. Fortunately, there are a number of DDNS services, many of which offer a basic service for free such as no-IP (**www.noip.com/free**). A quick search for 'dynamic DNS' will give you many options. Which you choose will

Fig.7.25. Our test page being viewed on an Android smart phone to test external access

depend on your own preference/budget and which services are compatible with your router (check the manufacturer's documentation).

When you register with a provider you will be asked to choose an address, usually a sub-domain of one of their own domain names – eg, **yourname.no-ip.biz**, as well as create a username/password login. These details are then entered into a dynamic DNS client (software that monitors your external IP address and sends updates to your dynamic DNS provider when it changes). Most routers support acting as a dynamic DNS client, so you would need to enter these details into the administration console of your router. If your router does not have dynamic DNS support there are dynamic DNS clients available for all platforms including Raspbian.

If you've managed to complete all of the steps successfully you now have a web server available across the world-wide web. You can test if it's all worked by visiting your server at your new address from an external device. A good way of testing externally without leaving your house is to use a mobile phone over cellular data (*not* connected to your Wi-Fi). Alternatively, there are services that will take a screenshot of your website from a remote location (often used by web designers to see what their sites 'look like' from other external browsers and operating systems). A good example of this is Browser Shots (**www.browsershots.org**) where you simply enter the URL and request screenshots from various external computers (note that there can be delay for the free service).

It should be noted that you may not be able to use your dynamic DNS address to access your web server from within your own network. This is known as 'local loopback' where you are effectively trying to send a web page request out to the web and then back in again. This behaviour is often blocked by routers as a security measure. Therefore, you'll need to use your local address from within your network and your dynamic domain name address from outside.

As we explained at the onset, setting up your Pi as a web server is relatively easy, but it does rely on getting all of the steps right and working together. Trying to troubleshoot where a problem is can be frustrating and we'd suggest making sure that each step is right before moving on to the next. To try and make the process a little easier we've created a checklist to follow as you prepare to 'go global' with your Raspberry Pi.

So, whether you have visions of running your own website from your front room or creating a next generation web-controlled robot drone, our guide will hopefully help to get you started and get your Pi on to the web.

In next month's Teach-In with Raspberry Pi

In next month's *Teach-In 2014*, our *Pi Class* introduces stepper motors and our *Pi Project* features the construction of a complete stepper motor controller. Our *Home baking* feature shows you how to connect a camera to the Raspberry Pi and how you can use this as the basis of a complete Internet-connected webcam application.

Pi Web server Checklist

Follow these steps to get your Raspberry Pi web server working:

1. Use `sudo apt-get update` then `sudo apt-get install apache2` to install Apache web server software.

Check: Visit `http://127.0.0.1/` on your Pi's web browser

2. Install PHP server-side scripting language using `sudo apt-get install php5`

Check: Create a `phptest.php` file that calls the `phpinfo()` test function and is saved in the root directory (`/var/www`). Open `http://127.0.01/phptest.php` on your Pi's web browser

3. Setup your Pi with a static IP address by either setting up a port reservation in your router settings or manually changing the Pi's network settings (advanced).

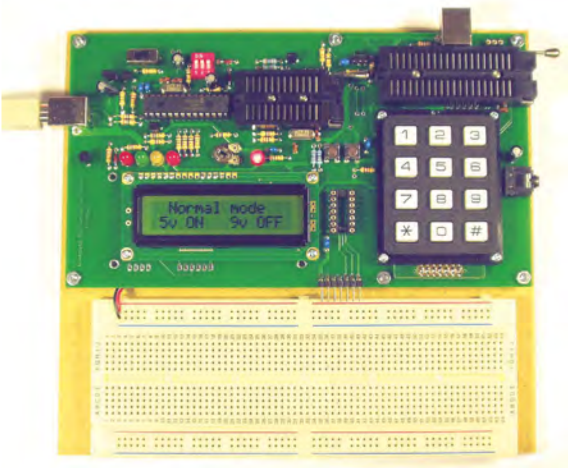
Check: Visit the web server and PHP test page on another device on your network using the URLs in step 1 and 2 but replacing the `127.0.0.1` with your Pi's new static IP address

4. Create a port mapping on your router from/to port 80 pointing towards your Raspberry Pi's IP address.

5. Register with a dynamic DNS provider that's compatible with your router and enter the settings in to your router.

Check: Visit your Pi web server at your dynamic domain name address from a device outside of your network.

PIC Training Course



P931 Course £148

The control PIC of our programmer has two modes of operation, its normal programming mode, and a USB to USART mode. Programme your PIC in the usual way then flip the red switches and your PIC can use the control PIC as a serial link to your PC. All designed to make the learning process as straightforward as possible.

The course follows the same well proven structure. We begin learning about microcontrollers using the incredible value 18 pin PIC16F1827. At the heart of our system are two real books which lie open on your desk while you use your computer to type in the programme and control the hardware. Start with four simple programmes. Run the simulator to see how they work. Test them with real hardware. Follow on with a little theory.....

Our PIC training course consists of our PIC programmer, a 318 page book teaching the fundamentals of PIC programming, a 304 page book introducing the C language, and a suite of programmes to run on a PC. Two ZIF sockets allow most 8, 18, 28 and 40 pin PICs to be programmed. The programming is performed at 5 volts then verified at 5 volts and 2 volts or 3 volts.

P931 PIC Training & Development Course comprising.....

USB powered 16F and 18F PIC programmer module
+ Book Experimenting with PIC Microcontrollers
+ Book Experimenting with PIC C 6th Edition
+ PIC assembler and C compiler software on CD
+ PIC16F1827, PIC16F1936 & PIC18F2321 test PICs
+ USB cable. £148.00

(Postage & insurance UK £10, Europe £20, Rest of world £30)

Experimenting with PIC Microcontrollers

This book introduces PIC programming by jumping straight in with four easy experiments. The first is explained over seven pages assuming no starting knowledge of PICs. Then having gained some experience we study the basic principles of PIC programming, learn about the 8 bit timer, how to drive the liquid crystal display, create a real time clock, experiment with the watchdog timer, sleep mode, beeps and music, including a rendition of Beethoven's *Fur Elise*. Then there are two projects to work through, using a PIC as a sinewave generator, and monitoring the power taken by domestic appliances. Then we adapt the experiments to use the PIC18F2321. In the space of 24 experiments, two projects and 56 exercises we work through from absolute beginner to experienced engineer level using the very latest PICs.

Experimenting with PIC C

The second book starts with an easy to understand explanation of how to write simple PIC programmes in C. Then we begin with four easy experiments to learn about loops. We use the 8/16 bit timers, write text and variables to the LCD, use the keypad, produce a siren sound, a freezer thaw warning device, measure temperatures, drive white LEDs, control motors, switch mains voltages, and experiment with serial communication.

Web site:- www.brunningsoftware.co.uk

Serial Coms Extension £31

This third stage of our PIC training course starts with simple experiments using 18F PICs. We use the PIC to flash LEDs and to write text to the LCD. Then we begin our study of PC programming by using Visual C# to create simple self contained PC programmes. When we have a basic understanding of PC programming we experiment with simple PC to PIC serial communication. We use the PC to control how the PIC lights the LEDs then send text messages both ways. We use Visual C# to experiment with using the PC to display sinewaves from simple mathematics. Then we expand our PC and PIC programmes gradually until a full digital storage oscilloscope is created. For all these experiments we use the programmer as our test bed. When we need the serial link to the PC we flip the red switches to put the control PIC into its USB to USART mode.

In the second part of *Experimenting with Serial Communications* 4th Edition we repeat some of the serial experiments but this time we use a PIC18F2450 with its own USB port which we connect directly to a USB port of your PC. We follow this with essential background study then work through a complete project to use a PIC to measure temperatures, send the raw data to the PC, and use the PC to calculate and display the temperature.

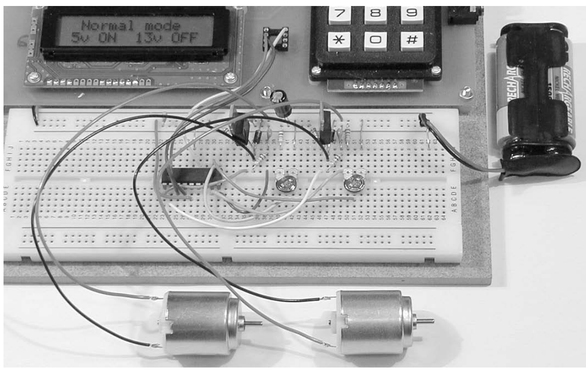
290 page book + PIC18F245 test PIC + USB lead.... £31

P942 Course £173

This has the same books and features as the P931 course. The P942 programmer/development module can be powered from a separate PSU (programming verified at 5.5 volts, 5 volts and 2 or 3 volts) or powered from USB (programming verified at 5 volts and 2 volts or 3 volts). The P942 can programme 3.3 volt as well as 5 volt 16F and 18F PICs, and has an RS232 port as well as the USB port for experimental use. See website for details.

Ordering Information

Our P931 and P942 programmers connect directly to any USB port on your PC. All software referred to operates correctly within Windows XP, NT, 2000, Vista, 7 and 8 etc. Telephone for a chat to help make your choice then go to our website to place your order using PayPal, or send cheque/PO, or request bank details for direct transfer. All prices include VAT if applicable.



White LED and Motors

Our PIC training system uses a very practical approach. Towards the end of the PIC C book circuits need to be built on the plugboard. The 5 volt supply which is already wired to the plugboard has a current limit setting which ensures that even the most severe wiring errors will not be a fire hazard and are very unlikely to damage PICs or other ICs.

We use a PIC16F1827 as a freezer thaw monitor, as a step up switching regulator to drive 3 ultra bright white LEDs, and to control the speed of a DC motor with maximum torque still available. A kit of parts can be purchased (£31) to build the circuits using the white LEDs and the two motors. See our web site for details.

Mail order address:

Brunning Software

138 The Street, Little Clacton, Clacton-on-sea, Essex, CO16 9LS. Tel 01255 862308

PIC n' Mix

Mike Hibbett

Our periodic column for PIC programming enlightenment

Goodbye MPLAB, Hello Kickstarter

THIS month, we say goodbye to Microchip's MPLAB IDE, look towards the future with the Kickstarter phenomenon and tie both into a review of where we are with our development board.

So it is finally time to say farewell to MPLAB, and the C18 compiler. Microchip have decided it is time for us to migrate to the MPLAB-X development tools, and have removed all of the software downloads relating to MPLAB from their website.

It was inevitable, and the writing has been on the wall for several years, but that doesn't make it easier to accept. MPLAB wasn't perfect, but it had suited many of us over the years. This is the problem with proprietary software – it may be free, but you have no control over its continued development. If MPLAB was open source software, someone would continue to develop it, continue to add new device support.

So we must move on. We were going to discuss installing the MPLAB C18 compiler this month; instead, we will be looking at installing both a new IDE (MPLAB-X) and a new compiler (XC8)

Development board

Let's do a quick review of where we are. We have a board which is small, cheap and easy to build; we have some template source code to enable us to bring a new design to life quickly. And we have made a simple kitchen timer to demonstrate tying all the parts together.

The board can be made cheaply, although you may end up having to buy enough components to build five boards just to avoid delivery charges! We have a solution to that, which we will mention in a minute.

One of the main outstanding issues with the project so far, however, is to do with the use of assembly language. It's a very efficient language, both in terms of code size and speed, but it is not the easiest language to work with. While it is possible to create easy to read, easy to maintain software in assembler, it's very difficult and time consuming. And not much fun.

The 'C' language

Writing in a higher level language such as BASIC or C, while requiring additional learning time, will result in you being able to implement more functionality, in less time, with less effort. Writing software becomes fun rather than a challenge.

There are a number of languages that we could have moved to (C, C++, Lua, BASIC, Forth to name the more obvious ones.) We have decided to focus on the C programming language rather than any other for a number of reasons that not everyone will agree with the choice of programming language for any given task, it can become almost a religious argument. We made this decision based on:

Efficiency - The language produces very efficient code, in both size and speed of execution.

Ease of use C - is not a very complicated language, and there is not much to it. Books on C are *thin*.

Availability - A compiler is available, for free. You do not need to buy or install any other software.

Our focus over the next few months will be moving over to MPLAB-X, picking up programming 'C', and exploring some cool new peripherals.

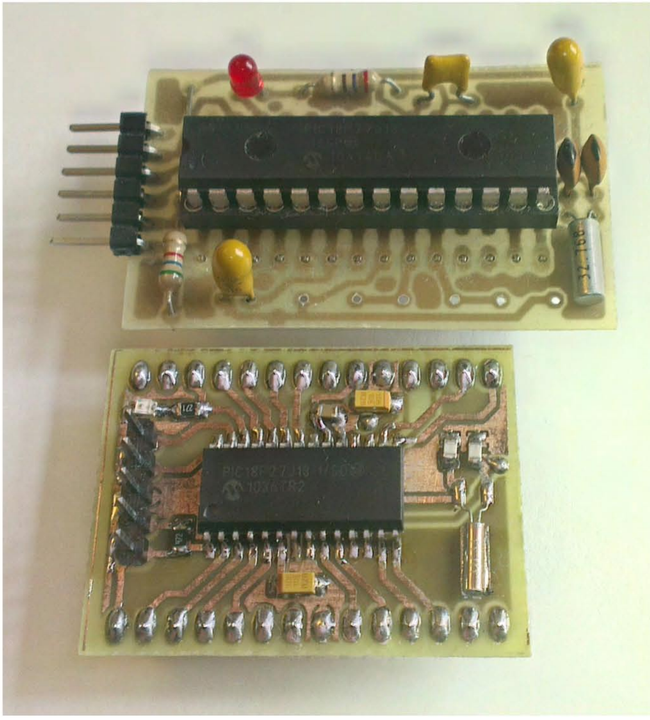


Fig. 1 Surface-mounted component version

Power

One of the questions that has come up recently is about powering the board. It will run with a supply of 2.0 to 3.6V, making it ideal for two alkaline batteries. However, what if you want to power it from rechargeable batteries? They have a lower terminal voltage, 1.2V, and so 'start off' at a lower voltage.

Most nickel-metal hydride (NiMH) batteries are actually quite powerful and will perform as well as many alkaline batteries, but there is a simple solution – get a three cell battery holder. This will give you a full charge voltage of 3.6V, matching the maximum normal working voltage of the processor. In our opinion, for battery power applications, the three-cell NiMH solution is the best, and will give you maximum battery life for least cost (alkaline batteries are one of the most expensive forms of energy you can purchase!)

While our design uses through-hole components for ease of assembly, the design works just as well with surface mount components. This is actually our preferred format, as there is less chance of damaging components when the board is transported. It also results in a more compact board. An example prototype which we hand built is shown in Fig.1. The size of the board was kept to a minimum by not including holes for mounting the board; instead, we rely on the board being glued in place. It's light enough and the reverse of the board is free of components making it ideal for fixing with hot-melt glue.

Kickstarter

A number of readers have asked whether we are going to make the development board available to purchase. At the moment we don't, and when thinking about how we might make it available, we realised it would require at least one hundred people to be interested for us to be able to bring the cost low enough to make it interesting. What would be the best way to present the idea, and get the interest and commitment necessary? Then the penny dropped: this is an ideal opportunity to test out the Kickstarter process.

Kickstarter, for those who have yet to to discover it, is a website that was established in 2009 with the aim of helping people bring creative projects to life. It enables people to present their idea and gain backing from interested parties. The website provides a standardised platform for detailing the project and supplying details of what returns people will get for whatever 'pledge' they make. The process is called 'crowdsourcing', and Kickstarter were the first to make it popular. Since 2009 they have helped bring over 50,000 projects to life.

So how will this work for us? While the details of how Kickstarter operate are best discovered from their website at **www.kickstarter.com**, the principle is that we will offer a surface-mount version of the 'Low Power, Low Cost' PIC development board available for a pledge of £9.95, inclusive of delivery cost. The board will be professionally manufactured, with assembly in the UK and Ireland. Tutorials and template projects will be made available at the authors website, **mjhdesigns.com**. There will be a number of different pledge options, including one for early access to prototype hardware, and sponsorship by providing space on the back of the board for a name or web address. To achieve the price, we will need at least 100 pledges; any less than that, and the project will not go ahead. The pledge value will only be deducted from your card if the project goes ahead.

We will publish the results of running the Kickstarter campaign later in the year. Until then, why not help make it happen by signing up to one of the pledges. You can find us on Kickstarter by search for 'Low Power, Low Cost Development Board'. The campaign will run from early March until the end of April, at which point – should we have received enough support – we will start supplying boards.

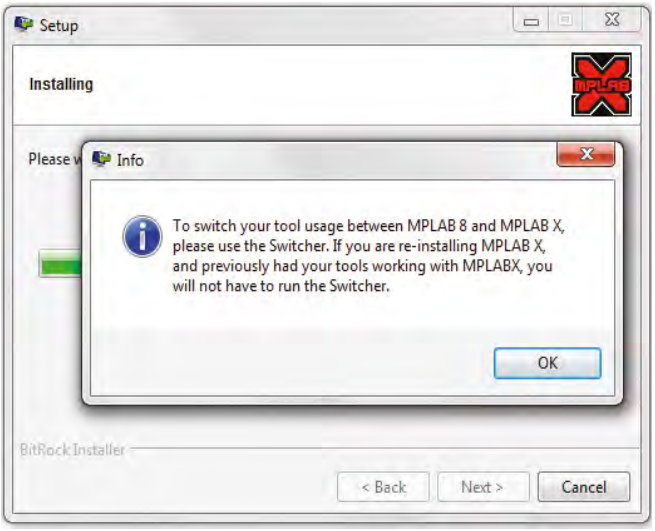


Fig. 2 Switcher dialog

MPLAB-X installation

Let's return to the software development environment for our development board. To support the 'C' programming language we must ditch MPLAB and install instead the new IDE, MPLAB-X.

We start by installing the MPLAB-X IDE, and then the XC8 C compiler. Both are available as a free download from the Microchip site; MPLAB-X from **www.microchip.com/pagehandler/en-us/family/mplabx/**, XC8 from **www.microchip.com/pagehandler/en-us/devtools/mplabxc/home.html**. These programs are now supported on Linux and Mac OSX in addition to Windows. For the purposes of these articles we will follow a Windows installation path. Together, these two files weigh in at over half a gigabyte in size, zipped. One has to wonder what they put in there.

Double click on the **mplabx** zip file, and then double click on the .exe file within it. Click 'Accept' to the licence agreement, and then 'Next', to allow the program to install to the default location.

When installation completes, you are presented with an unusual message, as shown in Fig.2. This indicates that MPLAB-X does not play nicely when installed alongside MPLAB; we will have to remember this should problems occur later. Closing the dialog reveals the final installation dialog message. Clear the tick box requesting that you download the compiler (we already did this ourselves) and click on 'Finish'. At this stage, three icons will have been placed on your desktop.

Now run the second file that we downloaded from the Microchip website, **xc8-v1.30-win.exe** (the version text, v1.30 in our case, may have changed by the time you come to download it.) Click 'Next', followed by the inevitable 'I accept', 'Next' to the licence terms, then 'Next' to each dialog that follows, accepting the default settings. At the end of installation click 'Next' to accept the default, free licence, and then 'Finish'.

Fig. 3 MPLAB-X start-up screen

Starting up

Let's fire 'er up. Double click on the 'MPLAB X IDE' icon on the desktop (*not* the 'MPLAB IPE' – that's the programming tool.) After a few seconds the welcome screen is displayed, as seen in Fig.3. There are a few 'Minute Video' tutorials available from this window, and it is worth watching the first one to get familiar with the system.

Next month

Next month we re-create out project template files, this time in 'C'. We will revise the kitchen timer; it will be interesting to see what the differences in code size will be. Watch this space!

Not all of Mike's technology tinkering and discussion makes it to print. You can follow the rest of it on Twitter at @MikeHibbett, and from his blog at mjhdesigns.com


THE No 1 UK MAGAZINE FOR ELECTRONICS TECHNOLOGY & COMPUTER PROJECTS
EPE EVERYDAY PRACTICAL ELECTRONICS
www.epemag.com
USB MULTI-INSTRUMENT PROJECT
PC controlled
2-channel digital scope
Spectrum analyser
DMM and frequency counter
Audio function generator
WIN A MICROCHIP REMOTE CONTROL DEMO BOARD WITH ZENA WIRELESS ADAPTOR
CLASSiC-D AMPLIFIER
CONSTRUCTING OUR HIGH-POWER HIGH-PERFORMANCE AMP
Teach-In 2014 Raspberry Pi – Part 3
EPE's comprehensive guide to Raspberry Pi
DEC 2013 £4.40
PLUS: NET WORK, READOUT, CIRCUIT SURGERY, TECHNO TALK, PIC N' MIX & INTERFACE

INTERFACE

By Robert Penfold

Using the Raspberry Pi camera

THE previous *Interface* article covered the basics of using edge-triggered interrupts with the Raspberry Pi, and utilising the tiny camera add-on with the CSI port. In this article things are taken a stage further, with circuits that use the interrupt facility and the camera add-on (Fig.1.) to automatically take still pictures or videos. An essential first step to this end is to find a way of operating the camera under software control.

As pointed out in the previous *Interface* article, there is some supporting software available for the camera, but this is a command line utility. This makes it possible to take pictures and videos by typing commands into the LXE terminal program, and some control over the camera's features is also available via this route. However, this software does not act as a normal add-on for Python, and as yet there is no Python add-on to give control of the camera. Fortunately, in common with most programming languages, Python does have facilities for calling and executing operating system commands. This method provides a slightly circuitous but effective way of controlling the camera from a Python program.

It will be assumed here that the Raspberry Pi has up-to-date versions of the operating system and supporting software installed, and that the camera and supporting software has also been installed, as described in the previous *Interface* article. Neither the camera nor the interrupt method of interfacing to the GPIO port will operate unless suitably recent versions of the operating system and software modules are installed.

Getting the call

There are at least two ways of calling commands from a Python program, and probably more than that. Matters are complicated slightly by the need to pass a parameter or two to the `raspistill` or `raspivid` commands in order to get them to do anything worthwhile, but it still seems to be quite easy to obtain the desired result. The method shown in Listing 1 was used simply because it was the first one I tried that actually gave the desired result. It uses the 'call' method to access the command line, but the 'os' (operating system) method seems to work equally well.

The first five lines of the program set up the GPIO port to use pin 26 of the port as an output, and they also import the `Call` instruction. The latter is used later in the program to access and control the `raspistill` instruction, and take the picture. Next, pin 26 is set high, and then a `Call` instruction is used to activate the `raspistill` command and pass it the basic parameters needed in order to take a photograph. Then pin 26 is set low and 'photo taken' is printed on the screen. I used an LED and a 1k series resistor on pin 26 of the GPIO port to indicate that the taking of a photograph was in progress, but this additional hardware and corresponding parts of the program are not essential.

Listing 1

```
import RPi.GPIO as GPIO
from subprocess import call
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(26, GPIO.OUT)
GPIO.output(26, GPIO.HIGH)
call (["raspistill -o photo1.jpg"], shell=True)
GPIO.output(26, GPIO.LOW)
print ("photo taken")
GPIO.cleanup()
```

By default, `raspistill` provides a five-second delay before a photograph is taken. This delay normally enables the user to quickly compose the shot so that the camera takes a picture of the right thing! With automatic operation of the camera, this delay will not normally be required. Instead, the camera must be aimed in what is (hopefully) the right direction, and the picture must be taken as soon as the system is triggered. The delay can be altered using the *–t* parameter, and the figure used here specifies the delay in milliseconds. The preview facility normally gives a digitally zoomed view, but it can be forced to the full screen mode using the *–f* modifier, or it can be removed completely using the *–n* option.

The camera has an *f* 1:2.9 3.6mm lens that gives a wide angle of view. The actual angle of view is something in the region of 53 by 40 degrees, which means that in most cases you do not have to aim the camera very accurately in order to get the subject fully within the frame. It also means that the camera will need to be relatively close to the subject when you do require a frame-filling shot.

Time doesn't fly

It is only fair to point out that it takes a short but significant time for a photograph to be taken even if a delay time of zero is used. The camera is not restricted to photographing fairly distant subjects, and it worked well at short distances when I tried a simple test shot (Fig.2.). For such a simple camera the results are surprisingly good. It seems to be able to cover a wide range of light levels and colour temperatures. The delay when taking a photograph is presumably due to the camera having to make exposure and other adjustment prior to taking and storing an image. Obviously, it is not well suited to catching high speed action and fleeting events, where the lack of an electronic flash option would probably rule out its use anyway.

With the camera under software control it is possible to do such things as time-lapse photography and automatic triggering using light or sound detectors, infra-red sensors, and so on. It is automatic triggering that will be considered here. The software for this can be extremely simple indeed, and the short program of Listing 2 will suffice.

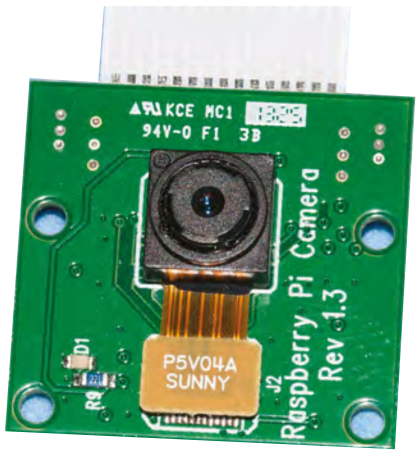


Fig.1. Although tiny, the Raspberry Pi camera module is more sophisticated than one might expect. It has three metering modes and variable ISO sensitivity

Listing 2

```
import RPi.GPIO as GPIO
from subprocess import call
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(26, GPIO.OUT)
GPIO.setup(24, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.wait_for_edge(24, GPIO.BOTH)
GPIO.output(26, GPIO.HIGH)
call (["raspistill -t 0 -o photo1.jpg"], shell=True)
GPIO.output(26, GPIO.LOW)
print ("photo taken")
GPIO.cleanup()
```

This is much the same as Listing 1, but it has a couple of extra lines that first set up pin 24 as an input having an internal pull-up resistor, and then set it for operation as an interrupt input that will detect high-to-low or low-to-high transitions. The program will therefore halt when it reaches this line and wait for a transition on pin 24. An ordinary push-button switch connected from pin 24 to the 0V supply rail is all that is needed to test the program. On pressing the switch, the program should move on and take the picture as before.

Fig.2. The camera can focus quite close, but with its wide-angle lens it still covers a relatively large area. This shot was taken at a distance of only about 300mm from the subject

Seeing the light

A number of sensors of various types have been described in previous *Interface* articles, and it should be quite easy to modify any of these to operate with the Raspberry Pi. Fig.3 shows the circuit diagram for a light-change sensor that can be used to trigger an input of the Raspberry Pi's GPIO port. It would presumably be possible to obtain a similar function by getting the camera to take pictures periodically, with the image data being checked for significant changes from one picture to the next. On the other hand, using additional hardware and interrupts probably gives a quicker response time and is potentially more sensitive as well.

The circuit of the light sensor is essentially the same as one featured previously in this series. Phototransistor TR1 is the light sensor, and its leakage level increases and decreases in response to changes in the received light level. This produces small changes in the voltage at its collector (c) terminal, and these are amplified by a simple common-emitter stage based on TR2.

The output signal of TR2 is coupled by C3 to a window discriminator based on the two operational amplifiers (op amps) in IC1. Under standby conditions, the input of the window discriminator is biased to half the supply potential by R4 and R5. If it is taken above two thirds of the supply potential by the signal from TR2, the output of IC1a goes high and switches on TR3. The output of IC1b goes high and switches on TR3 if the signal from TR2 takes the input of the window discriminator below one third of the supply voltage. The circuit will therefore detect any sudden increases or reductions in the received light level.

In the original circuit, TR3 had a collector load resistor and was powered from a 5V supply. In this case, no external load resistor is needed because the program of Listing 2 sets pin 24 to operate as an input having an integral pull-up resistor. The collector of TR3 can therefore be connected to pin 24 of the GPIO port and it will pull this input low when the sensor is activated. As pointed out in previous *Interface* articles, the inputs and outputs of the GPIO port operate at 3.3V and not 5V. It is important not to directly drive the inputs from logic circuits that operate at 5V or more, since this could cause a malfunction or even damage the chip at the heart of the Raspberry Pi.

All pull together

This open-collector method of driving a GPIO input, in conjunction with an internal pull-up resistor, provides a simple means of ensuring that the input is driven at the correct signal voltages. It does not matter whether the main circuit operates at 3V, 30V, or anywhere in between. With this method, the input always receives an adequate high potential, and it can go no higher than the 3.3V supply used for the internal pull-up resistor.

Normally, it is not acceptable to have several outputs driving a single input unless a suitable gate circuit is used. However, with the open-collector method there is no problem in having several outputs driving a single input. The transistors driving the input form a simple NOR gate, and the input will be pulled low if any of the transistors are switched on.

It should be noted that the CA3240E has a PMOS input stage, and it requires the usual anti-static handling precautions. Virtually any phototransistor for use in the visible light spectrum should suffice for TR1. Since no connection is made to its base terminal, devices that do not have the base connection accessible are suitable for use in this circuit. A photo-resistor would probably work quite well, but some of these components are quite slow in operation and might give reduced performance. Any high gain silicon npn transistors can be used for TR2 and TR3.

The light change sensor requires a supply potential of about 9V to 15V, but the Raspberry Pi's GPIO port can only provide 3.3V and 5V supplies. The sensor circuit can either have its own power source, such as a 9V battery, or it could be powered from the GPIO port's 5V supply via a suitable DC-to-DC converter. The current consumption of the circuit is only a few milliamps.

Feeling the heat

The circuit of Fig.4 is essentially the same as the passive infrared detector

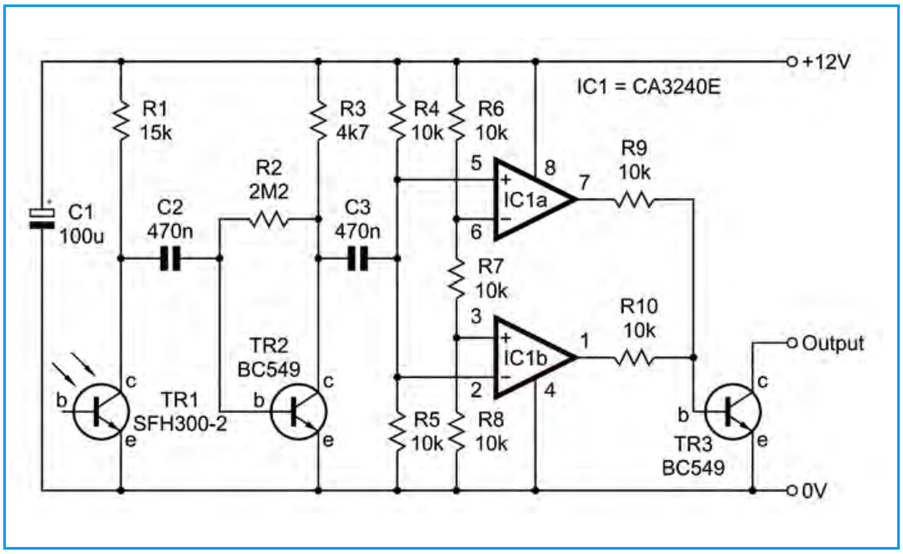


Fig.3. The light change detector circuit. Although it operates from a 12V supply, its open collector output stage can safely drive a 3.3V logic input of the GPIO port

featured in a recent *Interface* article. It is designed to detect the body heat of anyone passing in front of the sensor. IC1 is a passive infra-red detector that is given a fairly narrow angle of view so that anyone passing in front of it produces a small change in its output voltage. This voltage is fed to a high gain amplifier based on TR1 and TR2, and then to a level detector that utilises operational amplifier IC2. VR1 is adjusted for an output voltage that is just low enough to take the output of IC1 low and switch off TR3.

When the circuit is activated, the voltage at the collector of TR2 will go below this threshold voltage on negative half cycles, the output of IC2 will go high, switching on TR3 and activate the interrupt input of the GPIO port. Again, the output stage of the circuit has been modified to suit a GPIO input that has a built-in pull-up resistor.

It is not necessary to use two inputs and modified software in order to use the light and heat detector circuits together. Due to their open collector output stages they can both drive a single GPIO input.

Of course, the sensors featured here do not have to be used in conjunction with the Raspberry Pi camera, and they could be used to activate any software routine and add-on hardware. When used with the camera they should work equally well using `raspivid` to capture video clips. If you do obtain the camera it is worthwhile downloading the online documentation for it, which details the various controls that are available via the camera's software module. The camera is more sophisticated than one might expect. It can, for example, be set to use evaluative, spot, or centre-weighted metering. The software has some useful tricks, such as the ability to do time lapse photography.

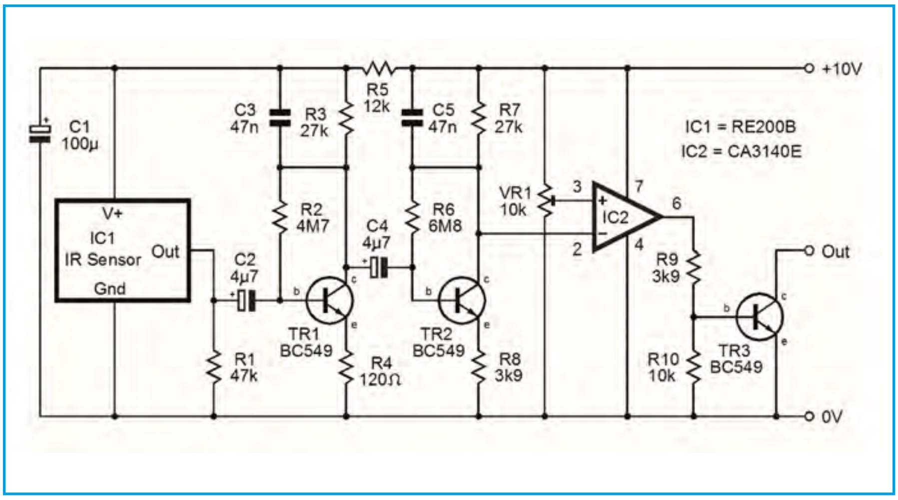


Fig.4. Like the light change detector, this passive infrared detector has a modified output stage that can drive a GPIO input. The light and infrared circuits can be used to drive a single input of the GPIO port, with the two output transistors acting as a simple NOR gate

Max's Cool Beans

By Max The Magnificent

Welcome to the Pleasure Dome!

Things have been racing along here in the 'Pleasure Dome' (my office). In fact, so many things have been happening that I don't really know where to start. Much of what I've been doing has been focused on my ongoing Arduino-powered *Pedagogical and Phantasmagorical Inamorata Prognostication Engine* project (try saying that ten times quickly). I really will have to show you a photograph of this one day, but for the moment I will just have to describe it.

His 'n' hers

In fact, there are two main units. The *Inamorata Prognostication Engine*, which reflects the female portion of this project, will be the larger unit on the bottom. This is going to be housed in a beautiful 1929 wooden radio cabinet I have here in my office. The main control panel on the center will be formed from aged brass, which is to be festooned with antique switches and knobs and buttons and analogue meters. There are also going to be over 100 tri-colored LEDs, which are going to be mounted behind small mother-of-pearl 'dots' embedded in the panel. In addition to numerous knobs and switches, five potentiometers will be used to specify things like 'mood,' 'personality,' and so forth. The idea is to manipulate all of the controls in such a way as to move the main analogue meter (whose scale will be marked 'Extremely Disgruntled' at one end and 'Fully Gruntled' at the other) into its 'green' zone. The problem is that there are no instructions (any textual annotations will be presented in the form of Elvish script), and the various controls will work in a somewhat random and erratic fashion. The end result is that you might get tantalisingly close, but never actually...

Motorised pots

One of the cool aspects to this is the potentiometers, which are motorised. Thus, if someone were to try to modify one of the settings by turning any of the potentiometers, the others will automatically adjust themselves to compensate. Sometime later, when no one is looking, they will return themselves to their original positions. One of the analogue meters will reflect the current temporal proximity to the next full moon (not a good time). Another will reflect the time to the next blue moon, which occurs approximately once every three years or so (one's chances of moving the main meter into its 'green zone' are slightly improved on the evening of a blue moon).

But we digress... There's also going to be an *Ultra-Macho Prognostication Engine*, which reflects the male portion of this project. This will be presented in an antique-looking box that sits on top of the *Inamorata Prognostication Engine*. It will be a much more straightforward affair. I've acquired the most amazing antique single-pole, dual-throw Bakelite knife switch and I've reserved my biggest and best antique analogue meter. There will also be a light behind a large mother-of-pearl 'window.' The thing here is that, irrespective of the positioning of the switch, the meter will always be in its 'green' zone and the light will always be illuminated (the meter will of course wander around a bit and the light will vary slightly in intensity, just to provide some visual interest).

There are so many other aspects to this project, such as the fact that the main engine will be equipped with proximity sensors, so the meters will 'twitch' and the LEDs will flicker when someone approaches. There are also going to be some mega-cool sound effects. If you toggle a switch, for example, you will hear sounds like ball bearings rolling down slopes and mechanical mechanisms activating and so forth. I don't know if it's obvious, but I'm having a humongous amount of fun with this

Tubes and LEDs

But none of this is what I wanted to tell you about. I decided that for additional visual effect, it would be a good idea to have another brass panel mounted flat on the top of the *Ultra-Macho Prognostication Engine.* Sticking out of this plate will be some old vacuum tubes. My original plan was to use regular-sized tubes

The project's huge vacuum tube – do watch the YouTube link to see how I will illuminate this beauty!

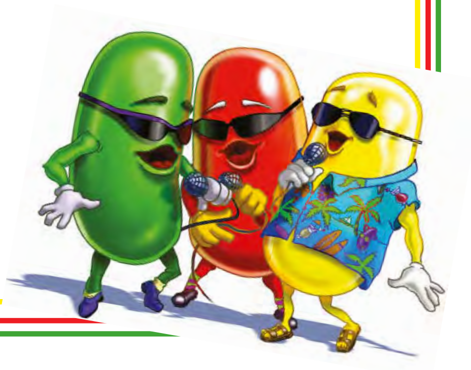
and to light them from below using tri-colored LEDs, which would also be controlled by one of my Arduinos. But then, while rummaging around at my local electronics emporium, I discovered a box of the most magnificent and voluptuous vacuum tubes I've ever seen. These bodacious beauties range from 8 to 10 to 12 inches tall.

The only problem is that they have metal bottoms, which will prevent me from mounting my tri-colored LEDs underneath. As it turns out, this isn't going to be a major issue. I've been experimenting with the NeoPixel Strip (**http://bit.ly/1bCuheV**) I acquired for my *Infinity Mirror* project. I'm going to use a 3D printer to create pseudo metal bands that go around the bottom of the tubes. Inside these bands will be NeoPixel strips. I only just started to experiment with this. Take a look at a video I posted on YouTube (**http://youtu.be/fEiMlxZCVpA**). All I'm doing in this video is lighting one LED after the other, but the result is pretty spectacular – it appears as though the structures in the tube are spinning around.

Can you imagine what a whole bunch of these tubes are going to look like on top of my *Ultra-Macho Prognostication Engine*? I can think of all sorts of effects to play with, including the fact that the display will change as someone approaches the engine.

Feedback please!

I'd love to hear what you think about all of this. Just send a letter to the editor and he will pass it on to me (he may even publish it in *EPE*). Until next time, have a good one!



BY IAN BELL

MOSFET basics – Part 2

LAST month, we started looking at the basics of MOSFETS following a request for such information from ***basementboy*** in a thread on *EPE's Chat Zone*. We looked at the physics of the MOSFET and using a very rough analysis obtained some equations describing its operation. We will start with a brief recap of this and then look at some MOSFET characteristics and basic circuits.

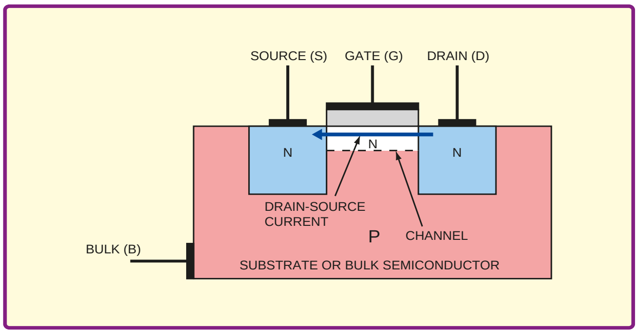


Fig.1. Cross section of N*-Channel MOSFET*

Basic structure

The cross section of an *N*-channel MOSFET is shown in Fig.1. Strictly speaking, the device has four terminals: the gate, source, drain and bulk. The bulk and source are commonly connected together within the device structure, leaving the familiar three terminal discrete components (with gate, source and drain). Within some integrated circuit designs the voltage on the bulk on some transistors is controlled independently to achieve particular functions or performance advantages.

The simplest view of MOSFET operation is as a switch with the gate voltage controlling conduction from source to drain (on or off). With a small, zero, or negative gate voltage (with respect to source/bulk) the *N*-channel MOSFET will not conduct from source to drain in either direction. This is because the *PN* diode junctions between source and bulk and drain and bulk will be surrounded by a non-conductive depletion region (like a reverse-biased diode). This non-conducting mode of operation is called the cut-off region. Similarly, a *P*-channel MOSFET will be cut-off with small, zero, or positive gate voltage.

The gate of a MOSFET is a capacitor. This means that no current flows into the gate if its voltage remains constant and often the gate can simply be assumed to be an open circuit. However, in high frequency amplifiers and high speed switching circuits gate capacitance may be significant.

Gate voltage

Applying a positive voltage to the gate of an *N*-channel MOSFET (with respect to source/bulk) will attract electrons to a region directly under the gate. If the density of these electrons is sufficient, they will outnumber the holes from the *P*-type bulk semiconductor transforming it into *N*-type. At this point, we have an *N*-type channel from source to drain (see Fig.1) which bypasses the *PN*-junction depletion regions and allows conduction between source and drain. The gate voltage at which the conducting channel forms is called the threshold voltage, V_T. Similarly, a *P*-channel MOSFET will conduct with negative gate voltage (with respect to source/bulk) which is greater than its threshold.

Typically, for an *N*-channel device, the source is a 0V and positive voltage, V_{DS}, is applied to the drain. For low values of V_{DS} the MOSFET acts a voltage-controlled resistor with increasing gate voltage resulting in lower drain-source resistance. This resistance is not linear – the source-drain current, I_{DS}, partly depends on the square of V_{DS} – however, for very low V_{DS} values it is close to linear (I_{DS} proportional to V_{DS}). This mode of MOSFET operation is called the linear region. For a *P*-channel device the source to drain voltage is negative.

When using a MOSFET we think of the gate voltage as single value, however, the operation of the device depends on the voltage difference between the gate and bulk semiconductor/channel at each point along the channel. If the source and drain voltages are not equal then the gate-to-channel voltage will vary along the channel. If the gate-to-channel voltage is less than the threshold voltage (with V_{GS} above threshold) then a portion of the channel will be pinched off.

Pinch off does not prevent conduction, but it does prevent further significant increase in drain current with increasing drain-source voltage. To a basic approximation, the drain source current (at a given V_{GS} over threshold) becomes a constant value, independent of V_{DS} for V_{DS} greater than (V_{GS}–V_T). In this mode of operation, called the saturation region, the MOSFET behaves like a constant-current source, with the current set by the gate-source voltage. In reality I_{DS} tends to increase a little with increasing V_{DS}.

Fundamental MOSFET equations

Last month, using a very rough analysis of the device physics, we obtained the equations for device operation in the linear and saturation regions. Adding the somewhat trivial expression for cut-off, we get a third equation and full coverage of device operation.

Cut-off region: V_{GS} less than V_T

$$I_{DS} = 0$$

Linear region: V_{DS} less than (V_{GS}–V_T)

$$I_{DS} = \beta\left[(V_{GS} - V_T)V_{DS} + \frac{V_{DS}^2}{2}\right]$$

Saturation region: V_{DS} greater than (V_{GS}–V_T)

$$I_{DS} = \frac{\beta}{2}(V_{GS} - V_T)^2$$

These equations relate the drain-source current, I_{DS}, to the gate-source voltage, V_{GS}, and drain-source voltage, V_{DS}, applied to the device. The equations also include the threshold voltage, V_T, and a transistor gain parameter β. The values of V_T and β are determined by the physical structure/size of the device, properties of the materials used to form the device, temperature and fundamental physical constants. A more detailed analysis of the device physics than we performed last month can obtain equations for V_T and β in terms of the device's physical parameters.

To get a better feel for these equations it helps to plot them on a graph or two. We will simply plot the equations, without trying to make them fit a specific device, although the values will be in a

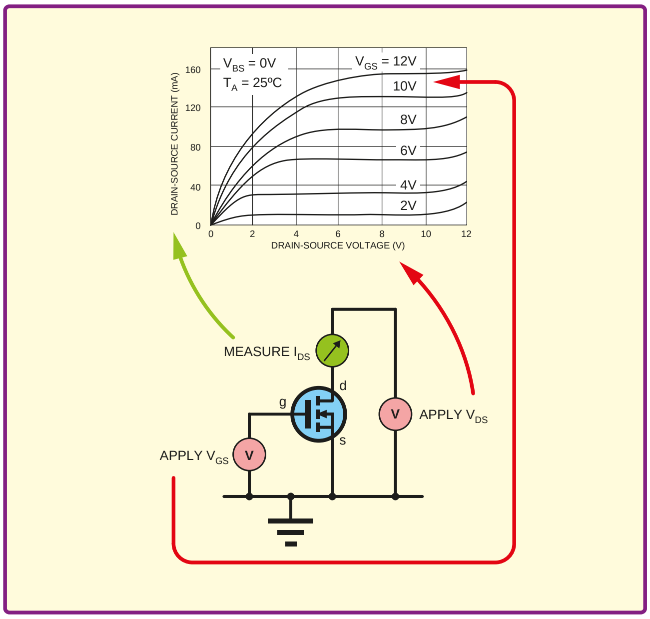


Fig.2. Measuring output characteristics. Note the bulk is connected to the source (V_{BS}=0).

similar range to real devices. This will show us the general shape form of the MOSFET drain current, or output characteristics. For a real device (or via simulation) we can obtain the same curves, as shown in Fig.2. We apply a certain V_{GS} to the MOSFET and then apply a range of V_{DS} values, measuring I_{DS} at each V_{DS} value. From this we plot a curve before moving on to a new V_{GS} value and a new curve. The output characteristics will usually be one of the first graphs presented in a MOSFET datasheet.

MOSFET linear region

Fig.3 shows a plot of the linear region equation for various V_{GS} values. The curves reach a maximum as V_{DS} is increased, with the maximum being higher for larger V_{GS} values. The maximum point is where saturation occurs. The curves for V_{DS} values beyond this point, which is where V_{DS} is greater than (V_{GS}–V_T), do not apply to

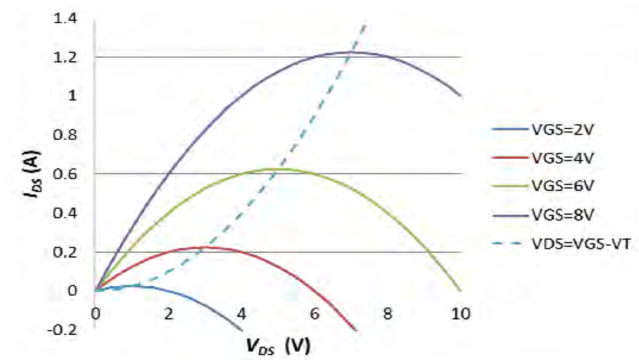


Fig.3. Plot of the MOSFET linear region equation for various gate voltages. The V_{DS}=V_{GS}–V_T curve (dotted) indicates the limit of applicability of this equation. It only applies left of this line

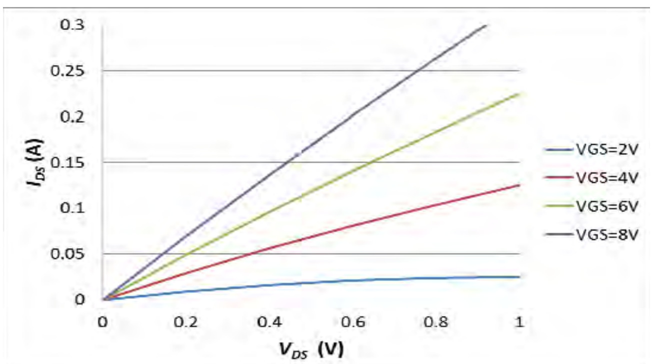


Fig.4. Plot of the MOSFET linear region equation for small values of V_{DS}. Note the (almost) straight-line relationship between voltage and current indicating a resistor-like behaviour

MOSFET characteristics. The saturation point is indicated on Fig.3 by a plot of I_{DS} for all points where V_{DS} = (V_{GS}–V_T), this is the light blue dotted line.

The graph in Fig.4 is based on the same equation and values as Fig.3 – it is just a zoom-in for small values of V_{DS}. Here we see more-or-less straight lines for current variation with applied voltage – just like a resistor. The resistance, indicated by the slope of the line, is controlled by V_{GS}. The steeper the line, the lower the resistance – so increasing V_{GS} results in lower effective resistance between drain and source. When the MOSFET is used as a switch it is this part of characteristic which is of importance. MOSFET datasheets will usually state the on resistance $R_{DS(on)}$ at one or more V_{GS} values.

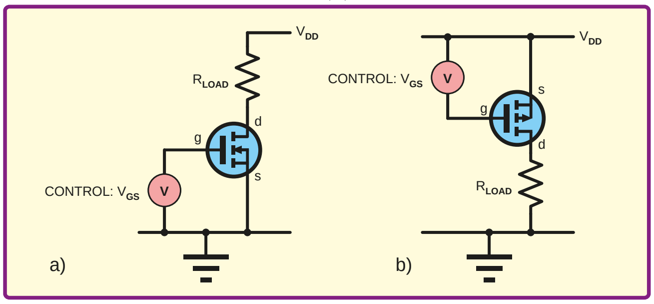


Fig.5. a) N-Channel MOSFET load switching circuit b) P-Channel MOSFET load switching circuit

MOSFET switching

The circuit in Fig.5a is a basic load-switching circuit using an *N*-channel MOSFET. If V_{GS} is well below threshold voltage (typically V_{GS} = 0) the MOSFET will be in the cut-off region and no current will flow in the load. V_{DS} will be close to V_{DD} – so the MOSFET must be rated to withstand this voltage. To switch the load on, a relatively large V_{GS} is applied (well above threshold). The MOSFET will switch on and into the linear region. It will be in the linear region because V_{DS} will be very small, this means that most of V_{DD} will be across the load – which is what we want when it is switched on. The voltage across the MOSFET (V_{DS}) will equal the load current times $R_{DS(on)}$. If V_{DS} is too large with the load on, then the MOSFET may dissipate excessive power, or the load may receive insufficient voltage. In which case a higher V_{GS} (within device limits) may help, or a different MOSFET with a lower $R_{DS(on)}$ is needed. In practice, many other things have to be considered if the load is inductive and/or if very fast switching is required.

The circuit in Fig.5b is a basic *P*-channel MOSFET loading switching circuit. It operates on the same principle as Fig.6, but V_{GS} is a negative voltage with respect to V_{DD}.

Fig.6 shows plots of the saturation region equation for various V_{GS} values. The curves are all simply straight lines parallel with the voltage axis. This is because in saturation I_{DS} does not depend on V_{DS}. There is no V_{DS} term in the saturation equation above. The MOSFET is acting as a constant current source with the current set by V_{GS}. These curves do not apply for V_{DS} less than (V_{GS}–V_T), again this is plotted as the light blue dotted line.

To obtain the full characteristic curve of the MOSFET we can combine the appropriate parts of the curves on Fig.3 and Fig.6. This is shown in Fig.7. For comparison the output characteristics for a couple of real device, the 2N7000G small-signal MOSFET from On Semiconductor is shown in Fig.8 and the ALD1101 matched MOSFET pair is shown in Fig.9. The characteristics are not exactly the same as Fig.7, not each other, but have the same general form. The equations on which Fig.7 is based are only an approximate model of a real device.

Transconductance

Another important MOSFET characteristic curve, which is also usually found on datasheets, is the transfer or transconductance characteristic. This is a plot of I_{DS} against V_{GS} in the saturation region. A plot using the basic equations is shown in Fig.10, this relates directly to Fig.6 in terms of the saturation I_{DS} for a given V_{GS}. Note that the curve starts at I_{DS} = 0 at the point the device just switches on, that is V_{GS} = V_T. For a real device the circuit in Fig.2 can also be used to make the relevant measurements. For comparison, plots from real devices are shown in Fig.11 and Fig.12.

The plot for the ALD1101 is interesting because it show different curves for various bulk source voltages. As discussed last month and mentioned above, the MOSFET is really a four-terminal device, but usually the bulk is connected to the source to give three terminals. The ALD1101 is a dual MOSFET (with matched characteristics) aimed at precision analogue applications, in this it differs from the majority

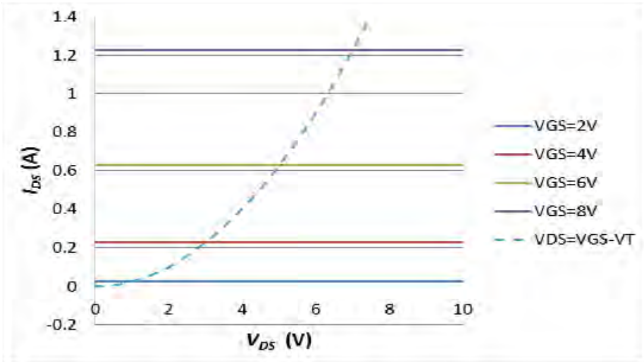


Fig.6. Plot of the MOSFET saturation region equation for various gate voltages. The V_{DS}=V_{GS}–V_T curve (dotted) indicates the limit of applicability of this equation. It only applies right of this line

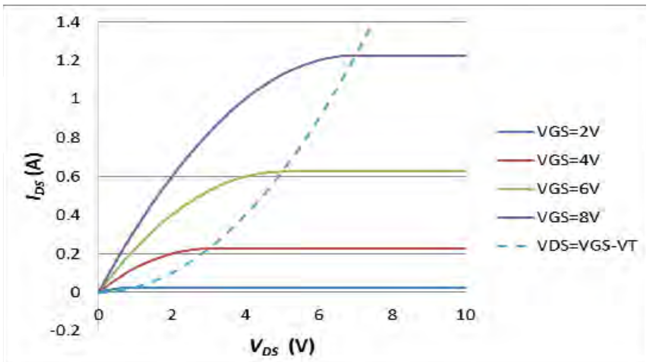


Fig.7. Plot of the MOSFET output characteristics using the appropriate parts of the curves from Fig.3 and Fig.5. The V_{DS}=V_{GS}–V_T curve (dotted) shows the boundary between the linear and saturated regions of operation

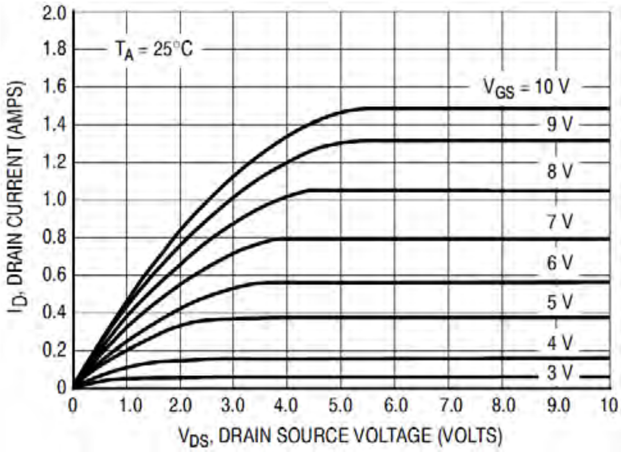


Fig.8. Plot of the MOSFET output characteristics from the 2N7000G datasheet (onsemi.com)

of MOSFETS on the market, which are primarily aimed at switching operation. The bulk, shared by the two devices, is provided as a separate connection (see Fig.13). It is available from Digikey (**digikey.co.uk**).

The transconductance curves relate to the use of the MOSFET as a linear amplifier. The slope of the curve in Fig.10 is the gain of the transistor. A small change in V_{GS} (input) produces a corresponding change in I_{DS} (output). The gain (output/input) is change in I_{DS} divided by change in V_{GS}. The curve is not a straight line, so the gain varies with V_{GS}.

The term transconductance refers to the fact the gain is current divided by a voltage. Recall Ohms law: R = V/I, so I/V = 1/R. The term 1/R is conductance, which had symbol *G* and is measured in siemens (*S*), or sometimes mhos ('ohm' backwards). If we divide a current by a voltage we get a conductance quantity. A gain of the form I/V is a 'transfer conductance' or transconductance, which has symbol g_m ('transfer' because it is from input to output, not just across a two terminal component). The *m* in g_m stands for mutual – the gain is 'mutual' relationship between input and output.

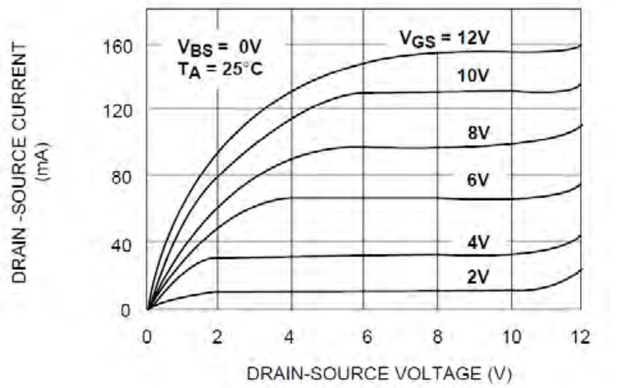


Fig.9. Plot of the MOSFET output characteristics from the ALD1101 datasheet (aldinc.com)

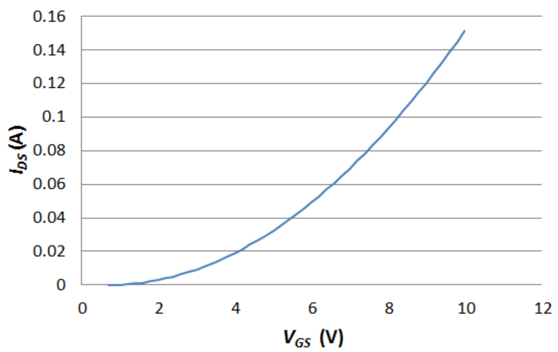


Fig.10. MOSFET transconductance characteristics based on the same equation/data as Fig.5

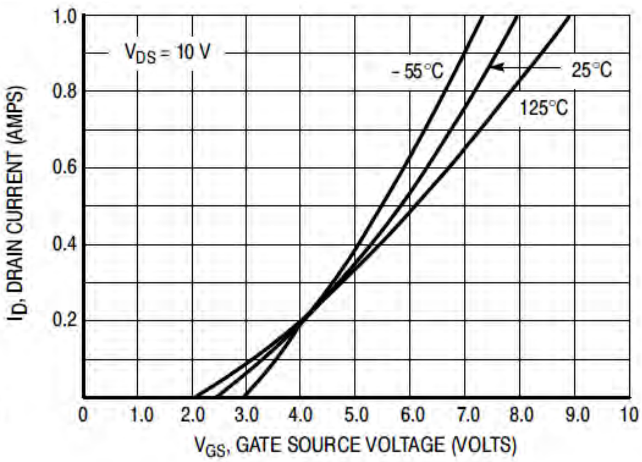


Fig.11. Plot of the MOSFET transconductance characteristics from the 2N7000G datasheet (onsemi.com)

We can obtain an expression for g_m from the saturated I_{DS} equation above using calculus (differentiation). The result is

$$g_m = \sqrt{2\beta|I_{DS}|}$$

In which I_{DS} is the absolute value of I_{DS} to ensure the square root is of a positive value. The equation shows that gain varies with I_{DS}, so the MOSFET is not really linear, however for small changes in V_{GS} at the input (due to the signal being amplified) the value of I_{DS} can be assumed to be constant so the gain is also constant. Larger amplifiers also usually employ feedback which reduces the effects of non-linearity in the transistors.

Basic MOSFET amplifier

Thus, the first step in designing a basic MOSFET amplifier is to decide on a suitable operating point (I_{DS} value and its corresponding V_{GS}) at which to bias the transistor. The amplifier circuit is shown in Fig.14. The MOSFET is a 2N7002, chosen simply because it is available in the LTSpice library (we will simulate the circuit later). This circuit structure may look familiar because the same circuit arrangement can be used with BJTs and JFETs. In this circuit, *R1*

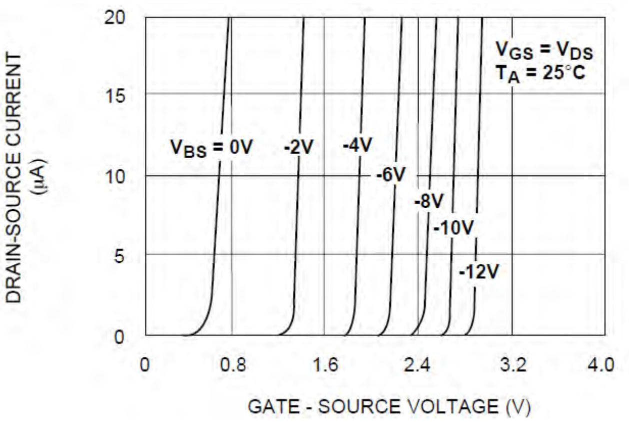


Fig.12. Plot of the MOSFET transconductance characteristics from the ALD1101 datasheet (aldinc.com)

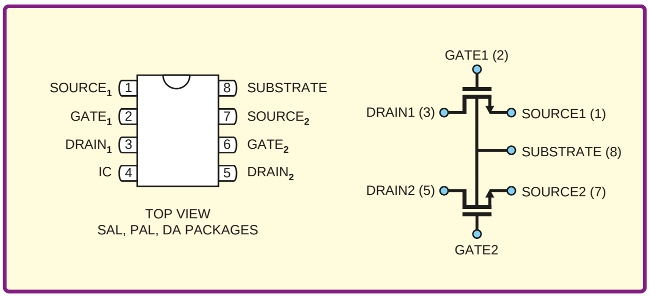


Fig.13. ALD1101 pinout details showing separate bulk connection (aldinc.com)

and *R2* set the bias voltage at the gate, the output voltage is developed across *R3* (due to I_{DS} flowing through it), and *R4* provides negative feedback to stabilise the bias.

The voltage at the gate V_G is given by the standard potential divider equation $V_G = V_{DD}R_2/(R_1+R_2)$ and controls the gate-source voltage along with *R4*. This determines the quiescent (no signal present) drain current, I_{DS}, which we see in the g_m equation. *R1* and *R2* can be large because the MOSFET does not need a continuous current to the gate to operate (in an ideal device the gate in an open circuit). High values of these resistors help make sure the amplifier's input impedance is high. Similar circuits using BJTs cannot usually employ such high values due to the need to supply base current without loading the potential divider.

The quiescent drain-source current is usually chosen by the designer rather than calculated. A value which gives best performance can be selected using the data sheet, but other considerations such as power consumption may come into play. If we know the required value of I_{DS} we can find the corresponding value of V_{GS} using the following equation or using a suitable graph from the data sheet.

$$V_{GS} = V_T + \sqrt{\frac{2I_{DS}}{\beta}}$$

This is just the saturation drain current equation from above, rearranged to give V_{GS}. To use the equation we also need to know the threshold voltage V_T and the transistor gain parameter β.

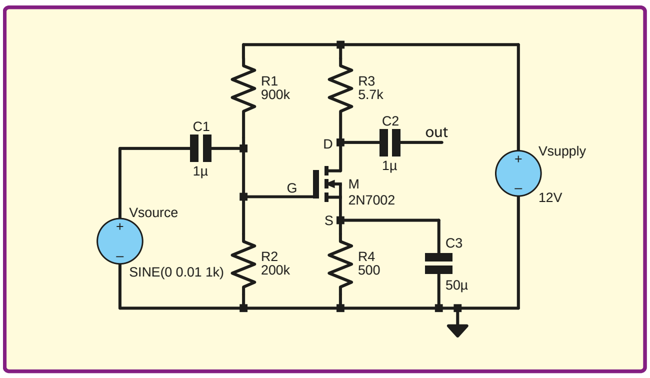


Fig.14. Basic MOSFET amplifier circuit (for LTSpice Simulation)

For example, if we have a MOSFET for which V_T = 1.6V and β = 0.17 A/V² and we choose a quiescent drain-source current of 1mA we require a gate source voltage of 1.6+√(2×0.001/0.17) = 1.71V. Unfortunately, you usually will not find β on the data sheet. However, it is related to the transconductance by the g_m equation above. This equation can be rearranged to:

$$\beta = \frac{g_m^2}{2I_{DS}}$$

Thus, if the datasheet can be used to determine g_m at a suitable I_{DS}, the value of β can be found. The above example has a transconductance of about 18.4mA/V at I_{DS}=1mA.

R4 helps stabilise the bias point against variations between individual devices (in different copies of our circuit) or variations for an individual device with factors such as age and temperature. Each device will have slightly different values of β and V_T and hence different values of I_{DS} will result from a fixed V_{GS}. Similarly I_D will vary with temperature for a fixed V_{GS}. If *R4* was not present these variations would upset the bias conditions. The gain of the circuit is reduced by the negative feedback used to stabilise the bias, but this can be overcome by bypassing *R4* with a large capacitor – effectively a short circuit for AC.

The negative feedback operates as follows. Imagine we have set up the circuit with a particular drain-source current, but due to (say) a change in temperature the drain-source current increases. This results in a larger voltage drop across *R4*, which decreases the gate-source voltage, which in turn reduces the drain-source current, and tends to cancel out the original increase.

If we choose 500Ω for *R4* we get 0.5V at the source from the 1mA bias current. This means, for a V_{GS} of 1.71V, the *R1/R2* potential divider must give 0.5 + 1.71 = 2.21V, so we could use *R1*=900kΩ and *R2*=200kΩ.

The value of *R3* influences the voltage gain and output swing. The drain voltage should not fall below the gate voltage by more than V_T (1.6V in our example), so the minimum drain voltage is 0.61V. The maximum drain voltage is the supply voltage (12V), so we have a maximum swing of 11.4V or ±5.7V centred on 6.31V. For a drain voltage of 6.31V with no signal we need *R3* = (12 – 6.31)/1mA = 5.69kΩ

Results of an LTSpice simulation of the circuit are shown in Fig.15. The voltage and current levels are in line with the calculations above. A 10mV input signal produces an output signal of about 1V, a gain of about 100. The voltage gain can be calculated using g_mR3 = 0.0184×5700=105

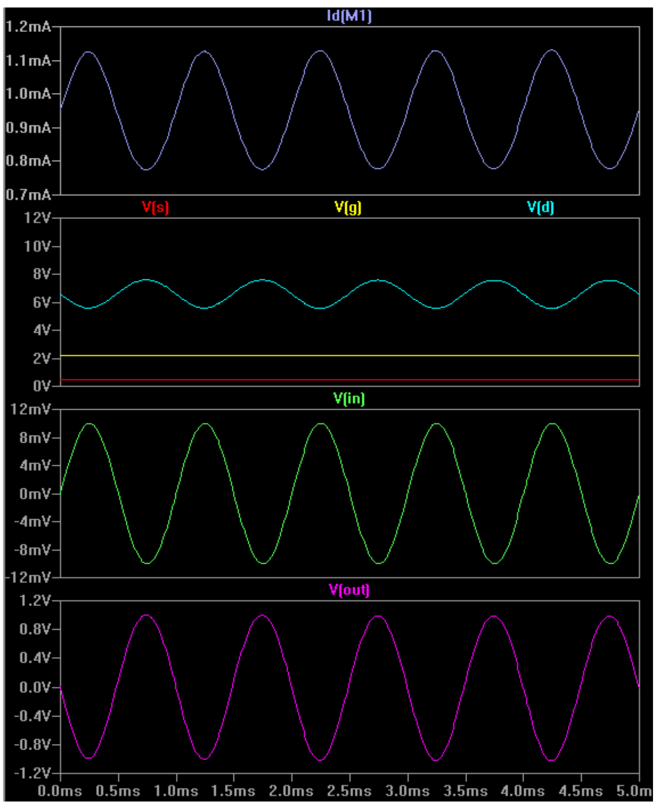


Fig.15. Simulation of the circuit in Fig.14

EPE IS PLEASED TO BE ABLE TO OFFER YOU THESE

ELECTRONICS CD-ROMS

NEW OUT NOW

FLOWCODE 6

Now you can develop complex electronic and electromechanical systems with ease

Flowcode is one of the world's most advanced environments for electronic and electromechanical system development. Engineers use Flowcode to develop systems for control and measurement based on microcontrollers, on rugged industrial interfaces or on Windows compable personal computers

FREE VERSION
The free edition of Flowcode allows users to sample the fully featured product for 30days, after which it will revert to a more limited form and will lose its ability to compile to micro-controllers. It will continue to provide most of its functionality with Matrix hardware devices like the Matrix Forumula Flowcode robot, MIAC and ECIO.

CHIP PACK ONLY - £79.00
Purchasing a Chip Pack only will provide the ability to compile to hardware. Chip Pack versions are suitable for users who only require the basic Flowcode components. The Chip Pack version is equivalent to the Flowcode v5 Student/Home version. Multiple chip packs can be purchased to allow Flowcode to compile to more hardware targets.

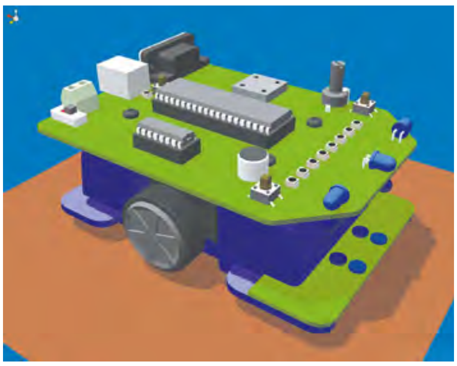
PROFESSIONAL UPGRADE PACK - £220.00
Adding the Professional bolt-on pack to the Flowcode licence will unlock all the functions and components available in Flowcode. It will also be licensed for commercial use.

ACADEMIC VERSIONS
Special arrangements can be made for Academic versions of Flowcode. Academic multi-user versions (10 user, 50 user) are also available. Only educational schools, colleges and universities are able to apply to purchase an Academic version of Flowcode. Please contact us for more information.

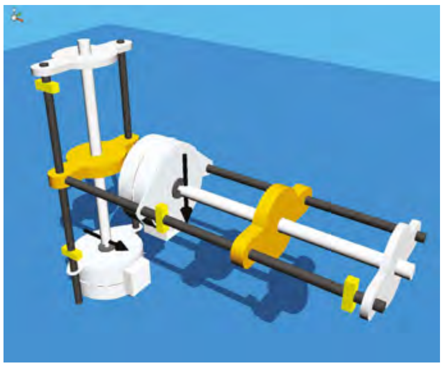
FLOWCODE 5 TO FLOWCODE 6 UPGRADE
Upgrades from Flowcode 5 to Flowcode 6 will be given a discount of 50% to the appropriate licence prices in Flowcode v6.

FLOWCODE 4 TO FLOWCODE 6 UPGRADE
Upgrades from Flowcode 4 to Flowcode 6 will given a discount of 30% to the appropriate licence prices in Flowcode v6 (pay 70% of the price).

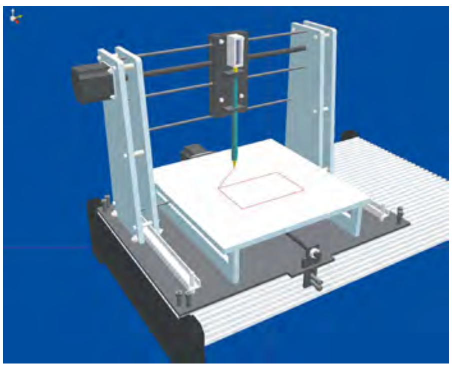
New simulation, 3D and test features allow you to design a greater range of systems including:



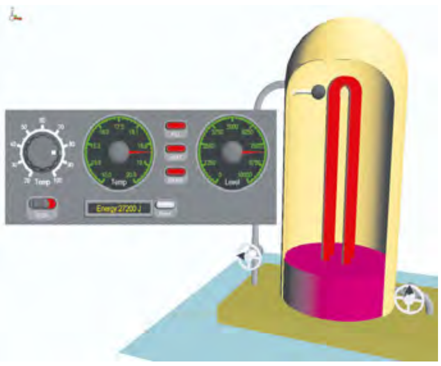
...robotic systems design...



...electromechanical system design...



...machine building...



...pumping and heating systems

PRICES Prices for each of the items above are:

(Order form on third page)

(UK and EU customers add VAT to 'plus VAT' prices)

Flowcode 6

PICMICRO/AVR/DSPIC/PIC24 & ARM Download only

Chip Pack (Hobbyist/Student)	**£94.80 inc. VAT**
Pro-pack (Schools/HE/FE/Industry)	**£299.00 plus VAT**
Flowkit	**£64.90 plus VAT**

Please note: Due to popular demand, Flowcode PICmicro, AVR, DSPIC, PIC24 & ARM V6 are now available as a download. Please include your email address and a username (of your choice) on your order. A unique download code will then be emailed to you. If you require the CDROM as a back-up then please add an extra £14 to the above price.

PICmicro TUTORIALS AND PROGRAMMING

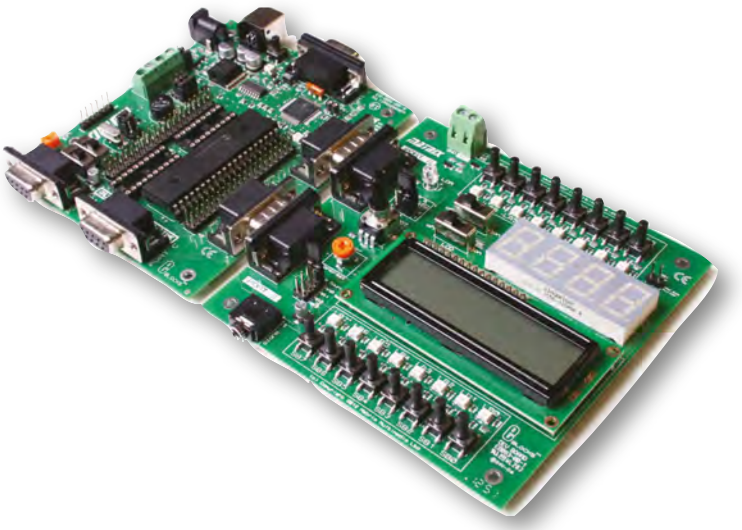
HARDWARE

PICmicro Multiprogrammer Board and Development Board

Suitable for use with the three software packages listed below

This flexible PICmicro microcontroller programmer board and combination board allows students and professional engineers to learn how to program PICmicro microcontrollers as well as program a range of 8, 18, 28 and 40 pin devices from the 12, 16 and 18 series PICmicro ranges. For experienced programmers all programming software is included in the PPP utility that comes with the multiprogrammer board. For those who want to learn, choose one or all of the packages below to use with the hardware.

- Makes it easier to develop PICmicro projects
- Supports low cost Flash-programmable PICmicro devices
- Fully featured integrated displays – 16 individual LEDs, quad 7-segment display and alphanumeric LCD display
- Supports PICmicro microcontrollers with A/D converters
- Fully protected expansion bus for project work
- USB programmable
- Compatible with the E-blocks range of accessories



£167 including VAT and postage, supplied with USB cable and programming software

SOFTWARE

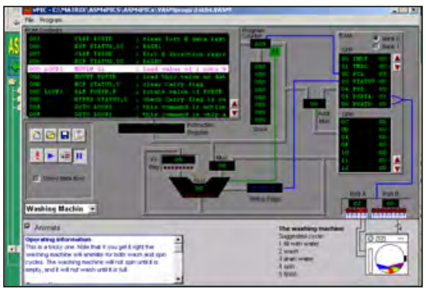
ASSEMBLY FOR PICmicro V4

(Formerly PICtutor)

Assembly for PICmicro microcontrollers V3.0 (previously known as PICtutor) by John Becker contains a complete course in programming the PIC16F84 PICmicro microcontroller from Arizona Microchip. It starts with fundamental concepts and extends up to complex programs including watchdog timers, interrupts and sleep modes.

The CD makes use of the latest simulation techniques which provide a superb tool for learning: the Virtual PICmicro microcontroller, this is a simulation tool that allows users to write and execute MPASM assembler code for the PIC16F84 microcontroller on-screen. Using this you can actually see what happens inside the PICmicro MCU as each instruction is executed, which enhances understanding.

● Comprehensive instruction through 45 tutorial sections ● Includes Vlab, a Virtual PICmicro microcontroller: a fully functioning simulator ● Tests, exercises and projects covering a wide range of PICmicro MCU applications ● Includes MPLAB assembler ● Visual representation of a PICmicro showing architecture and functions ● Expert system for code entry helps first time users ● Shows data flow and fetch execute cycle and has challenges (washing machine, lift, crossroads etc.) ● Imports MPASM files.

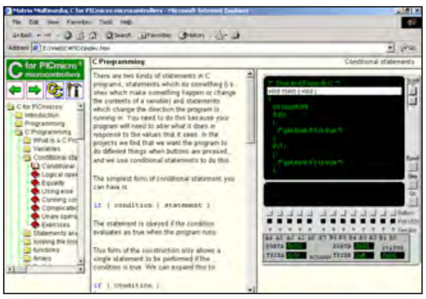


'C' FOR 16 Series PICmicro Version 4

The C for PICmicro microcontrollers CD-ROM is designed for students and professionals who need to learn how to program embedded microcontrollers in C. The CD-ROM contains a course as well as all the software tools needed to create Hex code for a wide range of PICmicro devices – including a full C compiler for a wide range of PICmicro devices.

Although the course focuses on the use of the PICmicro microcontrollers, this CD-ROM will provide a good grounding in C programming for any microcontroller.

● Complete course in C as well as C programming for PICmicro microcontrollers ● Highly interactive course ● Virtual C PICmicro improves understanding ● Includes a C compiler for a wide range of PICmicro devices ● Includes full Integrated Development Environment ● Includes MPLAB software ● Compatible with most PICmicro programmers ● Includes a compiler for all the PICmicro devices.



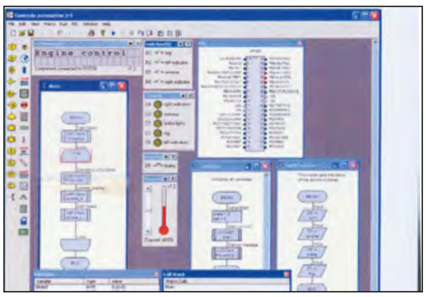
Minimum system requirements for these items: Pentium PC running, 2000, ME, XP; CD-ROM drive; 64MB RAM; 10MB hard disk space.
Flowcode will run on XP or later operating systems

FLOWCODE FOR PICmicro V6 (see opposite page)

Flowcode is a very high level language programming system based on flowcharts. Flowcode allows you to design and simulate complex systems in a matter of minutes. A powerful language that uses macros to facilitate the control of devices like 7-segment displays, motor controllers and LCDs. The use of macros allows you to control these devices without getting bogged down in understanding the programming. When used in conjunction with the Version 3 development board this provides a seamless solution that allows you to program chips in minutes.

- Requires no programming experience
- Allows complex PICmicro applications to be designed quickly
- Uses international standard flow chart symbols
- Full on-screen simulation allows debugging and speeds up the development process.
- Facilitates learning via a full suite of demonstration tutorials
- Produces ASM code for a range of 18, 28 and 40-pin devices
- 16-bit arithmetic strings and string manipulation
- Pulse width modulation
- I2C.

Features include panel creator, in circuit debug, virtual networks, C code customisation, floating point and new components. The Hobbyist/Student version is limited to 4K of code (8K on 18F devices)



PRICES

Prices for each of the CD-ROMs above are:

See previous page for Flowcode Hobbyist/Student prices

(Order form on next page)

(UK and EU customers add VAT to 'plus VAT' prices)

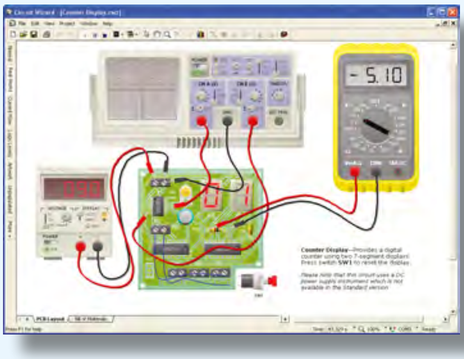
Hobbyist/Student	£58.80	inc VAT
Professional (Schools/HE/FE/Industry)	£150	plus VAT
Professional 10 user (Network Licence)	£499	plus VAT
Site Licence	£999	plus VAT
Flowcode 10 user (Network Licence)	POA	plus VAT
Flowcode Site Licence	POA	plus VAT

CIRCUIT WIZARD

Circuit Wizard is a revolutionary software system that combines circuit design, PCB design, simulation and CAD/CAM manufacture in one complete package.
Two versions are available, Standard or Professional.

By integrating the entire design process, Circuit Wizard provides you with all the tools necessary to produce an electronics project from start to finish – even including on-screen testing of the PCB prior to construction!

✱ Circuit diagram design with component library (500 components Standard, 1500 components Professional) ✱ Virtual instruments (4 Standard, 7 professional) ✱ On-screen animation ✱ Interactive circuit diagram simulation ✱ True analogue/digital simulation ✱ Simulation of component destruction ✱ PCB Layout ✱ Interactive PCB layout simulation ✱ Automatic PCB routing ✱ Gerber export ✱ Multi-level zoom (25% to 1000%) ✱ Multiple undo and redo ✱ Copy and paste to other software ✱ Multiple document support



This software can be used with the *Jump Start* and *Teach-In 2011* series (and the *Teach-In 4* book).

Standard **£61.25** inc. VAT Professional **£91.90** inc. VAT

GCSE ELECTRONICS

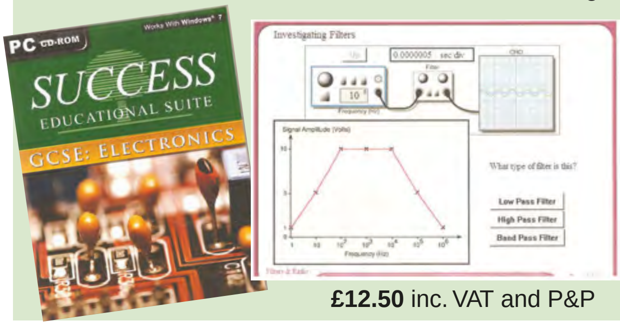
Suitable for any student who is serious about studying and who wants to achieve the best grade possible. Each program's clear, patient and structured delivery will aid understanding of electronics and assist in developing a confident approach to answering GCSE questions. The CD-ROM will be invaluable to anyone studying electronics, not just GCSE students.

✱ Contains comprehensive teaching material to cover the National Curriculum syllabus ✱ Regular exercises reinforce the teaching points ✱ Retains student interest with high quality animation and graphics ✱ Stimulates learning through interactive exercises ✱ Provides sample examination ques-tions with model solutions ✱ Authored by practising teachers ✱ Covers all UK examination board syllabuses ✱ Caters for all levels of ability ✱ Useful for self-tuition and revision

SUBJECTS COVERED

Electric Circuits – Logic Gates – Capacitors & Inductors – Relays – Transistors – Electric Transducers – Operational Amplifiers – Radio Circuits – Test Instruments

Over 100 different sections under the above headings



£12.50 inc. VAT and P&P

Minimum system requirements for these CD-ROMs: Pentium PC, CD-ROM drive, 32MB RAM, 10MB hard disk space. Windows 2000/ME/XP, mouse, sound card, web browser.

Please send me: CD-ROM ORDER FORM

☐ Assembly for PICmicro V4
☐ 'C' for 16 Series PICmicro V4

Version required:
☐ Hobbyist/Student
☐ Professional
☐ Professional 10 user
☐ Site licence

☐ Flowcode for PICmicro V6 (DOWNLOAD ONLY)
☐ Flowcode for AVR V6 (DOWNLOAD ONLY)
☐ Flowcode for ARM V6 (DOWNLOAD ONLY)
☐ Flowcode for dsPIC V6 (DOWNLOAD ONLY)
☐ Flowcode for PIC24 V6 (DOWNLOAD ONLY)
☐ Flowkit

Version required:
☐ Chip pack (Hobbyist/Student)
☐ Pro pack (Schools/HE/FE/Ind.)

Email: . . . .

Username: . . . .

Note: The software on each version is the same, only the licence for use varies.

☐ PICmicro Multiprogrammer Board and Development Board V4 (hardware)

☐ Circuit Wizard – Standard
☐ Circuit Wizard – Professional
☐ GCSE Electronics

Full name: . . . .

Address: . . . .

. . . . Post code: . . . . Tel. No: . . . .

Signature: . . . .

☐ I enclose cheque/PO in £ sterling payable to WIMBORNE PUBLISHING LTD for £ . . . .
☐ Please charge my Visa/Mastercard/Maestro: £ . . . .
Valid From: . . . . Card expiry date: . . . .
Card No: . . . . Maestro Issue No. . . . .
Card Security Code . . . . (The last 3 digits on or just under the signature strip)

ORDERING
ALL PRICES INCLUDE UK POSTAGE

Student/Single User/Standard/Hobbyist Version price includes postage to most countries in the world
EU residents outside the UK add £5 for airmail postage per order

Professional, Multiple User and Site License Versions – overseas readers add £5 to the basic price of each order for airmail postage (**do not add VAT** unless you live in an EU (European Union) country, then add VAT at 20% or provide your official VAT registration number).

Send your order to:
Direct Book Service
Wimborne Publishing Ltd
113 Lynwood Drive, Merley,
Wimborne, Dorset BH21 1UU

To order by phone ring

01202 880299. Fax: 01202 843233

Goods are normally sent within seven days

E-mail: orders@wimborne.co.uk

Online shop:

www.epemag.com

NET WORK

by Alan Winstanley

What's cooking?

WHILE writing this month's copy for *Net Work*, I'm listening to a playlist of MP3s streaming from my NAS. If Wi-Fi is enabled then it's simple enough to listen to a cloud-hosted music collection hosted on, for example, Amazon Cloud instead; if you don't know the artist or album of a particular music track then apps such as Shazam and Soundhound will recognize it for you, and with a mouse-click Amazon will sell it and add it to your MP3 collection effortlessly.

Networks are evolving into something that seamlessly connects appliances, data, media and your bank account together with the minimum of fuss. This sounds fine in theory, but plenty of snags and teething problems exist that cause frustrations for hapless end-users. A few days ago I proudly uploaded some family wedding photos onto my desktop NAS and we gathered around the Smart TV to view them, but the Samsung TV then exclaimed that the files could not be read and the obdurate TV disconnected itself from the NAS and brought the shutters down on the family slideshow. Meantime, my newly-married niece thought nothing of sharing a Dropbox folder where guests could upload their own photos online. This was also a sobering reminder about the myriads of users who have never known life without the web, digital photos, Facebook, MP3s or a mobile phone.

Network-ready

South Korean maker Samsung has teased us with a glimpse of futuristic electrical goods that will be Internet-aware. In the Samsung 'Smart Home' described last month, networked TVs, smartphones and appliances such as deepfreezers will arrive network-ready and can be remote-controlled by apps on a tablet, phone or wristwatch. Samsung's vision of tomorrow alludes to networkable air conditioning, LED colour-changing lighting or even a network-ready vacuum cleaner. Some devices will contain a webcam that allows you to keep an eye on your property, an idea that Samsung has dubbed 'Home View'. Whether at home or abroad, simply open the app and then you can scan around your home, turn up the lighting or view a TV program recorded previously on a home network drive.

Samsung is not alone in creating smart network products and appliances. I thought that April Fool's Day had come early when I saw a press release from Belkin, the maker of domestic networking and computer accessories. Belkin USA is promising a slew of new network-aware products this year, starting with, of all things, a Crock-Pot slow cooker. The US conglomerate Jarden (**www.jarden.com**) will be unfamiliar to UK readers, but it's the name behind many popular consumer brands including Yankee Candle, Sunbeam and the famous Crock-Pot. Their European portfolio includes Bionaire, CampinGaz, Coleman, Breville and others, so with one eye on the future it makes sense for Jarden to team up with a technology player like Belkin.

Hence the Crock-Pot Smart Slow Cooker that can be controlled with an iOS or Android app. As well as the traditional on-off neon of a typical slow cooker, the Smart Slow Cooker sports a Wi-Fi button that hints at the Crock-Pot's network-readiness: as every slow-cooker owner knows, you should never lift the lid to see how the food's doing (tsk!) but now you'll be able to remotely turn the heat up or down, check the timing or receive reminders using your tablet or smartphone. Just the thing for a busy cook. It may sound trivial, but this is absolutely a sign of things to come, and 'connected' coffee machines, domestic heaters, humidifiers and air purifiers are all promised this year.

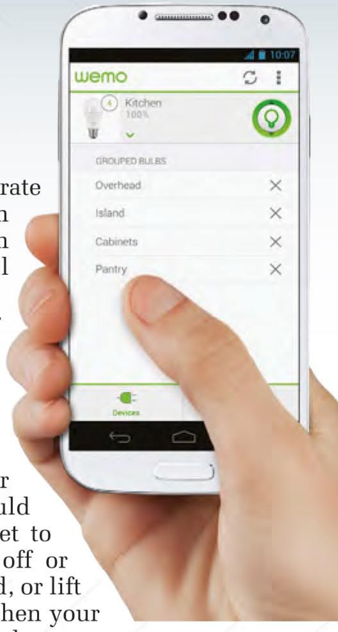
WeMo

Just as Samsung's Smart Home is controlled by a Samsung-powered cloud-based server, Belkin's WeMo home automation products rely on Belkin's cloud for control, but the roll-out in the UK and Europe is lagging behind developments in North America. Perhaps that is no bad thing because at least one security alert claims that WeMo devices could be hacked into remotely, demonstrating a worked example of turning a table lamp on and off very quickly. In theory, a hacker could endeavour to take over somebody's kettle or electric radiator, and the roll-out of Smart Grid meters for gas and electricity monitoring (*Net Work* June 2011) causes similar security concerns. Firmware updates and tiresome patching will be the order of the day, but for most users I expect WeMo will offer adequately-secure and convenient control of lighting or mains outlets.

WeMo LED lights are also produced (not yet available outside the US, though the American light-switch installation video was fascinating stuff for a Brit nonetheless) which will offer simulated occupancy or holiday modes. You can however buy Belkin WeMo remote control mains sockets, a baby monitor and a motion detector kit in the UK (Maplin, Argos, Amazon). Some products are iOS compatible only,

This Smart Slow Cooker from Crock-Pot is at the forefront of the next wave of Internet-aware electrical appliances

A Belkin WeMo mains outlet switch (UK version)



A smartphone app allows control of WeMo devices

so check this carefully if you are an Android or Windows user.

Promised later this year is WeMo Maker, which will interest electronics constructors a lot: it allows you to build your own control systems using a WeMo Maker module to work with 5V sensors or DC switches. You can then manage your own devices from the same WeMo app to control a favourite project, for example. At the time of writing it's not known whether WeMo Maker will make it to Europe.

What IFTTT?

Many years ago I received a phone call from a high-profile exec who travelled extensively. We shared an interest in electronics and technology, and he mentioned a problem that he suffered when travelling around the UK: his house would be cold when he arrived home because the heating had invariably turned off. Was there a way of remotely-controlling it, he asked? At that time, I put him in touch with an *EPE* advertiser who handled X10 remote control systems, which are controlled mainly via ring-mains, Homeplug-style. After installing a simple X10 controller and a slightly unofficial X10 telephone adaptor my friend was delighted that he could now phone home and turn on the heating in advance of his arrival.

X10 is touted as a simple plug-and-play solution to home automation, and unlike ten years ago, iPhone and Android apps are now available to operate them from a tablet or smartphone. Caution is needed when implementing such a system because mains interference from appliances can affect reliability, and the only true way to find out if it works is to suck it and see. One supplier is UK Automation (**www.uk-automation.co.uk**) who retail a range of different remote systems, including X10 and Z-Wave.

Last month, I floated the idea that future networked devices such as 'smart' refrigerators might be controlled automatically by external factors such as the weather forecast. How could an event like a web-based weather alert be used to trigger a particular action like controlling a refrigerator or flashing a red warning light? It's in this area that the idea of **IFTTT** (*If This Then That*) is starting to win friends. IFTTT is a cloud-based macro system that uses web-based events from a select number of 'channels' to trigger a desired 'action'. It already complements many popular web-based services such as Gmail and Twitter, but simple macro routines can be customised to fulfill a more sophisticated routine.

The macros are simple to set up and require little specialist knowledge. IFTTT users can also share their own macros on the IFTTT website at **ifttt.org** and a good place to start is by viewing the expanding list of channels at **https://ifttt.com/channels**.

A trigger might be, for example, the receipt of a Gmail with a file attachment, or an email from a certain address, or a new article appearing on an RSS newsfeed. Then the trigger is connected to an 'Action channel' where the action is configured in detail (such as, the body of an SMS or email to be sent somewhere). IFTTT calls any content (eg, an email, text or SMS) an 'Ingredient' and the complete macro is therefore called a 'Recipe'. By browsing the library of existing Recipes you soon get the idea of how they can automate useful sequences of events.

Practical examples of IFTTT Recipes include an email reminder to pay the rent (Google Calendar), or a weather reminder if it's going to rain the day you start your vacation. Other Android or iOS apps offer more automation still: as many smartphones have GPS, location-sensitive Recipes could be built. Helped by an Internet-based mains controller, your security lights could therefore be switched on as you drive home; send a Tweet automatically to the wife when you're nearly there; post holiday photos automatically to Facebook; receive job alerts as they are published by a company on LinkedIn; get an SMS text reminder before a Google Calendar event starts. There are currently 79 IFTTT channels including email, your Gmail account, Facebook, the weather, Dropbox, Google Calendar, Drive, LinkedIn and YouTube.

More potential

IFTTT has more potential yet, thanks to the channels dedicated to mains hardware controllers like Philips Hue LED lighting and Belkin WeMo. This allows IFTTT users to tie their WeMo or Hue devices to a timed event or trigger them with user's smartphone location, for example. Already, users can turn their LED smart lights to orange automatically (for example) when the Superbowl starts on TV, use iPhone Siri to operate the lights, flash a light a certain colour when mail from an address is received, or turn the lights on automatically when you get home, using a GPS-controlled geo-fence to trigger an SMS. You could even send a Tweet to turn your lights off or send a text instead, or lift the garage door when your phone is a few yards away.

The scope for IFTTT automation is huge and it could well be the missing link that enables non-programmers to devise simple macros dedicated to a particular task. By utilising a choice of IFTTT input and output channels with the latest control hardware, the world of home automation opens up to anyone who can use a phone and plug in an electronic adaptor. It looks promising, and I'm sure my friend would have been thrilled with the idea of turning on the central heating automatically when his car was 75 miles from home.

IFTTT (If This Then That) is a cloud-based service that triggers an Action from a specified Channel. Use it to automate many tasks

That's all for this month's *Net Work*. Next month I'll visit my Synology Network Attached Storage device again and describe a useful upgrade, and I'll bring more news from the network-connected world. You can email the author at **alan@epemag.demon.co.uk**

DIRECT BOOK SERVICE

ELECTRONICS TEACH-IN BUNDLE – SPECIAL BUNDLE PRICE £14 FOR PARTS 1, 2 & 3

ELECTRONICS TEACH-IN 2 CD-ROM
USING PIC MICROCONTROLLERS A PRACTICAL INTRODUCTION

This *Teach-In* series of articles was originally published in *EPE* in 2008 and, following demand from readers, has now been collected together in the *Electronics Teach-In 2* CD-ROM.

The series is aimed at those using PIC microcontrollers for the first time. Each part of the series includes breadboard layouts to aid understanding and a simple programmer project is provided.

Also included are 29 *PIC N' Mix* articles, also republished from *EPE*. These provide a host of practical programming and interfacing information, mainly for those that have already got to grips with using PIC microcontrollers. An extra four part beginners guide to using the C programing language for PIC microcontrollers is also included.

The CD-ROM also contains all of the software for the *Teach-In 2* series and *PIC N' Mix* articles, plus a range of items from Microchip – the manufacturers of the PIC microcontrollers. The material has been compiled by Wimborne Publishing Ltd. with the assistance of Microchip Technology Inc.

CD-ROM **Order code ETI2 CD-ROM** **£9.50**

ELECTRONICS TEACH-IN 3 CD-ROM

The three sections of this CD-ROM cover a very wide range of subjects that will interest everyone involved in electronics, from hobbyists and students to professionals. The first 80-odd pages of Teach-In 3 are dedicated to *Circuit Surgery*, the regular *EPE* clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

The second section – *Practically Speaking* – covers the practical aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and indentifying components, are covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by the readers of *EPE*.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge, TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The Teach-In 1 series covers everything from Electric Current through to Microprocessors and Microcontrollers and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

CD-ROM **Order code ETI3 CD-ROM** **£8.50**

CD-ROMs **Order code ETIBUNDLE** Bundle Price **£14.00**

The books listed have been selected by *Everyday Practical Electronics* editorial staff as being of special interest to everyone involved in electronics and computing. They are supplied by mail order direct to your door. Full ordering details are given on the next page.

FOR A FULL DESCRIPTION OF THESE BOOKS AND CD-ROMS SEE THE SHOP ON OUR WEBSITE www.epemag.com

All prices include UK postage

ELECTRONICS TEACH-IN 5

FREE CD-ROM

ELECTRONICS TEACH-IN 5

Jump Start – 15 design and build circuit projects dedicated to newcomers or those following courses in school and colleges. The projects are: Moisture Detector, Quiz Machine, Battery Voltage Checker, Solar-Powered Charger, Versatile Theft Alarm, Spooky Circuits, Frost Alarm, Mini Christmas Lights, iPod Speaker, Logic Probe, DC Motor Controller, Egg Timer, Signal Injector Probe, Simple Radio Receiver, Temperature Alarm.

PLUS:

PIC' N MIX – starting out with PIC Microcontrollers and PRACTICALLY SPEAKING – the techniques of project construction.

FREE CD-ROM – The free CD-ROM is the complete Teach-In 2 book providing a practical introduction to PIC Microprocessors plus MikroElektronika, Microchip and L-Tek PoScope software.

160 Pages **Order code ETI5** **£8.99**

PROJECT CONSTRUCTION

IC 555 PROJECTS
E. A. Parr
167 pages **Order code BP44** **£5.49**

ELECTRONIC PROJECT BUILDING FOR BEGINNERS R. A. Penfold
135 pages **Order code BP392** **£5.99**

PRACTICAL FIBRE-OPTIC PROJECTS
R. A. Penfold
132 pages **Order code BP374** **£5.45**

ELECTRONICS TEACH-IN 4

ELECTRONICS TEACH-IN 4 CD-ROM

A Broad-Based Introduction to Electronics.

The Teach-In 4 CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the new Diploma in Engineering, Level 2.

The CD-ROM also contains the full Modern Electronics Manual, worth £29.95. The Manual contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to all those interested in learning about electronics or brushing up on their theory, be they hobbyists, students or professionals.

CD-ROM **Order code ETI4 CD-ROM** **£8.99**

RASPBERRY PI

RASPBERRY Pi MANUAL: A practical guide to the revolutionary small computer
176 pages **Order code H001** **£17.99**

RASPBERRY Pi USER-GUIDE
262 pages **Order code JW001** **£12.99**

PROGRAMMING THE RASPBERRY Pi
192 pages **Order code MGH4** **£10.99**

GETTING STARTED WITH RASPBERRY Pi
164 pages **Order code OR01** **£11.50**

MICROPROCESSORS

INTERFACING PIC MICROCONTROLLERS
Martin Bates
298 pages **Order code NE48** **£30.99**

PROGRAMMING 16-BIT PIC MICROCONTROLLERS IN C – LEARNING TO FLY THE PIC24 Lucio Di Jasio (Application Segments Manager, Microchip, USA)
496 pages +CD-ROM **Order code NE45** **£38.00**

INTRODUCTION TO MICROPROCESSORS AND MICROCONTROLLERS – SECOND EDITION
John Crisp
222 pages **Order code NE31** **£29.99**

THE PIC MICROCONTROLLER YOUR PERSONAL INTRODUCTORY COURSE – THIRD EDITION. John Morton
270 pages **Order code NE36** **£25.00**

PIC IN PRACTICE (2nd Edition)
David W. Smith
308 pages **Order code NE39** **£24.99**

MICROCONTROLLER COOKBOOK
Mike James
240 pages **Order code NE26** **£36.99**

THEORY AND REFERENCE

PRACTICAL ELECTRONICS HANDBOOK – Fifth Edition. Ian Sinclair
440 pages **Order code NE21** **£32.50**

STARTING ELECTRONICS – 4th Edition
Keith Brindley
296 pages **Order code NE100** **£18.99**

ELECTRONIC CIRCUITS – FUNDAMENTALS & APPLICATIONS – Third Edition Mike Tooley
400 pages **Order code TF43** **£25.99**

FUNDAMENTAL ELECTRICAL AND ELECTRONIC PRINCIPLES – Third Edition C.R. Robertson
368 pages **Order code TF47** **£21.99**

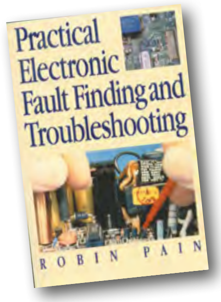
A BEGINNER'S GUIDE TO TTL DIGITAL ICs
R.A. Penfold
142 pages **OUT OF PRINT BP332** **£5.45**

UNDERSTANDING ELECTRONIC CONTROL SYSTEMS
Owen Bishop
228 pages **Order code NE35** **£36.99**

FAULT FINDING AND TEST EQUIPMENT

GETTING THE MOST FROM YOUR MULTIMETER
R. A. Penfold
96 pages Order code BP239 £5.49

PRACTICAL ELECTRONIC FAULT FINDING AND TROUBLESHOOTING
Robin Pain
274 pages Order code NE22 £41.99



HOW ELECTRONIC THINGS WORK – AND WHAT TO DO WHEN THEY DON'T
Robert Goodman
394 pages Order code MGH3 £21.99

OSCILLOSCOPES – FIFTH EDITION
Ian Hickman
288 pages Order code NE37 £36.99

HOW TO USE OSCILLOSCOPES AND OTHER TEST EQUIPMENT
R.A. Penfold
104 pages Order code BP267 £5.49

AUDIO & VIDEO

VALVE AMPLIFIERS – Second Edition
Morgan Jones
288 pages Order code NE33 £40.99

BUILDING VALVE AMPLIFIERS
Morgan Jones
368 pages Order code NE40 £29.00

DIGITAL AUDIO RECORDING
Ian Waugh
60 pages Order code PC121 £7.95

QUICK GUIDE TO MP3 AND DIGITAL MUSIC
Ian Waugh
60 pages Order code PC119 £7.45

MAKING MUSIC WITH YOUR COMPUTER
Stephen Bennett
92 pages Order code PC120 £10.95

QUICK GUIDE TO DIGITAL AUDIO RECORDING
Ian Waugh
208 pages Order code PC121 £7.95

ELECTRONIC PROJECTS FOR VIDEO ENTHUSIASTS
R.A. Penfold
109 pages Order code BP356 £5.45

COMPUTING AND ROBOTICS

NEWNES INTERFACING COMPANION
Tony Fischer-Cripps
295 pages Order code NE38 £41.00

HOW TO BUILD A COMPUTER MADE EASY
R.A. Penfold
120 pages Order code BP707 £8.49

EASY PC CASE MODDING
R.A. Penfold
192 pages + CDROM Order code BP542 £8.99

FREE DOWNLOADS TO PEP-UP AND PROTECT YOUR PC
R.A. Penfold
128 pages Order code BP722 £7.99

WINDOWS XP EXPLAINED
N. Kantaris and P.R.M. Oliver
264 pages Order code BP514 £7.99

THE INTERNET – TWEAKS, TIPS AND TRICKS
R. A. Penfold
128 pages Order code BP721 £7.99

eBAY – TWEAKS, TIPS AND TRICKS
R. A. Penfold
128 pages Order code BP716 £7.50

AN INTRODUCTION TO eBAY FOR THE OLDER GENERATION
Cherry Nixon
120 pages Order code BP709 £8.49

COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
308 pages Order code BP601 £8.99

ANDROIDS, ROBOTS AND ANIMATRONS
Second Edition – John Iovine
224 pages Order code MGH1 £16.99

ROBOT BUILDERS COOKBOOK
Owen Bishop
366 pages Order code NE46 £26.00

INTRODUCING ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
288 pages + Order code BP901 £14.99

MORE ADVANCED ROBOTICS WITH LEGO MINDSTORMS – Robert Penfold
298 pages Order code BP902 £14.99

HOW TO FIX YOUR PC PROBLEMS
R. A. Penfold
128 pages Order code BP705 £8.49

WINDOWS 7 – TWEAKS, TIPS AND TRICKS
Andrew Edney
120 pages Order code BP708 £8.49

GETTING STARTED IN COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
120 pages Order code BP704 £8.49

BOOK ORDERING DETAILS

All prices include UK postage. For postage to Europe (air) and the rest of the world (surface) please add £3 per book. Surface mail can take up to 10 weeks to some countries. For the rest of the world airmail add £4 per book. CD-ROM prices include VAT and/or postage to anywhere in the world. Send a PO, cheque, international money order (£ sterling only) made payable to **Direct Book Service** or card details, Visa, Mastercard or Maestro to: **DIRECT BOOK SERVICE, WIMBORNE PUBLISHING LIMITED, 113 LYNWOOD DRIVE, MERLEY, WIMBORNE, DORSET BH21 1UU**.

Books are normally sent within seven days of receipt of order, but please allow 28 days for delivery – more for overseas orders. *Please check price and availability (see latest issue of* Everyday Practical Electronics*) before ordering from old lists.*

For a full description of these books please see the shop on our website.
Tel 01202 880299 Fax 01202 843233. E-mail: dbs@wimborne.co.uk

Order from our online shop at: www.epemag.com

BOOK ORDER FORM

Full name: ....

Address: ....

....

....

.... Post code: .... Telephone No: ....

Signature: ....

☐ I enclose cheque/PO payable to DIRECT BOOK SERVICE for £ ....

☐ Please charge my card £ .... Card expiry date..............................

Card Number .... Maestro Issue No............................

Valid From Date .... Card Security Code .... (The last three digits on or just below the signature strip)

Please send book order codes: ....

....

Please continue on separate sheet of paper if necessary

PCB SERVICE

CHECK US OUT ON THE WEB

Basic printed circuit boards for most recent *EPE* constructional projects are available from the *PCB Service*, see list. These are fabricated in glass fibre, and are drilled and roller tinned, but all holes are a standard size. They are not silk-screened, nor do they have solder resist. Double-sided boards are **NOT plated through hole** and will require 'vias' and some components soldering to both sides. * **NOTE: PCBs from the July 2013 issue with eight digit codes** have silk screen overlays and, where applicable, are double-sided, plated through-hole, with solder masks, they are similar to the photos in the relevent project articles.

All prices include VAT and postage and packing. Add £2 per board for airmail outside of Europe. Remittances should be sent to **The PCB Service, Everyday Practical Electronics, Wimborne Publishing Ltd., 113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU. Tel: 01202 880299; Fax 01202 843233; Email: orders@epemag.wimborne.co.uk. On-line Shop: www.epemag.com**. Cheques should be crossed and made payable to *Everyday Practical Electronics* **(Payment in £ sterling only)**.

NOTE: While 95% of our boards are held in stock and are dispatched within seven days of receipt of order, please allow a maximum of 28 days for delivery – overseas readers allow extra if ordered by surface mail.

Back numbers or photocopies of articles are available if required – see the Back Issues page for details. WE DO NOT SUPPLY KITS OR COMPONENTS FOR OUR PROJECTS.

PROJECT TITLE	ORDER CODE	COST
MARCH '13		
Lightning Detector	888	£8.75
★ Digital Spirit Level	889	£8.75
★ SemTest – Part 2		
– Main/Lower Board	890	£16.52
– Display/Upper Board	891	£15.55
Interplanetary Voice	892	£8.75
Jump Start – DC Motor Controller	893	£8.55
APRIL '13		
Six-Decade Resistance Substitution Box	894	£10.10
SoftStarter (single-sided)	895	£8.36
Jump Start – Egg Timer	896	£8.36
SemTest HV DC Crowbar	897	£13.61
MAY '13		
Electronic Stethoscope	898	£9.72
PIC/AVR Programming Adaptor Board (d'ble-sided)	899	£23.33
Jump Start – Signal Injector Probe	900	£8.16
JUNE '13		
USB Breakout Box	901	£7.97
Jump Start – Simple Radio Receiver	902	£8.94
Mix-It	903	£11.28
JULY '13		
6-Decade Capacitance Substitution Box	04106121	£18.00
6-Decade Capacitance Substitution Box Panel/Lid	04106122	£18.00
SoftStarter For Power Tools	10107121	£9.00
High-Current Adaptor For Scopes And DMMs	04108121	£18.00
Jump Start – Temperature Alarm	904	£7.97
AUGUST '13		
Driveway Sentry	03107121	£18.00
Build A Vox	01207111	£22.00
Milliohm Meter Adaptor For DMMs	04102101	£18.00
SEPTEMBER '13		
MiniReg Adjustable Regulator	18112111	£6.00
USB Stereo Recording Interface	07106111	£18.00
Digital Sound Effects Module	09109121	£9.00
OCTOBER '13		
High-Temperature Thermometer/Thermostat	21105121	£22.00
LED Musicolour	16110121	£22.00
NOVEMBER '13		
Mains Timer For Fans Or Lights	10108121	£9.10
CLASSiC-D Amplifier (inc. 3 Rockby Capacitors)	01108121	£26.50
– Speaker Protector	01108122	£9.10
DECEMBER '13		
CLASSiC-D Amplifier – Power Supply	01109111	£16.66
USB Instrument Interface	24109121	£26.38
– Front Panel	24109122	£28.54

PROJECT TITLE	ORDER CODE	COST
JANUARY '14		
Champion	1109121/2	£8.88
Simple 1.5A Switching Regulator	18102121	£5.53
FEBRUARY '14		
2.5GHz 12-Digit Frequency Counter – Display PCB	04111122	£12.88
– Main PCB	04111121	£27.13
– Front Panel	04111123	£22.38
High-Energy Electronic Ignition System	05110121	£9.10
Mobile Phone LOUD Ringer!	12110121	£9.10
MARCH '14		
Extremely accurate GPS 1pps Timebase For A Frequency Counter	04103131	£8.88
Infrasound Detector – Pre CHAMP PCB	01107941	£5.54
– CHAMP PCB	01102941	£5.54
Automatic Points Controller (inc. 2 sensor PCBs)	09103132	£13.42
Automatic Points Controller – Frog Relay	09103133	£5.54
APRIL '14		
A Capacitor Discharge Unit For Twin-Coil Points Motors	09203131	£9.10
Deluxe GPS 1pp Timebase For Frequency Counters	04104131	£16.66
Jacob's Ladder	05110121	£9.10

*** See NOTE above left regarding PCBs with eight digit codes ***

Please check price and availability in the latest issue.
A large number of older boards are listed on, and can be ordered from, our website.
Boards can only be supplied on a payment with order basis.

EPE SOFTWARE

★ All software programs for *EPE* Projects marked with a star, and others previously published can be downloaded free from the Library on our website, accessible via our home page at: **www.epemag.com**

PCB MASTERS

PCB masters for boards published from the March '06 issue onwards can also be downloaded from our website (**www.epemag.com**); **go to the 'Library' section.**

EPE PRINTED CIRCUIT BOARD SERVICE

Order Code **Project** **Quantity** **Price**

. . . .

Name . . . .

Address . . . .

. . . .

Tel. No. . . . .

I enclose payment of £ . . . . (cheque/PO in **£ sterling only**) to:

Everyday Practical Electronics

Card No. . . . .

Valid From . . . . Expiry Date . . . .

Card Security No. . . . . Maestro Issue No. . . . . . .

Signature . . . .

Note: You can also order PCBs by phone, Fax or Email or via the Shop on our website on a secure server:

http://www.epemag.com

CLASSIFIED

ADVERTISEMENTS

ADVERTISERS INDEX

BETA LAYOUT . . . . 32 & 72
BRUNNING. 47
CRICKLEWOOD ELECTRONICS 65
ESR ELECTRONIC COMPONENTS 6
iCSAT 65
JPG ELECTRONICS 72
L-TEK POSCOPE 55
MICROCHIP . . . . Cover (iii)
LABCENTER . . . . Cover (iv)
LASER BUSINESS SYSTEMS 49
MIKROELEKTRONIKA . . . . Cover (ii)
PEAK ELECTRONIC DESIGN 4
PICO TECHNOLOGY 65
QUASAR ELECTRONICS 2/3
STEWART OF READING 4
TECHNOBOTS 57
TRINITY. 51

ADVERTISEMENT OFFICES:
113 LYNWOOD DRIVE, MERLEY, WIMBORNE,
DORSET BH21 1UU
PHONE: 01202 880299
FAX: 01202 843233
EMAIL: stewart.kearn@wimborne.co.uk
WEB: www.epemag.com

For editorial address and phone numbers see page 7

Next Month

The CLASSiC-D ±35V DC-DC Converter

This compact DC-DC converter has been designed to mate with our *CLASSiC-D Amplifier* (November and December 2013). It presents an efficient way to run the CLASSiC-D amplifier module from a battery to make it a compact powerhouse.

DMM auto power-down

Does your digital multimeter lack an auto power-down facility? If you forget to turn it off, next time you go to use it you might be tempted to say naughty words. This little circuit, which will cost just a couple of pounds, will stop a DMM chewing through batteries when you forget to turn it off.

Rugged Battery Charger from Bits'n'Pieces

Need a big, powerful, simple and cheap battery charger? Then look no further, we have just the design you've been looking for! (*delayed from the April 2014 issue!*)

Teach-In 2014: Raspberry Pi – Part 8

In next month's *Teach-In 2014*, our *Pi Class* introduces stepper motors and our *Pi Project* features the construction of a complete stepper motor controller. Our *Home baking* feature shows you how to connect a camera to the Raspberry Pi and how you can use this as the basis of a complete Internet-connected webcam application.

MAY '14 ISSUE ON SALE 3 APRIL 2014

Content may be subject to change

Published on approximately the first Thursday of each month by Wimborne Publishing Ltd., 113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU. Printed in England by Acorn Web Offset Ltd., Normanton, WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions INLAND: £23.50 (6 months); £43.00 (12 months); £79.50 (2 years). EUROPE: airmail service, £27.00 (6 months); £50.00 (12 months); £95.00 (2 years). REST OF THE WORLD: airmail service, £37.00 (6 months); £70.00 (12 months); £135.00 (2 years). Payments payable to "Everyday Practical Electronics", Subs Dept, Wimborne Publishing Ltd. Email: subs@epemag.wimborne.co.uk. EVERYDAY PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.

Looking to speed your analog development time?

PIC® MCUs with Intelligent Analog make designs easier

With a powerful combination of rich analog integration and low power consumption, the PIC24FJ128GC010 family enables a significant cost reduction over a multi-chip design as well as enabling lower noise, faster throughput, smaller PCB size and a faster time to market.

In addition to Microchip's first 16-bit ADC and a 10 Msps 12-bit ADC, the PIC24FJ128GC010 integrates a DAC and dual op amps to simplify precision analog design. The on-chip LCD driver provides the ability to drive displays with up to 472 segments for information-rich user displays; whilst mTouch™ capacitive touch sensing adds advanced touch capabilities.

The PIC24FJ128GC010 family helps to reduce noise to deliver more consistent analog performance in a very small form factor. Simply add sensors to the low-cost starter kit for easy prototyping.

GET STARTED IN 3 EASY STEPS:

1. Begin with the low-cost PIC24F Starter Kit for Intelligent Analog (DM240015)
2. Add custom sensors to the clean analog header to create a prototype
3. Re-use and modify the demo code to speed development

PIC24F Starter Kit for Intelligent Analog (DM240015)

For more information, go to: www.microchip.com/get/euGC010

MICROCHIP

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless