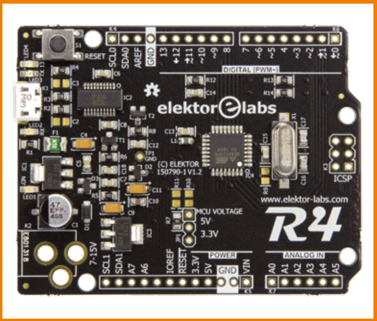
Edition 4/2016 | July & August 2016
www.elektormagazine.com

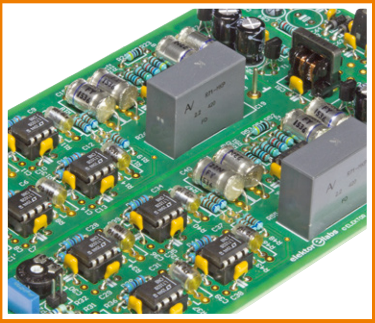
elektor

LEARN DESIGN SHARE



Elektor Uno R4
Four pins make all the difference

Bat Detector Plus
with amplitude recovery



SUPRA 2.0
super low-noise MM/MD phono preamp

Elektor SDR Reloaded

Software Defined Radio Shield for Arduino

PicoScope® 3000 Series

FLEXIBLE RESOLUTION OSCILLOSCOPE

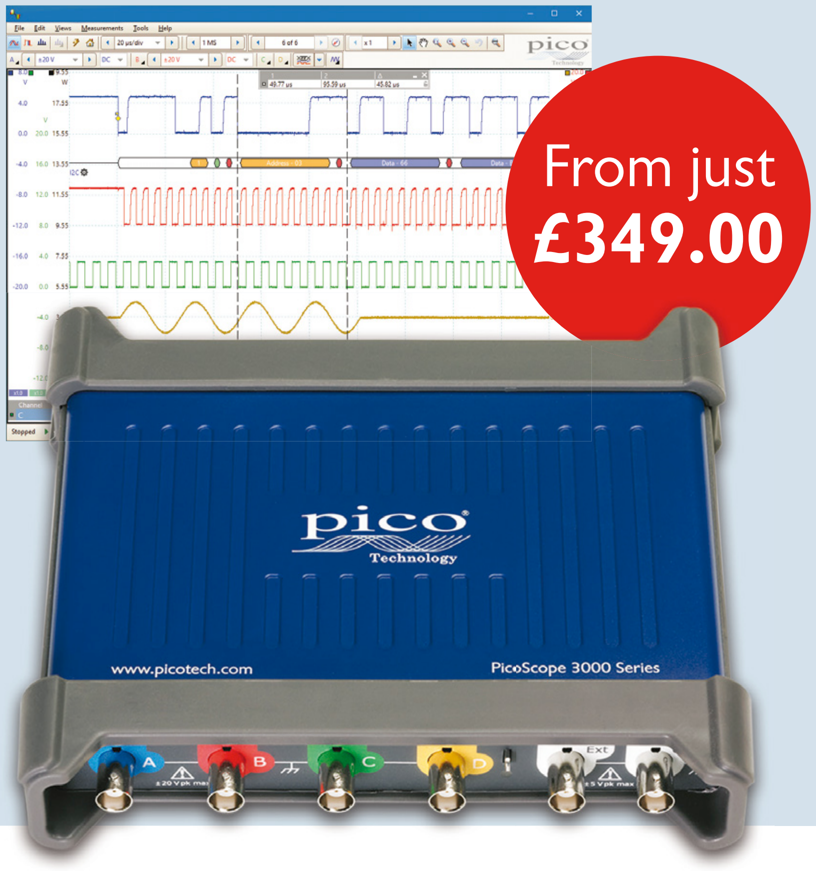
PC Oscilloscopes & Mixed Signal Oscilloscopes

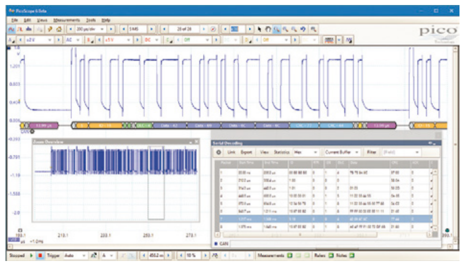
PicoScope 3000 Series USB-powered PC oscilloscopes are small, light, and portable and can easily slip into a laptop bag while offering a range of high-performance specifications.

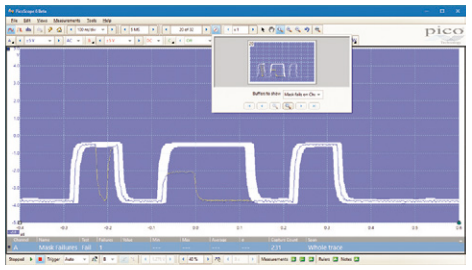
These oscilloscopes offer 2 or 4 analog channels and a built-in function/arbitrary waveform generator. MSO models add 16 digital channels.

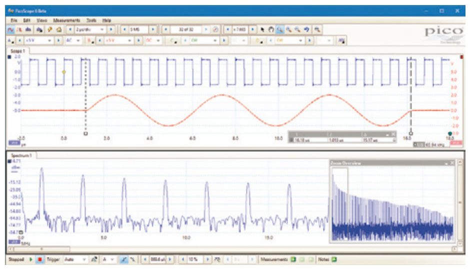
Key performance specifications:

- 200 MHz analog bandwidth
- 1 GS/s real-time sampling
- 512 MS buffer memory
- 100,000 waveforms per second
- 16 channel logic analyzer (MSO models)
- Arbitrary waveform generator
- USB 3.0 connected and powered









All models include full software and 5 year warranty. Software includes measurements, spectrum analyzer, advanced triggers, color persistence, serial decoding (16 protocols including 1-Wire, CAN, Ethernet, I²C, I²S, LIN, RS-232, SENT, SPI, USB 1.1), masks, math channels, with FREE updates. FREE Software Development Kit also available.

www.picotech.com/PS469

Edition 4/2016
Volume 42, No. 475 & 476
July & August 2016

ISSN 1757-0875 (UK / ROW distribution)

www.elektor.com
www.elektormagazine.com
www.elektor-labs.com

Elektor Magazine, English edition
is published 6 times a year by

Elektor International Media
78 York Street
London
W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344

Head Office:
Elektor International Media b.v.
PO Box 11
NL-6114-ZG Susteren
The Netherlands
Phone: (+31) 46 4389444
Fax: (+31) 46 4370161

Memberships:
Please use London address
E-mail: service@elektor.com
www.elektor.com/memberships

Advertising & Sponsoring:
Johan Dijk
Phone: +31 6 15894245
E-mail: johan.dijk@eimworld.com

www.elektor.com/advertising
Advertising rates and terms available on request.

Tuning for …

Within the Elektor editorial team I am no exception when it comes to “getting all the details in” so you, our readers, have the best chance of being able to replicate a project successfully, or start feeling confident with a new technology (like **LoRa**). With no counterforces like The Publishers, The Printers and the Chief Financial Officer to hold me back, almost any one article from this edition can be expanded, deepened, widened, technified, Elektorized, or embellished to fill double the amount of pages allocated to it in the current layout.
Or half the amount — and therein lies the problem. While some projects fare well with summary descriptions, a rough schematic, and the heading READER’s PROJECT, others, like **SUPRA 2.0** and **Bat Detector** in this very edition are different. They obviously require wide space not just to help you build them at home with good confidence of success, but also to be able to get the technical content across. Content that reflects the joint effort of the author, the Elektor Labs team, and the editors.
If the feedback the editorial team gets on certain articles “being too long” *versus* “lacking in detail” were the currents feeding a center-zero moving coil meter, the needle would be comfortably at 0 most of the time, with just a little movement to either side depending on the edition. Nothing to warrant retuning though, our in-built AFC will prevail. And we have not cheated with the series resistances in the bridge. Similar center-zero dials exist at Elektor for analog *versus* microcontroller, SMD *versus* through-hole, and new technologies *versus* boatachors.
I am a long-term F license radio amateur and from this here the transmitting side I’d be curious to know your receive report for this magazine’s center-tuning and signal strength. I hope the signal is strong enough with good readability, with no splatter or blocking the band to others. Over to you, I am at editor@elektor.com.

Happy Reading, 73,

Jan Buiting, Editor-in-Chief, PE1CSI

THIS EDITION

Volume 42 – Edition 4/2016

No. 475 & 476 July & August 2016

Elektor Uno R4

Four pins make all the difference

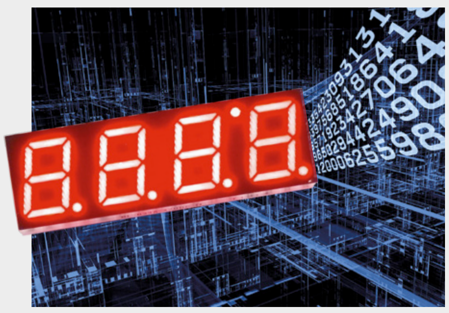
98

When a manufacturer releases a B version of an existing product the differences are marginal in most cases. Not so with the new ATmega328P, the processor at the heart of the Arduino Uno R3. The B upgrade of this MCU features new peripherals that justify a new revision of the Arduino Uno R3. Here is the R4.

24

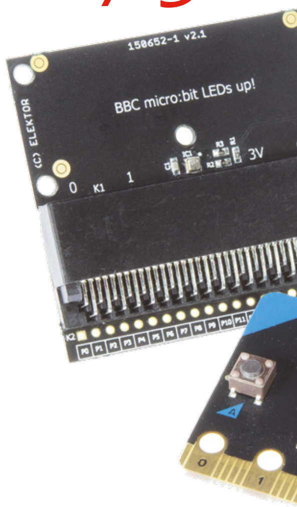
PIC ASSEMBLER CRASH COURSE 2.1

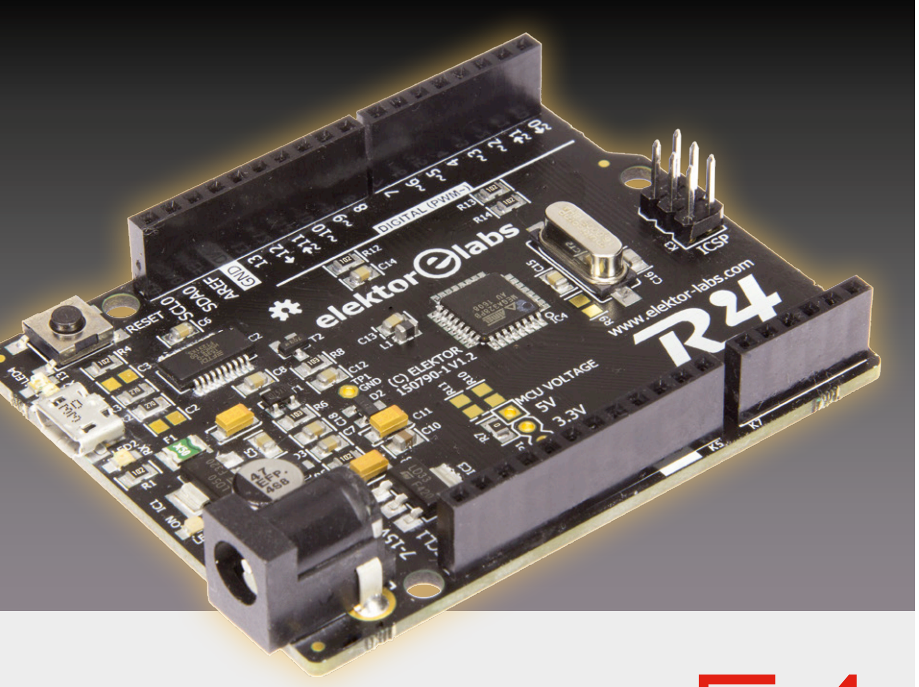
DRIVING LED DISPLAYS USING INTERRUPTS



Last year we published a three-part crash course in assembler programming for PIC microcontrollers, illustrating the theory and practice of programming these devices close to the metal. In this second series we delve deeper, beginning in this installment with a look at interrupts.

79

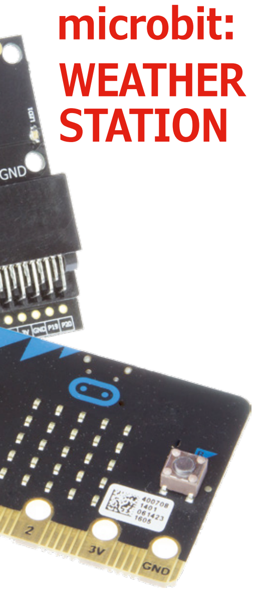




ELEKTOR SDR RELOADED 54

SDR SHIELD FOR THE ARDUINO

A Software Defined Radio is a universal tool in RF technology circles, one that can also be put to use for making measurements. The characteristics of the receiver are defined in software, which now gives us the opportunity to use an Arduino Shield as a front-end.



BBC microbit: WEATHER STATION

NEXT EDITION

Timeshift Radio

This project saves audio data received into a circular buffer (ring memory), enabling you to replay it on demand. A Silabs receiver, an audio codec and a small ARM processor board are the ingredients you'll need.

LEDitron

Conventional low-energy lamps are a dying breed as we have entered the LED era. LEDitron is a seven-segment display module made from LED filaments, similar to a vintage Numitron tube but with better energy efficiency.

A Simple Current Sensor Probe

If the multimeter is the most important test and measurement device for electronicists, there can be no dispute that the oscilloscope comes next. Its application is not only universal but also highly flexible, its optical display revealing exactly what's going on in dynamic signals or voltage curves. But herein lies the problem: unlike a multimeter, an oscilloscope just not innately equipped for measuring current flows. With this probe you can change that!

Elektor Magazine edition 5 / 2016 covering September & October is published on August 10, 2016.
Delivery of printed copies to Elektor Gold Members is subject to transport.
Contents and article titles subject to change.

The Elektor Community

LEARN | DESIGN | SHARE

Elektor breaks the constraints of a magazine. It's a community of active e-engineers — from novices to professionals — eager to learn, make, design, and share surprising electronics.

Elektor Web Store: 24/7 candy store for every electronics engineer! Permanent 10% discount for GREEN and GOLD Members.
www.elektor.com

Elektor Magazine: Six times per year a thick publication packed with electronics projects, news, reviews, tips and tricks.
www.elektormagazine.com

Elektor PCB Service: Order your own PCBs, both one-offs and larger runs.
www.elektorpcbservice.com

Elektor Weekly & Paperless: Your digital weekly news update. Free.
www.elektor.com/newsletter

Elektor Academy: Webinars, Seminars, Presentations, Workshops and DVDs ... Practice-oriented learning.
www.elektor-academy.com

Elektor Books: Arduino, Raspberry Pi, microcontrollers, Linux and more. Available in our online store with a 10% Member discount!
www.elektor.com/books

Elektor TV: Reviews, timelapse, unboxing and personal journals. Watching is learning.
www.youtube.com/user/ElektorIM

Elektor Labs: Showcasing your own projects and learning from others. We develop and test your ideas!
www.elektormagazine.com/labs

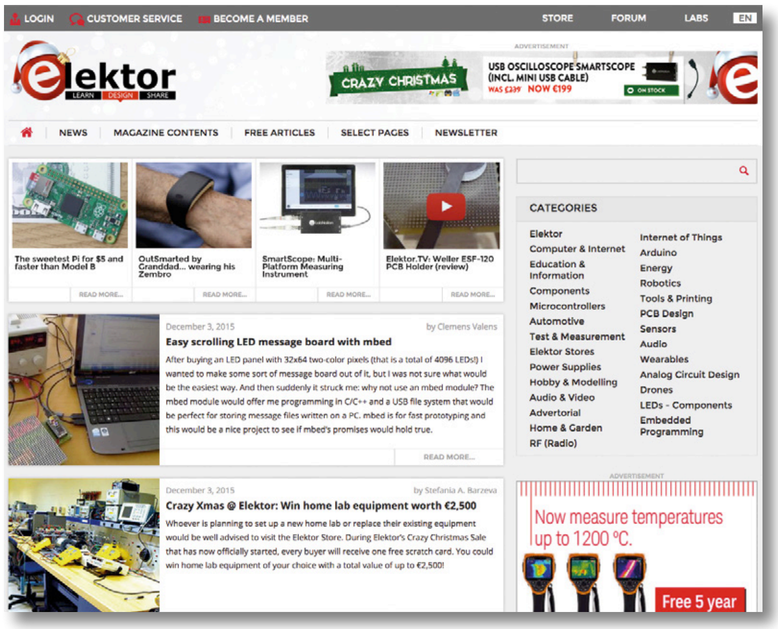
Become a member today!

	GREEN €5.67 per month £4.08 / US $6.25	GOLD €7.58 per month £5.50 / US $8.42	FREE
Elektor Annual DVD	✗	✓	✗
6x Elektor Magazine (Print)	✗	✓	✗
6x Elektor Magazine (Digital)	✓	✓	✗
Access to Elektor Archive	✓	✓	✗
Access to elektorlabs	✓	✓	✗
10% Discount in Elektor Store	✓	✓	✗
elektor weekly e-zine	✓	✓	✓
Exclusive Offers	✓	✓	✓
	www.elektor.com/green	www.elektor.com/gold	www.elektor.com/newsletter

www.elektormagazine.com

A World of Electronics News

Electronics enthusiasts can explore a world of projects, news, and movies on our completely revamped magazine website. Click on the top of the menu to choose the Dutch, English, German or French version, and use the intelligent search tools to find information and articles quickly. Sign up to our community as a GREEN or GOLD Member, and with your personal login details you will have full access to many extras such as special offers and discounts in our online store. You can also manage your account information, including your membership to the printed magazine and the Elektor weekly newsletter.



The sweetest Pi for $5 and faster than Model B

OutSmarted by Granddad... wearing his Zembro

SmartScope: Multi-Platform Measuring Instrument

Elektor.TV: Weller ESF-120 PCB Holder (review)

December 3, 2015 by Clemens Valens

Easy scrolling LED message board with mbed

After buying an LED panel with 32x64 two-color pixels (that is a total of 4096 LEDs!) I wanted to make some sort of message board out of it, but I was not sure what would be the easiest way. And then suddenly it struck me: why not use an mbed module? The mbed module would offer me programming in C/C++ and a USB file system that would be perfect for storing message files written on a PC. mbed is for fast prototyping and this would be a nice project to see if mbed's promises would hold true.

December 3, 2015 by Stefania A. Barzeva

Crazy Xmas @ Elektor: Win home lab equipment worth €2,500

Whoever is planning to set up a new home lab or replace their existing equipment would be well advised to visit the Elektor Store. During Elektor's Crazy Christmas Sale that has now officially started, every buyer will receive one free scratch card. You could win home lab equipment of your choice with a total value of up to €2,500!

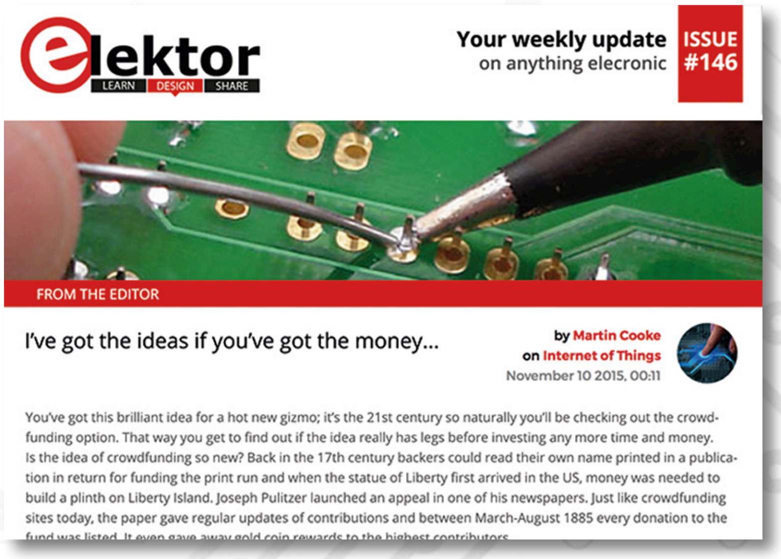
Elektor	Internet of Things
Computer & Internet	Arduino
Education & Information	Energy
Components	Robotics
Microcontrollers	Tools & Printing
Automotive	PCB Design
Test & Measurement	Sensors
Elektor Stores	Audio
Power Supplies	Wearables
Hobby & Modelling	Analog Circuit Design
Audio & Video	Drones
Advertorial	LEDs - Components
Home & Garden	Embedded Programming
RF (Radio)	

elektor Weekly & Paperless

Get a jam-packed Elektor every week

Join the more than 120 K electronics enthusiasts who receive the free & paperless Elektor. Every week you get a selection of news, tips and trends in your email inbox. You will also get special offers and discounts for the online store.

Sign up today:
www.elektor.com/newsletter


elektor LEARN DESIGN SHARE
Your weekly update on anything elecronic
ISSUE #146
FROM THE EDITOR

I've got the ideas if you've got the money...

by Martin Cooke
on Internet of Things
November 10 2015, 00:11

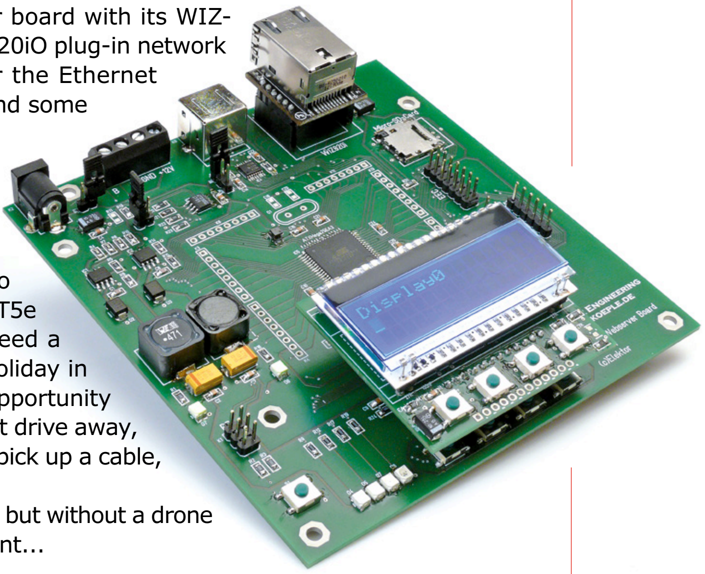
You've got this brilliant idea for a hot new gizmo; it's the 21st century so naturally you'll be checking out the crowd-funding option. That way you get to find out if the idea really has legs before investing any more time and money. Is the idea of crowdfunding so new? Back in the 17th century backers could read their own name printed in a publication in return for funding the print run and when the statue of Liberty first arrived in the US, money was needed to build a plinth on Liberty Island. Joseph Pulitzer launched an appeal in one of his newspapers. Just like crowdfunding sites today, the paper gave regular updates of contributions and between March-August 1885 every donation to the fund was listed. It even gave away gold coin rewards to the highest contributors.

Welcome to the LEARN section

By **Jens Nickel**

First… the physical hurdle

So here's the plan; today I am going to take the next step in my journey to the Internet of Things. On an overcast Whit Monday, Cloud was indeed the order of the day. I planned to hook up our old friend the popular Xmega web server board as Pilot platform to connect directly with the network (without using a PC in between as a relay station). Already I have a box full of hardware including the Xmega web server board with its WIZnet's WIZ820iO plug-in network module for the Ethernet interface and some interesting examples of demo code and necessary libraries.



In my apartment I hit the first snag; the router and my programming workspace are in opposite corners of the room. I had planned ahead but to my surprise when I unrolled the 10 m length of CAT5e network cable it was too short. I am going to need a 15 m cable and this is Whit Monday, a national holiday in Germany. Anyone with any sense is taking this opportunity to chill and the stores are closed. Cologne is a short drive away, there's bound to be somewhere there where I can pick up a cable, but where exactly?

The Internet of course offers any number of options but without a drone delivery service it looks like I will need to be patient...

Look, there's another Cloud

Clouds are all the rage now; as well as big names like Amazon, Microsoft and Google a lot of smaller operators such as the German component distributors Conrad are now offering customers the chance to load and view data stored to their own Cloud (https://conradconnect.de).

I got a text from an ex-colleague Jaime (now working at the Berlin startup Relayr) it went something like this: '*Hi Jens! Crazy how time flies… I need to drop by and meet up with you guys sometime… We are about to launch a brand new app which converts a smartphone into a kind of sensor kit, I was wondering if you wanted to give it a whirl? The app will be open-source and available on GitHub...*'

Sounds interesting… I will certainly take a look as soon as it shows up. Stay tuned, more information will be available shortly in our online edition at: **www.elektormagazine.com**!

Remote controlled Arduino

And so the learning continues: Professional writer and software specialist Tam Hanna shows in his article how you can remotely control an Arduino from a Windows smartphone or Tablet over USB or wirelessly via Bluetooth. I must admit I was not aware of the Firmata protocol; interesting that Microsoft have even provided their own .NET library for that. You can find out more on this topic elsewhere in this LEARN section. ⏮

(160012)

Tips and Tricks

From readers for readers

Here is another clever solution which can make the life of an electronics enthusiast that much easier.

Sticking pointer of a moving-coil meter #1

Contributed by Feike Hoogenbos

Recently I received an old adjustable power supply (Maxireg 761.1) made by Weir, with a large classic moving-coil meter, that had to be restored. After replacing a few components (the power supply's main filter capacitors) it could be switched on. The meter on the front panel indicated about 8 V, but turning the voltage control knob had no effect on that. The output voltage, however, did change accordingly. What was happening here?

There was a crack in the plastic housing of the meter and someone once tried to glue this, but it was very sloppy work. 'Free' on the workbench, viewing from the side it could easily be seen that the pointer was stuck to the clear front cover. Without this cover the pointer operated correctly. After asking some questions, it appears that someone had cleaned the plastic with white spirits and soap, and with this the anti-static coating was probably also removed. Good, but how to proceed?

I jumped into the deep end and bought a spray can of Kontakt 100 Antistatik. After cleaning the front cover using a clean cotton cloth with white spirits and following that with isopropyl alcohol (IPA), I sprayed both sides with Kontakt 100 and left it to dry outside.

However, after a few hours it still wasn't dry. But what was worse, it wasn't drying very smooth at all. 'Tears' of Kontakt 100 were running down the clear cover and now dust was also stuck to the surface. It all looked a bit greasy. I decided to polish it with a piece if old T-shirt. Just carefully wiping both sides dry. I worried that I would rub away the Kontakt 100 doing this, but what else could I do? This wiping worked surprisingly well and the cover became really clean and optically very clear. Now the big question: would the plastic be statically charged again?

To my great relief and pleasure the meter worked perfectly again! The operation was a complete success! Now I also understand why professional moving-coil meters use front panels with glass in them

Sticking pointer of a moving-coil meter #2

Feike Hoogenbos (again)

Not long after that an acquaintance came to me with large moving-coil meters from Monacor, probably originating from a classroom physics lab. The plastic of the meters was very dirty. He had tried to clean the front covers with warm water and dish-washing detergent, and with white spirits, but all to no avail. Subsequently he had a go with car polish (Commandant 4). With spectacular results! My acquaintance was very happy, until he put the front covers back on the meter frame. The meter pointers stuck to the window. If I had any advice please?

From my earlier experience I treated the front covers with Kontakt 100. This worked, to the extent that the static charge was gone. But the plastic did not tolerate the Kontakt 100 very well and became 'foggy'. Now what? Kontakt 100 does not dissolve in white spirits, methanol, ethanol or isopropyl, and MEK and acetone dissolve most plastics. So, car polish again. Now I was back at the beginning...

In the end, the trick that worked well: take a small cotton pad, spray some Kontakt 100 on the pad and rub the front panel with that. If you do that right, it dries quickly and is optically invisible. Apparently there is something in the propellant of Kontakt 100 that reacts with the plastic.

So, for meter pointers that stick to plastic front panels because of a static charge:

- *- remove the front panel*
- *- thoroughly clean on all sides and degrease*
- *- Spray Kontakt 100 on a cotton pad*
- *- Rub the front panel on both sides with the cotton pad*

Limit yourself to using white spirits, methanol, ethanol or isopropyl for the cleaning fluid and keep in mind that this can dissolve some paints (the text on the dial). And think about wearing gloves!

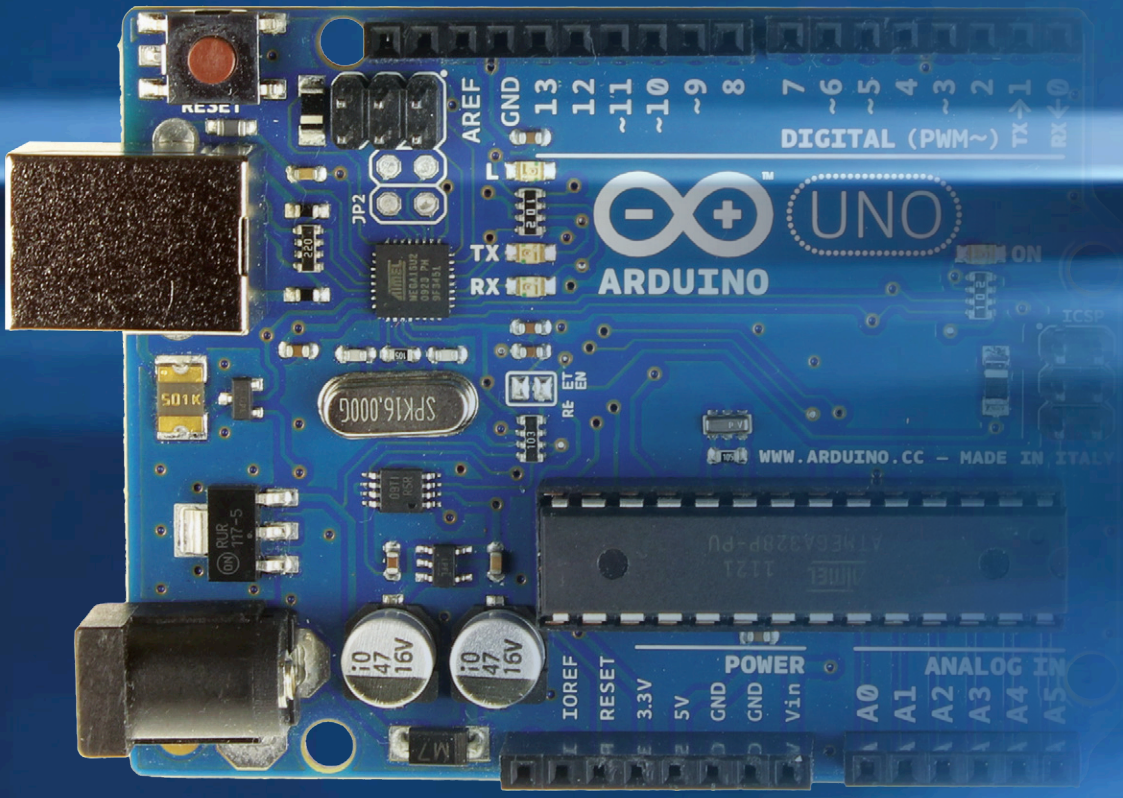
(from the Elektor forum)

(150768)

Have you come up with an inspired way of solving a really challenging problem? Or found an ingenious but 'alternative' way of using some component or tool? Maybe you've invented a better or simpler way of tackling a task? Do write in – for every tip that we publish, we'll reward you with $40 (or local equivalent)!

Windows Controls Arduino

using USB and Bluetooth



Windows smartphones have recently been gaining a noticeable market share. The new Windows Mobile 10 operating system allows you to use a Windows phone or tablet as a controller for your own projects in various ways. For example, it is possible to communicate with an Arduino board over USB or Bluetooth, and Microsoft provides a suitable library to help you.

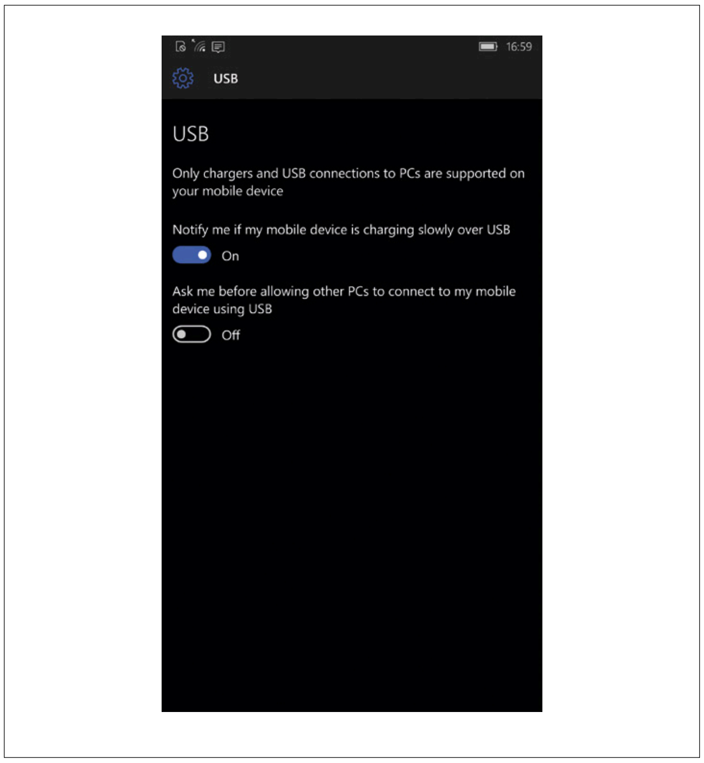
By **Tam Hanna** (Germany)

There are various methods that a developer can use to communicate with external hardware using Windows Mobile 10. The most obvious is to connect directly using a USB cable, and Microsoft makes the USB driver API available for use by third-party developers under Windows Mobile 10. However, for devices being made in small quantities and for hobby applications programming against this complex API is not really a practical approach. So Microsoft also offers an alternative programming interface to simplify the job of connecting smartphone to hardware.

Remote-controlled Arduino

A protocol called 'Firmata' [1] is popular in the Arduino world. This protocol, interestingly enough derived from the MIDI protocol, allows you to do such things as set specified port pins high and low; communicating in the opposite direction, you can receive readings from the Arduino taken using one of its analog inputs.

To this end Microsoft provides a purpose-designed 'Windows Remote Arduino' API which can also be used under Windows Mobile 10. You can use the functions in this API to build an application without knowing the underlying details of the pro-



USB

Only chargers and USB connections to PCs are supported on your mobile device

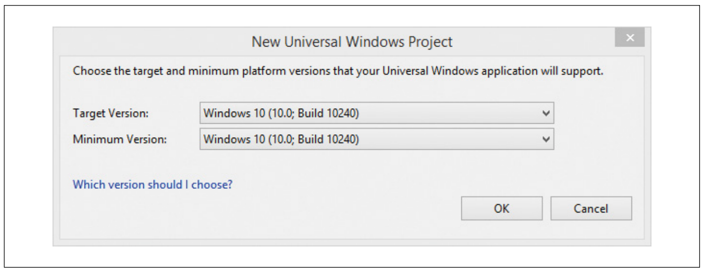
Notify me if my mobile device is charging slowly over USB

On

Ask me before allowing other PCs to connect to my mobile device using USB

Off

Figure 1. This phone does not support USB OTG.



New Universal Windows Project

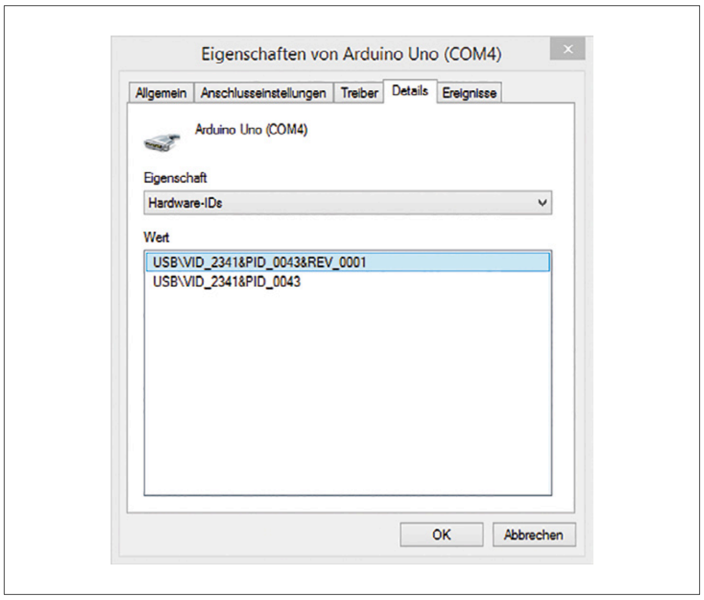
Choose the target and minimum platform versions that your Universal Windows application will support.

Target Version: Windows 10 (10.0; Build 10240)

Minimum Version: Windows 10 (10.0; Build 10240)

Which version should I choose?

Figure 2. Which version should I use?



Eigenschaften von Arduino Uno (COM4)

Arduino Uno (COM4)

Eigenschaft: Hardware-IDs

Wert:

USB\VID_2341&PID_0043&REV_0001
USB\VID_2341&PID_0043

Figure 3. Identifying the VID and PID of a USB device.

tocol. For example, to set the state of a pin on the Arduino:

```
myArduino.digitalWrite(13, Microsoft.Maker.RemoteWiring.
   PinState.LOW);
```

As we shall see (and test), the API can be used equally well to communicate over a USB cable as over a Bluetooth link. Before proceeding to read the rest of this article it would be a good idea to take a look back at the short series of articles 'Windows on the Raspberry Pi', published in Elektor from November 2015 [2]. There we looked, among other things, at how to use Visual Studio and .NET objects. As usual, we recommend that beginners start by copying our code examples below verbatim before experimenting with small changes. All the source code associated with this article can be downloaded from our website at [3].

USB cable

Let us start by trying out communication over a USB cable. For this we will need a Windows Phone 10, which offers full USB support. You can check this in the 'Settings' application: select 'Settings', and then under 'Devices' search for the entry 'USB'. On devices not offering full support a message like the one in **Figure 1** will appear.
The sequence 'Windows ➞ Universal ➞ Blank App (Universal Windows)' in Visual Studio 2015 will create a project that can run equally well on either a phone or on a PC. Our program example will thus be able to run on a PC connected to an Arduino, as long as the PC is running the Windows 10 operating system.

Sometimes Visual Studio may ask (as shown in **Figure 2**) which version of the Universal Windows Platform should be used. In such cases select the lowest-numbered version.
For a while now Microsoft has been offering its 'NuGet' package management system for Visual Studio. This can download commonly-used libraries and include them directly in the active project. Windows Remote Arduino is available in this way: click on 'Tools ➞ NuGet Packet Manager ➞ Packet Manager Console' to open the management tool. In the window that subsequently appears enter the command '`Install-Package Windows-Remote-Arduino`', which will download the necessary package from the Internet and include it in your project. Under Windows 10 permissions must be set correctly to allow communication with hardware, and this has to be done by hand. Right-click on the file 'Package.appxmanifest' and select the option 'View Code'. Change to contents of the manifest file to read as follows.

```
<Capabilities>
<Capability Name="internetClient" />
<DeviceCapability Name="serialcommunication">
<Device Id="any">
   <Function Type="name:serialPort"/>
  </Device>
  </DeviceCapability>
</Capabilities>
```

Note that a manifest file containing a 'DeviceCapability' attribute should not subsequently be edited using the graphical editor. This bug, which results in corruption of the file contents

in such cases, is known to Microsoft.
Now our preparations are complete. Obtain a USB OTG cable that will fit your phone, converting the mini-USB port of the phone to a normal USB host port. You can test the cable using a memory stick: if the stick is recognized then there should be no problems connecting the Arduino.

If you are not in a position to be able to use an original Arduino Uno then in the first instance you will need to connect the board to a PC in order to determine its VID and PID values.

Open the device manager and look for the Arduino in the lists of connections and ports. With a right click you can bring up the properties window, and the parameters we need are under 'Details ➞ Hardware IDs'. In the case of an Arduino Uno the windows will look like **Figure 3**.

Arduino code

On the Arduino side we can make use of the standard Firmata software, which is found under 'File ➞ Examples ➞ Firmata ➞ StandardFirmata'. The only change we will make is to improve the performance of the system by doubling the communications speed. Out of the box Firmata runs at a very conservative 57600 bps, which is only really called for when there are other significant loads on the processor or when using the I^{2}C functions (and see also [4]).

```
void setup()
{
  Firmata.setFirmwareVersion(FIRMATA_MAJOR_VERSION,
   FIRMATA_MINOR_VERSION);

   . . .

  Firmata.begin(115200);

   . . .

}
```

Windows code

The next step is to open the layout file of the Visual Studio project to add a label that is capable of displaying text.

```
<Grid Background="{ThemeResource
 ApplicationPageBackgroundThemeBrush}">
 <TextBlock Name="TxtStatus"/>
</Grid>
```

The Windows Remote Arduino programming environment is built in several layers. The code required for creating a connection over USB is shown in **Listing 1**.
All functions that need to be executed when the program starts up are in `MainPage()`, the constructor of MainPage. First an object of class `UsbSerial` is created: here we see why we need the VID and PID of the Arduino board. This USB object, called `myUSB`, is then passed as a parameter when creating the `myArduino` object.

If a connection error should occur a suitable message is displayed in the text label. Don't panic if this happens the first time you run the program: it should work the second time you try it. If setting up the connection succeeds we write a success message to the label and start a new thread at `runner()`. Here

Listing 1. Controlling a pin over USB.

```
public sealed partial class MainPage : Page
{
  UsbSerial myUSB;
  Microsoft.Maker.RemoteWiring.RemoteDevice myArduino;

  public MainPage()
  {
   this.InitializeComponent();
   myUSB = new UsbSerial("VID_2341", "PID_0043");
   myArduino = new Microsoft.Maker.RemoteWiring.
   RemoteDevice(myUSB);
   myUSB.ConnectionEstablished +=
   MyUSB_ConnectionEstablished;
   myUSB.ConnectionFailed += MyUSB_ConnectionFailed;
   myUSB.begin(115200, Microsoft.Maker.Serial.
   SerialConfig.SERIAL_8N1);
  }

  private void MyUSB_ConnectionFailed(string message)
  {
   TxtStatus.Text = „Connection failed“;

  }

private void MyUSB_ConnectionEstablished()
{
   TxtStatus.Text = "Connection to Arduino!";
   runner();
}


async private void runner()
{
   await Task.Run(() => innerRunner());
}

void innerRunner()
{
   myArduino.pinMode(13, Microsoft.Maker.RemoteWiring.
   PinMode.OUTPUT);
   for (;;)
   {
    myArduino.digitalWrite(13, Microsoft.Maker.
   RemoteWiring.PinState.LOW);
    myArduino.digitalWrite(13, Microsoft.Maker.
   RemoteWiring.PinState.HIGH);
    myArduino.digitalWrite(13, Microsoft.Maker.
   RemoteWiring.PinState.LOW);
    myArduino.digitalWrite(13, Microsoft.Maker.
   RemoteWiring.PinState.HIGH);
   }
}

}
```

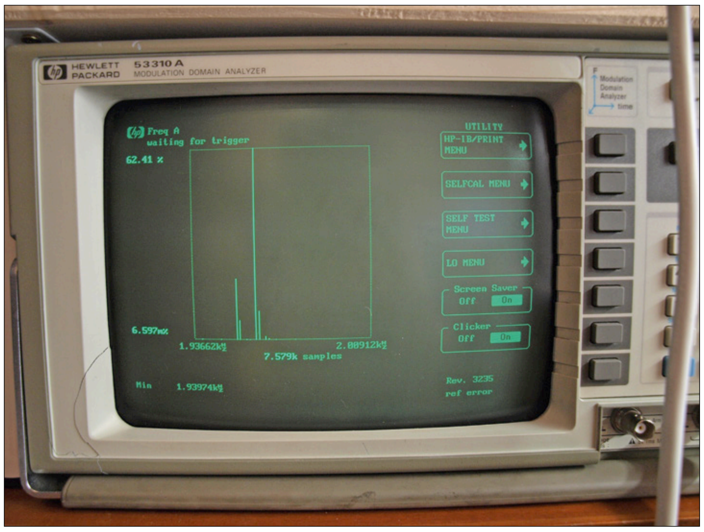


Figure 4. The squarewave output is reasonably stable.

Win10 only!

Even though the documentation for Windows Remote Arduino might lead you to believe otherwise, the product only works reliably under Windows 10: this advice comes to you as a result of the author's bitter experience trying to use Windows 8.1 as the target platform. As a host platform for Visual Studio, on the other hand, Windows 8.1 works fine as long as you have a physical Windows Phone 10.

***Et tu*, Yún?**

An Arduino Yún, with its built-in WLAN radio module, is the easiest way to get going with Windows Remote Arduino. Unfortunately a certain amount of configuration is needed, described in detail in [7].

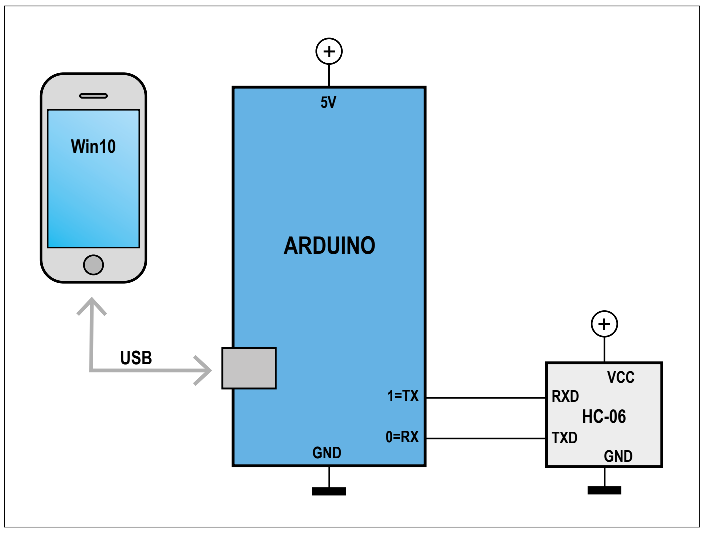


Figure 5. It is easy to add a Bluetooth radio to an Arduino.

pin 13 is taken alternately high and low, generating a square wave. Behind the scenes everything works in terms of command bytes sent to the Arduino, but this need not concern us as application developers. The API implemented in Windows Remote Arduino is closely related to the normal Arduino API, and the functions are very similar in appearance to those found in a normal Arduino sketch.

A Windows Mobile device will only accept our code if we switch it to developer mode under 'Settings ➞ Update and Security ➞ For Developers'. Now deploy the control program to the phone, taking care to ensure that the 'Architecture' field is set to the right value for the target platform.
When the program is launched the connected Arduino should emit a square wave. **Figure 4** shows the result on the author's MDA (modulation domain analyzer).

Bluetooth for the Arduino

Bluetooth works well on Windows Phone, and allows communication with the Arduino to be set up with the help, for example, of an HC-06 module of the type cheaply available from AliExpress. The set-up is so simple that a circuit diagram (**Figure 5**) is almost superfluous.

On account of a small incompatibility it is necessary at this point to downgrade the firmware library. Open the library manager in the Arduino IDE and install version 2.4.4: this works around the bug documented at [5]. It is then simply a matter of adapting the code to suit the baud rate of the module, which by default is 9600 bps.

```
void setup()
{
  Firmata.setFirmwareVersion(FIRMATA_MAJOR_VERSION,
   FIRMATA_MINOR_VERSION);
  . . .

  Firmata.begin(9600);

  . . .

}
```

Bluetooth under Windows

A new library is required under Windows Phone to allow access to Bluetooth. Its entry in the file 'Package.appxmanifest' again needs to be made by hand, as the editor does not handle the inner 'DeviceCapability' attribute correctly.

```
<Capabilities>
<Capability Name="internetClient" />
  <DeviceCapability Name="bluetooth.rfcomm">
<Device Id="any">
<Function Type="name:serialPort"/>
</Device>
  </DeviceCapability
</Capabilities>
```

The program running on the Windows Phone calls the function `btScanner()` from within the constructor `MainPage()` (see **Listing 2**). Instead of the USB serial object we used previously we here have an instance of `BluetoothSerial`, which

Listing 2. Establishing a Bluetooth connection.

```
BluetoothSerial myBTSerial;
Microsoft.Maker.RemoteWiring.RemoteDevice myArduino;
public MainPage()
{
  this.InitializeComponent();
  btScanner();
}


async public void btScanner()
{
  var radios = await Radio.GetRadiosAsync();
  var bluetoothRadio = radios.FirstOrDefault(radio =>
   radio.Kind == RadioKind.Bluetooth);



  if (bluetoothRadio != null && bluetoothRadio.State ==
   RadioState.On)  {
   TxtStatus.Text = “Scan!”;
   DeviceInformationCollection aColl = await
   BluetoothSerial.listAvailableDevicesAsync();
   TxtStatus.Text += “\n Result: “ + aColl.Count().
   ToString();
   if (aColl.Count() > 0){
    myBTSerial = new BluetoothSerial(aColl[0]);
    myArduino = new Microsoft.Maker.RemoteWiring.
   RemoteDevice(myBTSerial);
    myBTSerial.ConnectionEstablished +=
   MyBTSerial_ConnectionEstablished;
    myBTSerial.ConnectionFailed +=
   MyBTSerial_ConnectionFailed;
    myBTSerial.begin(9600, Microsoft.Maker.Serial.
   SerialConfig.SERIAL_8N1);
   }
  }
  else  {
   TxtStatus.Text = “No BT-Module!”;
  }
}
```

is designed for use with short-range radio communications. The function `btScanner()` serves many purposes. In its first part we use the `GetRadiosAsync()` method to generate a list of all radio modules within range of the phone. If the standard radio module is nearby and switched on, we can then proceed to the next step of the scanning process. The object `radios` returned by `GetRadiosAsync()` has the interesting property that if we inspect it in the debugger it appears to be empty: its contents only materialize on the display during enumeration. In our code we limit ourselves to selecting the first module that we see and then we attempt to set up communication with it. Before executing the program you must pair the module with the Windows Phone 10. The search code implemented in Windows Remote Arduino will only find radio modules that have already been successfully paired. The ID code required to pair with the module is set by default to 1234.

Microsoft’s Bluetooth stack does have a few foibles, which in conjunction with the not always particularly reliable firmware running on the radio module can make for some entertaining debugging sessions. The first line of defense is to reboot the module; if that doesn’t do the trick, then the phone itself will also have to be rebooted.

The remainder of the code on the Windows side is the same as in the USB example. Again, the code takes pin 13 alternately high and low.
Figure 6 shows the output signal as displayed on the MDA. It is clear that the wireless connection suffers more from latency problems than the wired connection.

Brass tacks

The Firmata library makes things a lot easier for developers who would prefer to avoid getting into low-level technical details, but if you want the absolute maximum performance from the hardware you will have to look at implementing your own protocol. We will look now at what is involved.
For motivation, and to get a better understanding of the problem, we will look at an example using an OLED display controlled over SPI: sharp-eyed readers will recognize the 0.96 inch display from the March & April 2016 issue of Elektor [6] as an old friend. Connect it to the Arduino (which is in turn connected to the Bluetooth module) according to the circuit shown in **Figure 7**.

Because this task is rather more complicated we will take the opportunity to create a new project. As in the previous code examples, the constructor again calls the asynchronous method `runner()`, which is responsible for the actual processing and data communication (see **Listing 3**).

After the somewhat modified scanning code we open a `StreamSocket`. Under Windows Universal Platform this class is also responsible for serial connections with external hardware. Once the connection has been successfully established we create

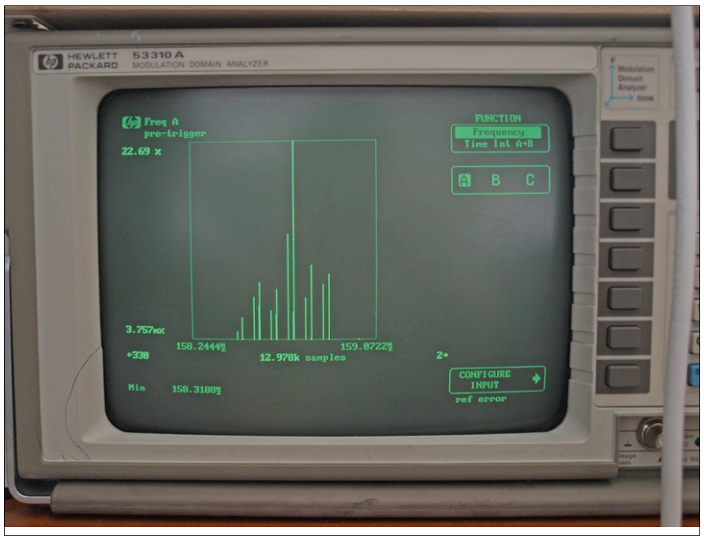


Figure 6. Bluetooth has more problems with latency than a wired connection.

Listing 3. Sending characters from the smartphone.

```
public sealed partial class MainPage : Page
{
RfcommDeviceService myService;
StreamSocket mySocket;
DataWriter myWriter;

public MainPage()
{
this.InitializeComponent();
runner();
}

async void runner()
{
DeviceInformationCollection dIC = await
   DeviceInformation.FindAllAsync(RfcommDeviceService.
   GetDeviceSelector(RfcommServiceId.SerialPort));
myService = await RfcommDeviceService.
   FromIdAsync(dIC[0].Id);
DeviceInformation a = dIC[0];
mySocket = new StreamSocket();
try
{
await mySocket.ConnectAsync(myService.
   ConnectionHostName, myService.ConnectionServiceName);
byte[] thisChar = { (byte)1, (byte)‘H‘, (byte)‘e‘,
   (byte)‘l‘, (byte)‘l‘, (byte)‘o‘, (byte)0 };
myWriter = new DataWriter(mySocket.OutputStream);
while (1 == 1)
{
myWriter.WriteBytes(thisChar);
Task<UInt32> aTask = myWriter.StoreAsync().AsTask();
await aTask;
await Task.Delay(250);
}
}
catch (Exception e)
{
e = e;

}
}
}
```

Listing 4. Receiving characters on the Arduino.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_MOSI   9
#define OLED_CLK   10
#define OLED_DC    11
#define OLED_CS    12
#define OLED_RESET 13
Adafruit_SSD1306 display(OLED_MOSI, OLED_CLK, OLED_
   DC, OLED_RESET, OLED_CS);

char mode=0;
char readCtr=0;
char myField[32];

void setup() {
  Serial.begin(9600);

  display.begin(SSD1306_SWITCHCAPVCC);
  display.clearDisplay();
  delay(2000);
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println(“Online!”);
  display.display();
}

void loop() {
  if(Serial.available())
  {
    switch(mode)
    {
      case 0: //Warte auf LOS
        if(Serial.available())
        {
          char val=Serial.read();
          if(val==1){
            mode=1;
            readCtr=0;
          }
        }
          break;
      case 1: //Lese und speie aus
        if(Serial.available())
        {
          char val=Serial.read();
          if(val==0){
            //Ende

            myField[readCtr]=val;
            display.clearDisplay();
            display.setTextSize(2);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println(myField);
            display.display();
            readCtr=0;
            mode=0;
          }
          else
          {
            myField[readCtr]=val;
            readCtr++;
          }
        }
          break;
    }
  }
}
```

the bit stream to be transmitted. The stream is then written to the socket connected to the Arduino using a `DataWriter` class.

The process, which might seem unnecessarily convoluted at first glance, is necessary because the writing of the data passed to `WriteBytes()` only happens within the `AsyncTask` returned by `StoreAsync()`. In the next step we wait for 250 milliseconds to give the Arduino enough time to receive the data we have sent. This is essential as otherwise it is possible for the buffer to overflow, which has the undesirable consequences shown in **Figure 8**.

At the end of the code we see another little .NET wrinkle. The assignment `e = e` makes it easier to set a breakpoint if we ever want to analyze the causes of an exception that might occur.

Arduino sketch

Now we can turn to the sketch running on the Arduino Uno. The driver library we need for the display is available for download from within the library manager under the name 'Adafruit SSD1306'. For reasons of convenience we will also make use of the Adafruit GFX library, which offers a range of graphics primitives.
Within the `setup()` method we use commands in the library to set up a connection with the display module (see **Listing 4**). The meat of the code, meanwhile, is within the `loop()` method, implemented in the form of a simple finite state machine. To explain: a finite state machine is a program that switches between a number of operating states. In our case the machine is either in the state of waiting for a '1' start bit or in the state of receiving bits, buffering them, and then passing them to the display.

We are now ready to try out this example. If you see an error along the lines of 'No more data is available. (Exception from HRESULT: 0x80070103)', try rebooting the phone and/or the radio module.

Conclusion

The poor reputation of Windows Mobile 10 is undeserved: as we have seen, communication using Bluetooth is by and large reliable enough. In this article we have only considered sending data from the phone to the Arduino: communication in the reverse direction is also possible, both using USB and using Bluetooth.

(150763)

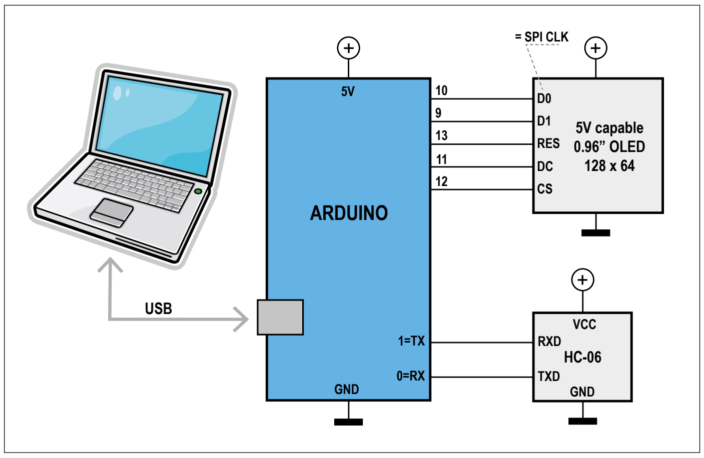


Figure 7. SPI and serial interfaces team up.

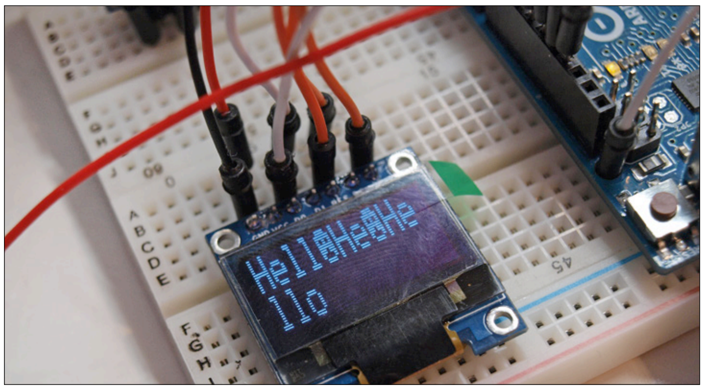


Figure 8. The result of trying to send too many bytes.

Web Links

[1] www.arduino.cc/en/Reference/Firmata
[2] www.elektormagazine.com/150465
[3] www.elektormagazine.com/150763
[4] https://sourceforge.net/p/firmata/mailman/message/22824115/
[5] https://github.com/ms-iot/remote-wiring/issues/80
[6] www.elektormagazine.com/150520
[7] https://create.arduino.cc/projecthub/mohanp/windows-remote-arduino-yun-wifi-and-networkserial-b3290a

It doesn't work!

When an application based on Windows Remote Arduino resolutely refuses to make contact with the Arduino, it is a good idea to use the 'Windows Remote Arduino' program available in the app store to verify correct operation of the hardware and phone, and that you are using the correct version of the Firmata library.

It's too slow!

The HC-06 module runs by default at just 9600 bps. If you would like things to run a little faster you need to send the right AT command to the module. More information on this can be found in the module's documentation.

CIRCUITMAKER Tips & Tricks (2)

By **Neil Gruending** (Canada)

Last time we looked at Altium CircuitMaker and some of its unique features. Now let's make a project and start a design to see how it works.

Creating a Project

A CircuitMaker project is a repository for all of the files needed to create a design. Projects can also be shared and have team members like we saw in Part 1. All of this possible because everything is stored online in the Community Vault with everything cached locally on your computer as well.

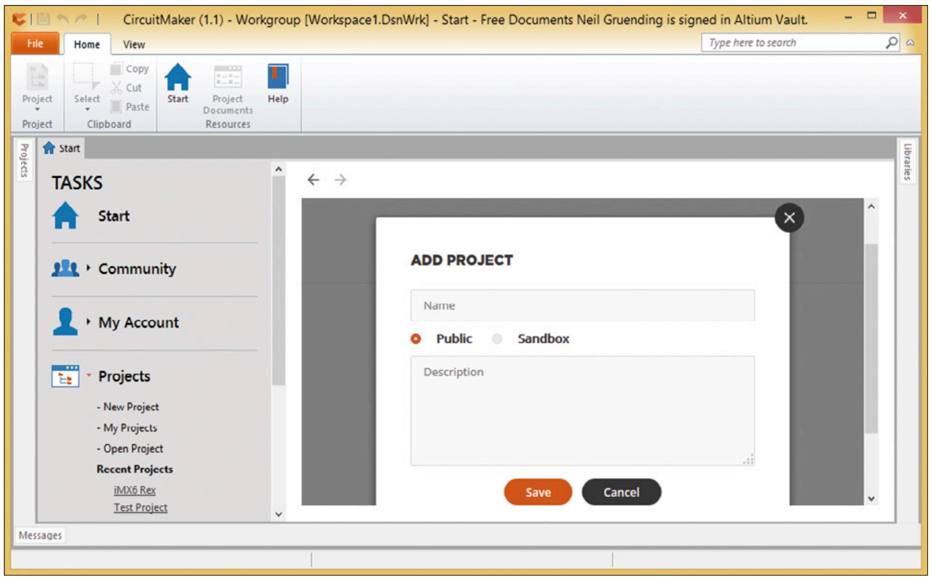


Figure 1. Adding a project.

The first step to creating a project is to log into your CircuitMaker account and then you will see the Start tab like in **Figure 1**. Click on the Projects task and then you can create a new project by clicking on New Project. That will open the Add Project window where you name your project and add a description. You can also choose if this will be a public project or a private one that gets stored in a private area of the CircuitMaker Vault called a Sandbox. We will be using a sandboxed project for this example but it's easy to make it public later.

The Projects task is also where you can access all of your projects or other public ones store in the Vault. My Projects will show all of your projects and let you edit their details. Open Project will show all of the public projects in the Vault so you can find the project you want to open. Recent Projects will show a short list of projects that you've previously opened so that you can click on them and open them right away. Once you open a project you will see something similar to **Figure 2** where I made a project called Elektor.

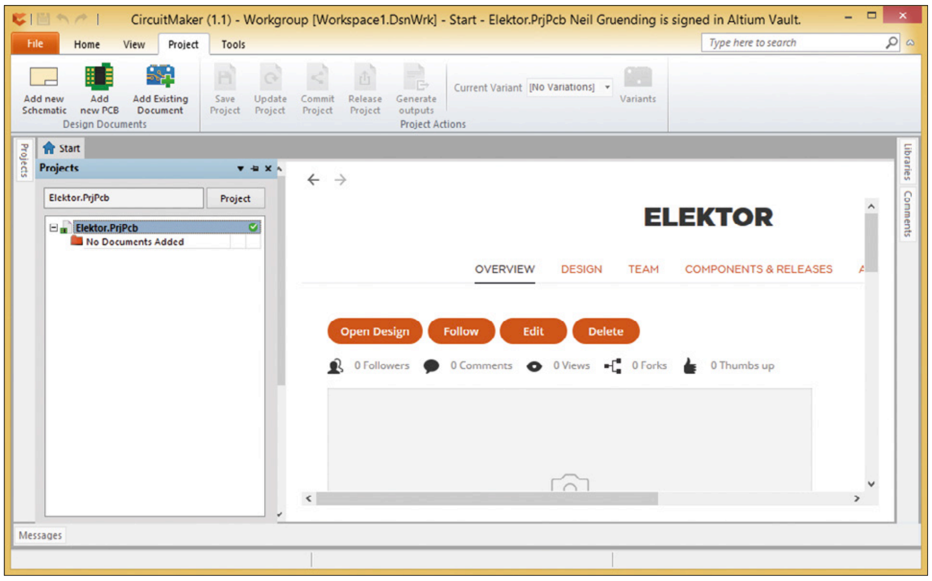


Figure 2: Open a project.

Creating a Schematic

Next we need to add a schematic to the design using the "Add new Schematic" button. You will be prompted for a file name for it and then CircuitMaker will add a blank schematic sheet to the design. Note that CircuitMaker stores one schematic page per file, which is different from other packages so I try to give pages file names based on their function (Power, MCU, etc) to make it easier to navigate a design. You can add as many pages as you want because CircuitMaker will keep track of them in the project file.

The new schematic sheet will open automatically and then you will need to click on the Home tab in the ribbon bar like in **Figure 3**. This is where you will spend most of your time in CircuitMaker because it has the most commonly used tools all in one place. For example all of electrical connections in a schematic are edited with the tools in the Circuit Element group. You can also add Graphical Elements that don't affect the electrical connectivity and the Collaboration Elements tool group lets you add comments to a design while you work on it with a team of people.

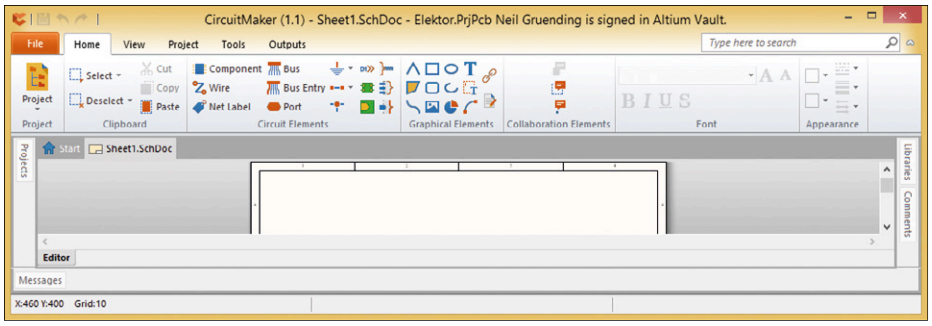


Figure 3: The Home ribbon bar.

Now let's add a component to the schematic by clicking on the Component tool in the Circuit Elements group. This will pop up the Place Component window that contains the last component that you used already selected. Click on the Choose button and the Browse Libraries window will open like in **Figure 4** where you can choose the library to search and then put the component you are searching for as the search mask.

In this example I searched for a MMBT3906 PNP transistor and once I chose the part CircuitMaker loaded a preview of the schematic symbol and the 3D model of the PCB footprint. Pressing OK will insert it into the schematic.

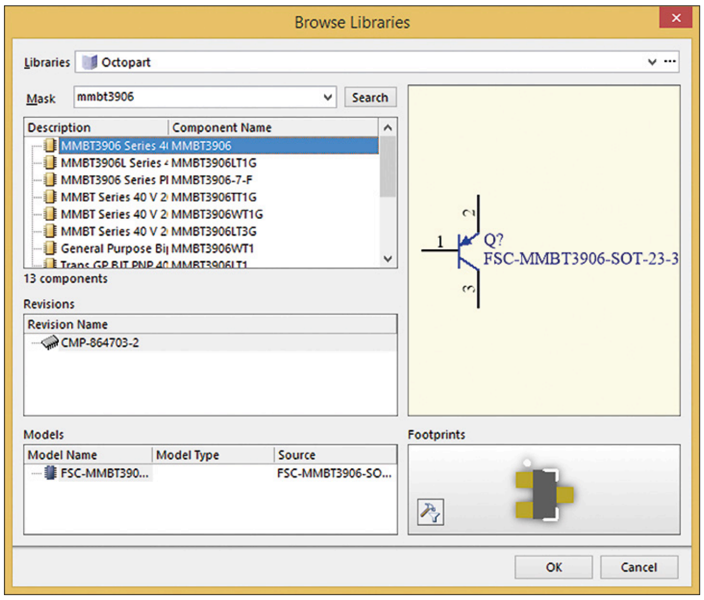
That's a quick method to add components to a schematic but what if the component you want isn't listed or isn't exactly how you want it? You can't change a component using the Component tool but you can use the Libraries window shown in **Figure 5** which is opened by pressing the Libraries button on the far right hand side of the screen.

Click on "Build your own" if you want to create a new component that isn't already in the library. CircuitMaker will then create a new blank component for you to modify as necessary. Once you commit the component into the Vault it becomes public for everyone to use.

You can also modify an existing component by either right clicking it and selecting Edit or by pressing the Build button. Editing a component will load all of the existing component data for you to edit whereas building a component will only load the basic component information without its models (schematic symbol, PCB footprint, etc). Either way CircuitMaker will create a new version of the component when it's committed back into the Vault. This way none of the previous information is lost and any projects that use the component can refer to the version that they want to use.

Once you have placed the components that you want to use you can use the Wire and Bus tools to electrically connect them as needed to complete the design. CircuitMaker also has keyboard shortcuts available for most common operations to speed up the process which are listed in the Shortcuts panel (View ➞ Help group ➞ Shortcuts). Then you will be ready to lay out the board which we will look at next time. ⏮

(150813)


Browse Libraries

Libraries: Octopart

Mask: mmbt3906 Search

Description	Component Name
MMBT3906 Series 4(	MMBT3906
MMBT3906L Series ‹	MMBT3906LT1G
MMBT3906 Series PI	MMBT3906-7-F
MMBT Series 40 V 2(	MMBT3906TT1G
MMBT Series 40 V 2(	MMBT3906WT1G
MMBT Series 40 V 2(	MMBT3906LT3G
General Purpose Bi(	MMBT3906WT1
Trans GP BJT PNP 40	MMBT3906LT1

13 components

Revisions

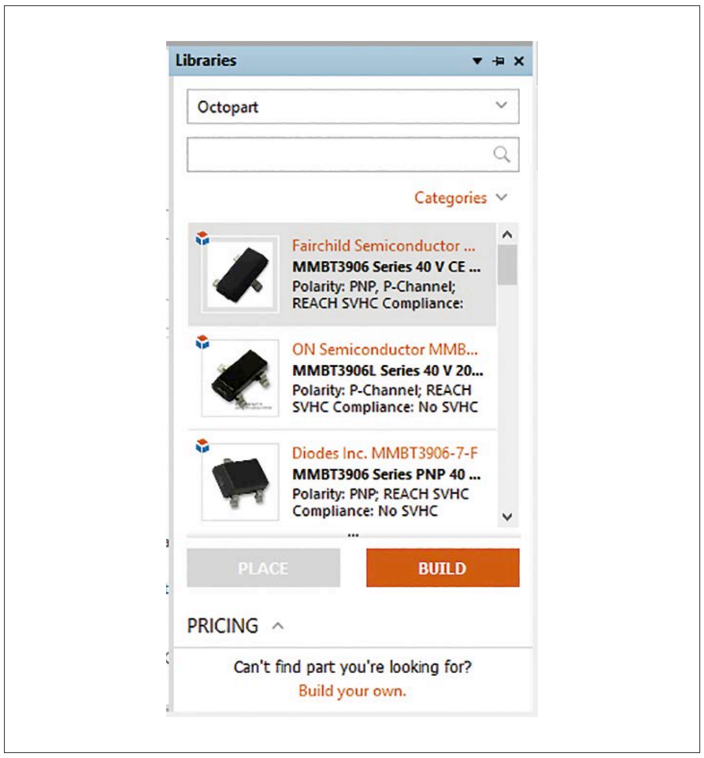
Revision Name
CMP-864703-2

Models

Model Name	Model Type	Source
FSC-MMBT390...		FSC-MMBT3906-SO...

Footprints

Figure 4. The Browse Libraries window.


Libraries

Octopart

Categories

Fairchild Semiconductor ...
MMBT3906 Series 40 V CE ...
Polarity: PNP, P-Channel; REACH SVHC Compliance:

ON Semiconductor MMB...
MMBT3906L Series 40 V 20...
Polarity: P-Channel; REACH SVHC Compliance: No SVHC

Diodes Inc. MMBT3906-7-F
MMBT3906 Series PNP 40 ...
Polarity: PNP; REACH SVHC Compliance: No SVHC

Can't find part you're looking for?
Build your own.

Figure 5. The Libraries window.

LoRa™ a Concise Introduction

Futuristic technology for M2M applications

By **Norbert Schmidt**, IMST Ltd (Germany)

The Internet of Things (IoT) is the next stage towards the all-embracing networking of machines and devices to one another — what people call M2M or machine-to-machine communication. Smart applications running on small, battery–powered devices will in future unburden life and also satisfy new demands that will arise using the underlying radio technology. A significant criterion is the range of the wireless technology employed. In this respect classic short-range wireless systems like WLAN or Bluetooth provide a range of only a few meters, whilst classic ISM (industrial, scientific and medical) band radio can manage up to a kilometer. LoRa is a solution that reaches out still further.

Plenty of extremely low-cost end-nodes are available for M2M communication. It follows that the wireless technology employed with them needs to be priced equally keenly. Unfortunately WLAN technology or cellular/mobile radio are both generally too expensive for RF (radio frequency) modules to be offered for use on the classic license-free ISM frequency bands.
Now, however, a new wireless technique called LoRa™ (an abbreviation for 'long range'), developed and patented by the Semtech company, offers a solution that is both economically viable and offers greater range. In contrast to the classic wireless systems on the ISM bands, LoRa delivers ranges up to 15 kilometers (approx. 10 miles) under line-of-sight conditions. In this way you can close the door on mobile communications and significantly extend the scope of wireless for M2M applications.

Improved receivers

A yardstick for evaluating the quality of a receiver for achieving this kind of range is to assess not only the permitted transmitter power but also the so-called 'sensitivity' of the receiver. The latter indicates the lower threshold of the input efficiency obtainable for 'normal' reception. The sensitivity of many short-range wireless systems lies in the magnitude of around –100 dBm, thus –90 dBm for Bluetooth and –100 dBm for Zigbee. If you can raise the sensitivity, then greater distances can be covered with the same transmit power. With LoRa technology Semtech has managed to increase the sensitivity of ISM band receivers up to –137 dBm, at the same time extending the range significantly.
This increase in sensitivity is achieved using a band-spread technique. The useful signal is transmitted using not the minimum bandwidth required but with an appreciably greater bandwidth. For this the signal is imprinted with a special 'signature'; the receiver executes a correlation with this signature. In this process the bandwidth is reduced back to the wanted bandwidth but the receive power per data-bit is raised by quite a few dB. Available implementations come close to the performance of the wireless chip given in the data sheets (SX1272, SX1276). **Figure 1** shows the sensitivity of the LoRa RF module iM880A-L from **Figure 2** compared against the values obtainable theoretically according to the data sheet applied to the spread factor. You can see that good compliance with the specifications in the data sheets can be observed, particularly with large spread factors.
The relationship between bandwidth and bit-rate need not be fixed; it can be varied for each transmission or for every communication channel. An important characteristic of LoRa

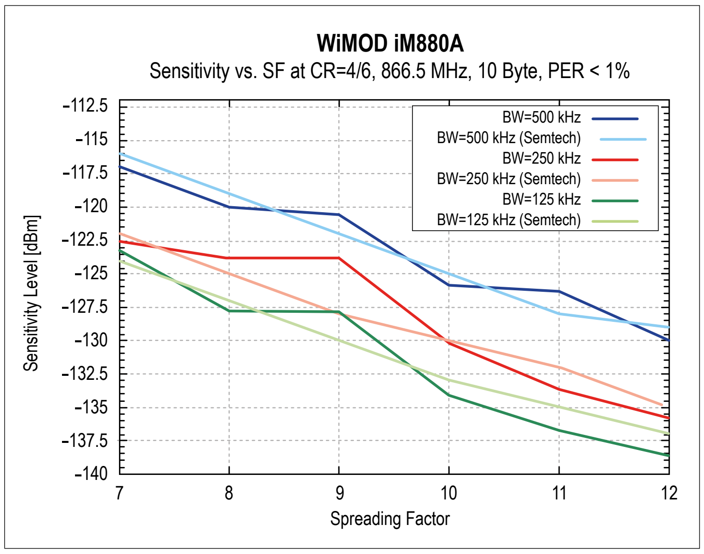


Figure 1. Sensitivity of the iM880A RF module as a function of bandwidth and spread factor.

wireless transceivers is consequently the separation (disconnect) between bandwidth and bit-rate. By means of the spread factor used for this band-spreading the relationship between bandwidth and bit-rate can be adjusted flexibly. This innovation using high spread factors makes it possible to span distances of more than 15 kilometers at low bit-rates. **Figure 3** shows the dependency of the bit-rate on the selected spread factor, whereby larger spread factors are generally applied for greater distances.
A communication channel for an end-node is characterized by the frequency that has been set and the spread factor — or put another way, the signature of the band-spreading. Several transmissions with differing signatures can take place simultaneously, as the signatures in the wireless ether are orthogonal to (statistically independent from) one another. Orthogonality ensures that during the correlation process in the receiver only information corresponding to its own signature is validated; received signals carrying other signatures are ignored and go into the bit bucket. With three specified bandwidths (125 kHz, 250 kHz, 500 kHz) and seven possible spread factors (SF6, SF7 up to SF12) 21 potential operating points can arise, resulting in nominal bit-rates from 290 bit/s to 37.5 kbit/s.

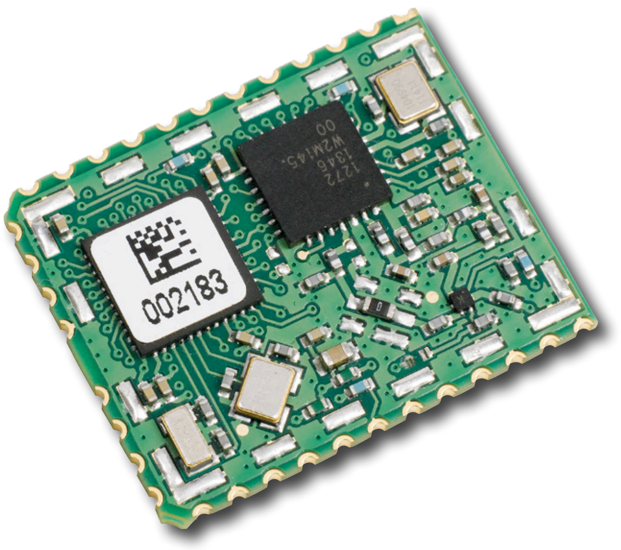


Figure 2. LoRa-RF module iM880A-L made by IMST Ltd.

Carrier frequencies for LoRa wireless technology lie in the European ISM band at 868 MHz. Other countries also have frequency ranges available around 915 MHz, as well as at 433 and 477 MHz.

Concentrator and spread factor

With LoRa Technology we can have wireless networks in which one single cell can cover an area of many square kilometers. One cell can have hundreds of end-nodes, albeit with all of them in this cell harmonized and administered. A centralized communications node located at the physical middle point of the cell — the so-called ‘concentrator’ forming the core component of a LoRa gateway — provides this through its own special architecture. Being the centre point of a star network, it needs to be able to receive multiple channels at the same time and in the process handle differing distances and data-rates. For constructing concentrators the Semtech company provides — just as it does for the sensor nodes — integrated wireless ‘building blocks’ (SX1255 / SX1257 / SX1301), which —thanks to their specialized and (compared with the end-nodes) more complex architecture — are able to achieve multiple reception in parallel.

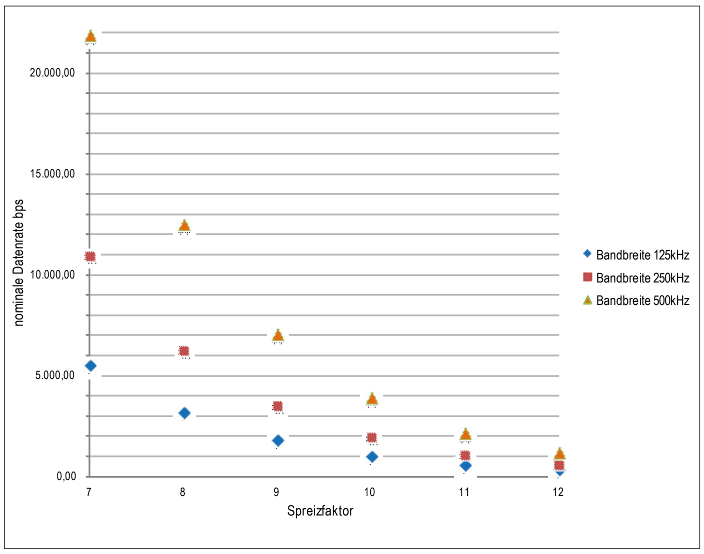


Figure 3. Schematic representation of the ratio between bit-rate and spread factor.

The concentrator from the IMST company is based on (among other things) the SX1301 chip from Semtech. For Europe this is certified according to the Radio and Telecommunications Terminal Equipment Directive (R&TTE) and offers extensive capabilities for media access/capture and networking. With a sensitivity of up to -138 dBm and maximum output power of 20 dBm, it fully supports the wide coverage range feasible with LoRa. Ten channels can be demodulated simultaneously and independently of each other. Eight of these paths are intended for LoRa reception with a bandwidth of 125 kHz. They support all the spread factors (SF7 to SF12) used in the LoRaWAN standard, without requiring the need to carry out fixed channel allocation. The orthogonality technique employed in the spread sequences used means that data packets sent with differing spread sequences can be received simultaneously on a given frequency. This makes possible genuine temporally-simultaneous reception of communications at different terminal devices.

The ninth communication path is a LoRa transmit-receive channel for wider bandwidths (250 kHz and 500 kHz). Here we do need to configure a fixed spread factor to be used. The tenth path is a transmit-receive channel using conventional (G)FSK modulation, like that standardized currently on ISM frequency bands. Thanks to the orthogonality of the spread sequences used, different users can share a single frequency so long as they each encode their datastreams with differing spread sequences. This makes it possible to have large wireless cells with hundreds of users.

The star-network structure with a centralized communications nexus at the center offers major advantages for the management of end-nodes. Whereas shared networks carry the burden of heavy protocol overheads, this can be reduced to a minimum if there is a single central control point and direct access to the concentrators through the end-nodes. This serves for optimal use of the available wireless capacity and to appreciably smaller power consumption at the end-nodes. This is a major advantage, since the end-nodes are often battery-powered.

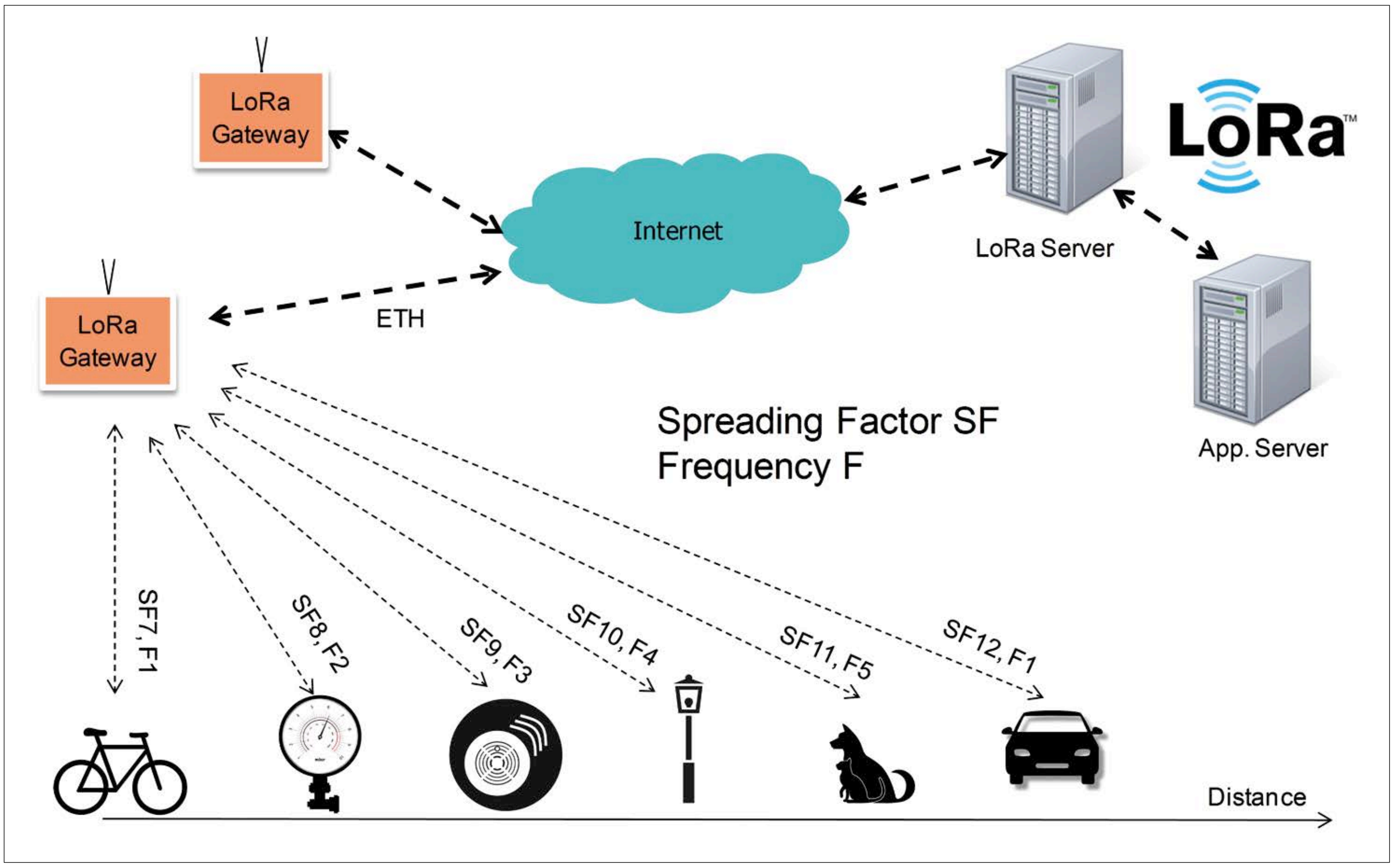


Figure 4. LoRaWAN network, comprising terminal devices, gateway, network and application server.

Software and protocols

It goes without saying that the software of the system that implements the protocol stack must reflect the novel capabilities of the air interface in an appropriate way. Media access must be optimized with respect to maximum capacity and the number of collisions in the radio channel and consequently the cumulative interference to users must be minimized for fault-free reception. This *inter alia* is achieved by constantly optimizing the output power and spread factors for the current wireless traffic level situation and determined specifically according to the bit-rates demanded and the distance between the node and the center of the star network. The medium access (MAC) layer solves this problem by means of the adaptive data rate (ADR) settings.

To fulfill the regulatory boundary conditions optimally techniques are also under consideration for adaptive frequency change ('adaptive frequency agility' or AFA) together with 'listen before talk' (LBT). In this way the current limitations on the duty cycle, resulting from the regulation, could be mitigated in the future.

Cells, gateways and networks

In future communication systems using LoRa technology should be able to enhance the mobile networks in an ideal manner — there are no costs for data transfer, both the end-nodes and the concentrators are very cost-effective, and distances in the order of magnitude of today's mobile radio cells are easily spanned. Consequently the cell topology of LoRa systems is a good match for those of the mobile radio operators. **Figure 4** illustrates the typical architecture of this kind of network. Gateways made up of concentrators facilitate connection to the Internet, using either cellular technology, DSL, Ethernet or WLAN links.

The most diverse variety of applications can profit by LoRa wireless technology therefore. Sensor systems call for moderate bit-rates over long distances and to name just a few application areas, we have the agricultural sector, industry, logistics, environmental monitoring, consumption recording, the intelligent city and the intelligent home.

The wide area that can be covered using a single concentrator means that multiple applications can be served within the range of this concentrator. It can be assumed that inside a single cell there will be dissimilar types of sensor and also diverse users. There arises here a major opportunity for standardizing the communications technology and not just this but also the supporting systems technology. It would be ideal if one single concentrator could act as gateway for the various users and make available infrastructure where users' data could be held individually and separately in a data bank and called up as required.

Desirable coordination

The mutual interaction and compatibility of the systems is being developed and standardized by a new harmonization body called the LoRa™ Alliance (www.lora-alliance.org). Semiconductor firms, manufacturers of wireless products, software firms, mobile network operators, IT firms and test houses are all collaborating in this alliance to adopt a harmonized standard for a LoRa eco-system.
Since the formation of the LoRa Alliance at the Mobile World Congress in March 2015 the number of Alliance members has developed from an initial 29 to a gigantic 225 undertakings already (situation at 22.2.2016). One reason for this has surely to be the growth in LPWAN applications forecast by analysts for the next few years.
The LPWAN (Low Power Wide-Area Network) concept embraces a group of technologies that, on account of their large wireless range, can create communications networks with 'wide area' coverage and simultaneously enable individual end-devices to operate at 'low power' for several years using batteries. These two stipulations (long range and low power) result in relatively low data rates that can be set up according to the actual application. A third stipulation for cost-effective solution components ('low cost') is consequently a significant criterion for the maximum possible take-up of a technology. LoRa wireless technology is a promising candidate for LPWAN implementation, particularly in combination with comprehensive standardization and harmonization.

Classes of device

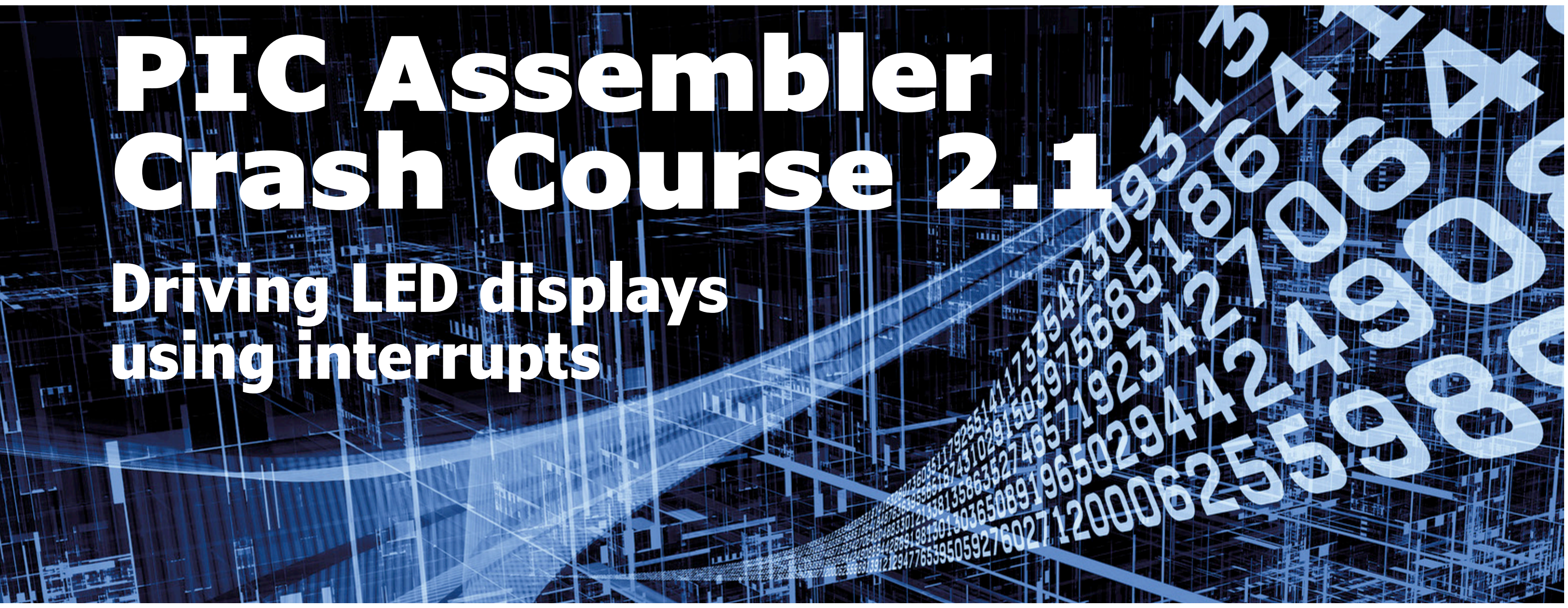
The use of star-type topologies enables the protocol overheads (in the form of a simple network layer) to be kept small and all of the system complexity to be shifted away from end-devices into the center point of individual star networks. This makes possible simple and cost-effective terminal devices. Within the LoRaWAN standards there are currently three classes of device defined (Class A, B, C). Class A terminal devices access the wireless channel to send their wireless data packets on an event-determined basis (e.g. on receipt of a sensor reading). Within these packets a requirement for acknowledgment of receipt can be included ('confirmed' data packet). Following the transmit cycle, the terminal device opens two successive time windows in order to receive the acknowledgment message and other possible data from the central gateway/star nexus. Class B terminal devices behave like Class A ones but are also in a position, thanks to a time-synchronous 'beacon' signal sent out by the gateway, to negotiate additional receive-windows with the server/gateway. Class C terminal devices are generally not battery-powered and can be configured to listen constantly in receive mode.
To warrant this standard the first test houses are already offering testing of Class A devices (Classes B and C will follow later). The test cases of OTAA (Over The Air Activation), ABP (Activation By Personalization) and Frequency Agility are already covered and recognized in a test report. As far as any faults or other shortcomings are identified during the test procedure, these can be remedied during the testing process.

Once certified, the various applications can be introduced seamlessly into LoRaWAN networks. In this way an ideal platform has been created for the efficient and compliant implementation of various applications in a network with high and consistent quality. In the next few years a large increase in LoRaWAN applications is expected on the equally rising number of networks. ◄

(150809)

IMST

IMST GmbH (www.imst.de) is a design house and development center for RF modules, communication systems, chip design, antennas, EDA software and regulatory certification with an accredited test center. IMST supplies both standard products such as RF modules with hardware and software and also complete system and product development capabilities.



PIC Assembler Crash Course 2.1

Driving LED displays using interrupts

Last year we published a three-part crash course in assembler programming for PIC microcontrollers, illustrating the theory and practice of programming these devices close to the metal. In this second series we delve deeper, beginning in this installment with a look at interrupts.

By **Miroslav Cina** miroslav.cina@t-online.de

First we shall take a look at an interesting range of PIC microcontrollers. In the first series of this course we used the Microchip PIC12F675 to illustrate our example projects. This microcontroller is indeed a very capable device, but better is always the enemy of the good: in **Table 1** we list the devices from Microchip's range of 'flash microcontrollers with nanoWatt XLP technology', which offer a wider range of possibilities and make programming much more straightforward.
There is a total of ten members of the family. The smallest chip in the range has just eight pins, of which six can be used for I/O functions, just like the PIC12F675 we previously used. The biggest devices have twenty pins and eighteen I/O lines. As Table 1 shows, devices in the range offer from 2 to 8 kwords of program memory, 256 bytes of EEPROM and from 128 to 1024 bytes of RAM. Despite all this, a PIC12F1840 costs just 15 cents more than a PIC12F675: money well spent.

Memory organization

Before getting down to some programming it is worth familiarizing ourselves with the memory structure of these 'enhanced mid-range PICs'. Because a larger amount of RAM data memory is available than on the PIC12F675, it is organized rather differently: it is split into 32 banks each of 128 bytes. Now at first sight this seems odd, as an address space of 32 x 128 bytes comes to 4 kbytes, which is more than the 1 kbyte offered by even the largest device in the family, let alone the 128 bytes offered by the smallest. We can understand what is going on by looking at how the memory banks are organized.
Each memory bank has 128 addressable locations, with hexadecimal addresses from 00h to 7Fh. The bank size is a consequence of the fact that instructions normally have a seven-bit field available for addressing RAM.
Each memory bank is also divided into various areas. The first 12 bytes are reserved for the so-called 'core registers', occupying addresses 00h to 0Bh. Core registers can fundamentally affect the execution of programs, as they include essential functions such as the program counter and the status register. The next 20 bytes are registers dedicated to special functions. These include access to I/O ports and other features of the microcontroller, such as counters, interrupt configuration and so on. This block of registers occupies addresses from 0Ch to 1Fh, and different special function registers appear in different banks.
After address 1Fh comes a RAM area that can be used freely by the programmer, from address 20h to 7Fh. However, just in case that was too simple, the area is not homogeneous: the area is further divided into two parts.

- **General purpose RAM**: this freely-usable memory area runs from address 20h to 6Fh, eighty bytes in total
- **Common RAM**: the remainder of the RAM address space, from address 70h to 7Fh, can also be used for any purpose, but there is a special feature: the same sixteen bytes are common to all banks. So, if you wish to define a variable that needs to be accessed frequently independent of the main bank selection, it is a good idea to allocate space for it in this area.

So, in each memory bank there is at most 80 bytes of 'genuine' RAM. How much is actually accessible varies from device to device. So, for example, the PIC12F1822, with its 128 bytes total RAM, has the full complement of 80 bytes available in bank 0, but only 32 bytes in bank 1. Why only 32 bytes? The reason is that we also need to count the 16 bytes of common RAM to arrive at the total of 128 bytes. Banks 2 to 31 do not

Table 1. PIC12F1822 / 1840 / PIC16F182X / 1847 Family Types

MCU	Program Memory	EEPROM	RAM	I/Os	ADCs	Comparators	Timers 8/16 bit	MSSP I²C / SPI
PIC12F1822	2 K	256	128	6	4	1	2/1	1
PIC12F1840	4 K	256	256	6	4	1	2/1	1
PIC16F1823	2 K	256	128	12	8	2	2/1	1
PIC16F1824	4 K	256	256	12	8	2	4/1	1
PIC16F1825	8 K	256	1024	12	8	2	4/1	1
PIC16F1826	2 K	256	256	16	12	2	2/1	1
PIC16F1827	4 K	256	384	16	12	2	4/1	2
PIC16F1828	4 K	256	256	18	12	2	4/1	1
PIC16F1829	8 K	256	1024	18	12	2	4/1	2
PIC16F1847	8 K	256	1024	16	12	2	4/1	2

contain any further RAM. If an address in the general-purpose RAM area that does not correspond to actual memory is read from (in bank 1 from address 40h to 6Fh or in banks 2 to 31 from address 20h to 6Fh), the result is always zero.

Figure 1 compares the layout of bank 0 between the PIC12F1822 and the PIC16F1847 microcontrollers. As you can see, the two family members are practically identical in this regard. What differences there are reflect the special features of the devices, such as the PORTB register at address 0Dh, which is missing from the PIC12F1822 as it does not have a PORTB. The device datasheets [1] give more details on the layout of the memory banks of each microcontroller type.

Direct addressing using the BSR

There are two ways by which memory can be accessed. The first, direct addressing, uses the BSR (bank select register); we will look at the second later in this course.
The BSR makes addressing RAM reasonably straightforward. In order to access a given memory bank, we simply write the bank number into the BSR. For example, to access bank 0 using the BSR we proceed as follows.

```
v_tmp EQU   H'30'
;
   movlw H'00'
   movwf BSR   ;select BANK 0
   movlw H'28' ;28h
   movwf v_tmp
```

In this example we write the value 28h to address 30h, having first set the BSR to point to bank 0 by writing 00h to it. If we had written 02h to the BSR instead, the value 28h would have been written to address 130h, which is in bank 2. In that case the code would look as follows.

```
v_tmp EQU   H'130'
;
   movlw H'02'
   movwf BSR   ;select BANK 2
   movlw H'28' ;28h
   movwf v_tmp
```

These examples show that just declaring the address of the variable in memory is not enough to ensure the correct memory location is accessed. As a further example, try declaring the variable `v_tmp` to be at address 230h as shown in **Listing 1**: in step 1 we write a value to `v_tmp` immediately after the declaration, but the actual memory location accessed is determined by just the bottom seven bits of the declared address, along with the contents of the BSR. In step 2, where we have selected

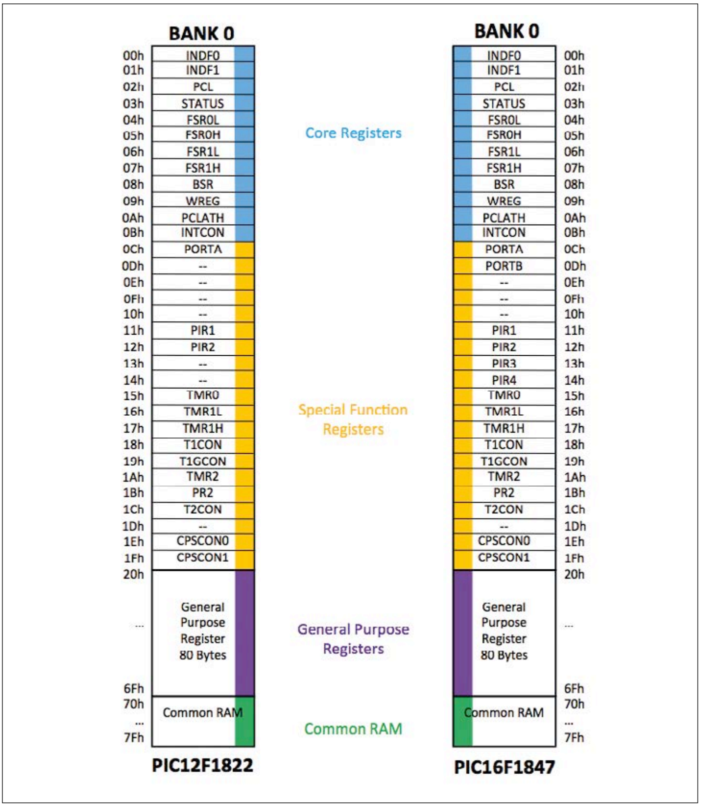


Figure 1. Differences between bank 0 in the PIC12F1822 and PIC16F1847 microcontrollers.

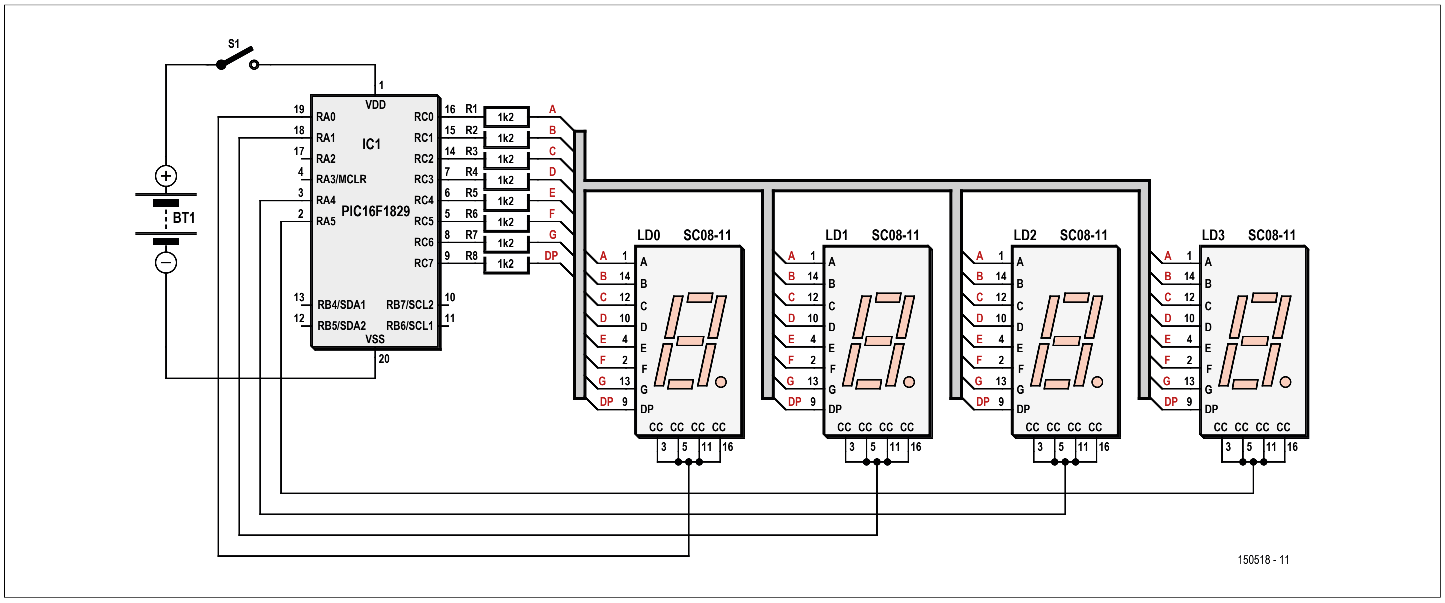


Figure 2. The complete example circuit consists of just the microcontroller, eight resistors and four LED displays.

bank 1, the value 28h will still not be written to address 230h but to B0h, which is made up of 30h from the declared address plus 80h, the start address of bank 1. The correct address of 230h is only used if we write the correct bank number, in this case 4, to the BSR as shown in step 3. Of course we have assumed that the chosen microcontroller device does indeed have sufficient total RAM that there is memory at this location: in this example a minimum of 512 bytes is required.

Controlling an LED display with interrupts

Having dealt with the basics of memory addressing we can now get down to the concrete example of writing some code to use interrupts to drive a four-digit LED display. The example will demonstrate how programming this newer family of devices is rather easier than programming less powerful microcontrollers such as the PIC12F675.

The hardware

In order to drive four seven-segment LED displays statically we need, according to my arithmetic, 4 x 8 = 32 control lines (not forgetting to take into account the decimal point included in each digit). Now, wiring up 32 signals for such a simple task seems too much like hard work, and so such displays are not usually driven statically, but rather using a multiplexed arrangement. The standard approach to multiplexing displays is to drive each segment of all four displays together, and then turn on each digit very quickly in succession for a brief period. If the time slices are short enough then the multiplex frequency will be so high that the eye is tricked into seeing a steady display. Thus we only need 8 + 4 = 12 lines to control a multiplexed display, which makes the hardware much more convenient. On the other hand, we need to implement the multiplexing in firmware in the microcontroller.

Figure 2 shows a complete fully-functional example circuit using a multiplexed four-digit LED display. The LEDs shown draw some 10 mA of current, and so it is practical to power the circuit from a battery (for example, 4.5 V from three AA or AAA cells) via switch S1. The other components are the PIC16F1829 microcontroller, which has more than enough memory and, with 20 pins in total, enough I/Os to drive the display; the eight series current-limiting resistors; and of course the four seven-segment displays. **Figure 3** shows an assembled prototype of this circuit.

It is important to note that each of the eight segment outputs of the microcontroller (RC0 to RC7) is connected via a single series resistor to the same segment on all four display digits. So, for example, the microcontroller drives the 'A' segment on all four displays through resistor R1 using output RC0. The displays used here have a common cathode, and each cathode is driven from one of the microcontroller's outputs from RB4 to RB7. Since the total current of all the segments of one digit flows through the common cathode, in order to get full brightness from the displays a drive transistor or similar should be added between the output of the microcontroller and the

Listing 1.

```
;Step 1
v_tmp EQU   H'230'   ;declaration
;
;Step 2
   movlw H'01'
   movwf BSR   ;select BANK 1
   movlw H'28' ;28h
   movwf v_tmp
;
;Step 3
   movlw H'04'
   movwf BSR   ;select BANK 4
   movlw H'28' ;28h
   movwf v_tmp
```

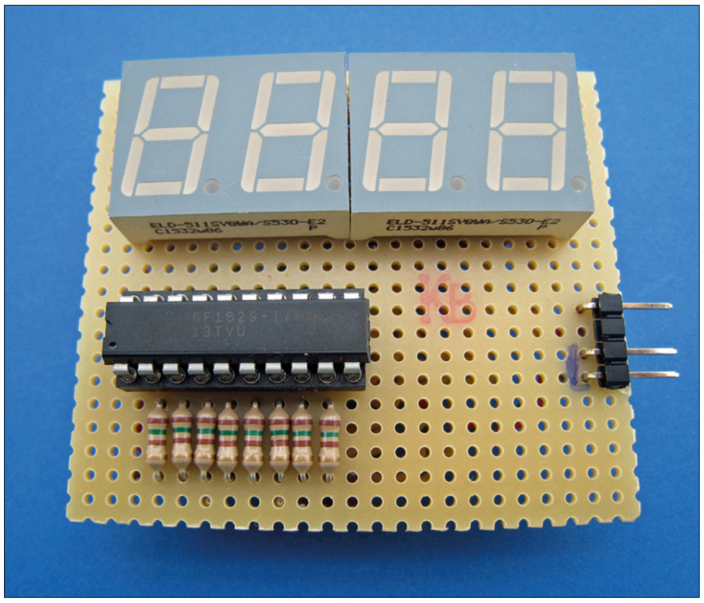


Figure 3. The circuit can easily be assembled on a small piece of prototyping board.

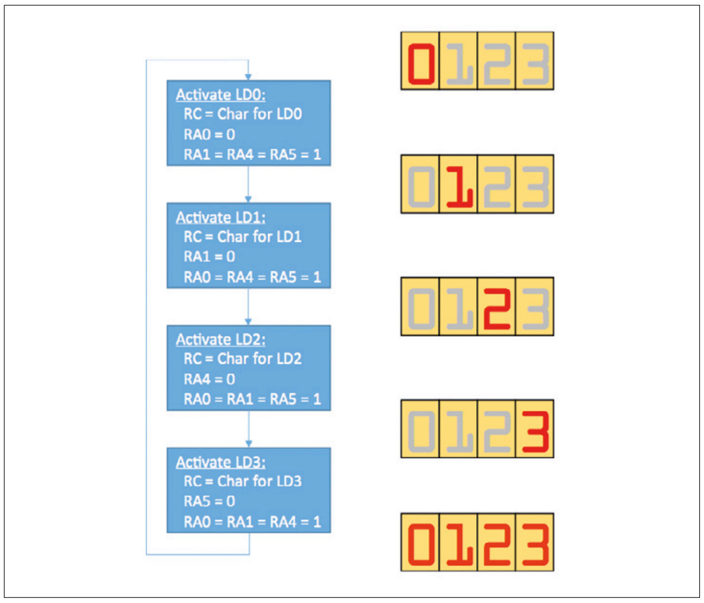


Figure 4. This pseudo-code illustrates 'multiplexing': the digits of the seven-segment LED display are driven in turn.

cathode connection. For testing and for indoor use, however, a lower brightness is adequate. With the values given for the segment resistors the total cathode current is is at most around 20 mA, which is below the maximum current of 25 mA that the microcontroller's outputs are rated to sink.
Multiplexing the display is now pretty straightforward: briefly set each of the cathodes of the four display digits low in turn, and during each period take the anode driver pins of the microcontroller corresponding to the pattern to be displayed on that particular digit high. Of course it is necessary to ensure that during this process only one cathode can be activated at a time, as otherwise unwanted segments will light.

The software

To drive the multiplexed display the software must ensure that first digit LD0 is briefly lit, then LD1, then LD2 and finally LD3; and then the process starts again from the beginning. **Figure 4** shows pseudo-code illustrating the idea.
For a steady display the frequency with which a complete cycle of the digits is completed must be considerably higher than the maximum frequency at which the human eye can perceive flicker; at low frequencies the flashing of the digits is clearly visible and the display is hard to read. Also, for even brightness each digit must be driven for the same period of time.
If the microcontroller has nothing else to do, an infinite loop would be sufficient to do the job described above. However, a working display that has nothing interesting to show doesn't make much sense, and so the microcontroller will also be busy generating or processing data such as the time or sensor readings. These activities take processor time, and if we use the infinite loop approach we somehow have to fit this extra time in between the driving of the individual digits. This might have the knock-on effect of altering the multiplex frequency or the active periods for the four digits relative to one another, neither of which is desirable. A better (and hence widely-used) approach is to use interrupts.

Using interrupts

An interrupt pauses the normal execution of a program and temporarily diverts execution to an interrupt service routine. If we arrange things so that an interrupt is triggered at regular intervals then we can use the service routine to drive the display. Microcontrollers generally include a timer module which is ideal for generating regular interrupts: in fact, the device we are using has several timers, called Timer0, Timer1, Timer2, Timer4 and Timer6. In our example we will use Timer2. Each time the interrupt service routine is called the next display digit in sequence will be activated, and this digit will remain lit until the next time the interrupt is called.

Timer2 can be configured for a wide range of functions. As **Figure 5** shows, the timer module contains three counters connected in series as follows.

- The prescaler is driven from the main oscillator divided by four. The prescaler can be configured so that its output frequency is equal to its input frequency divided by 1, 4, 16 or 64.
- TMRx (TMR2 in the case of Timer2), is incremented on each pulse from the output of the prescaler. On each change the contents of the TMR2 register are compared

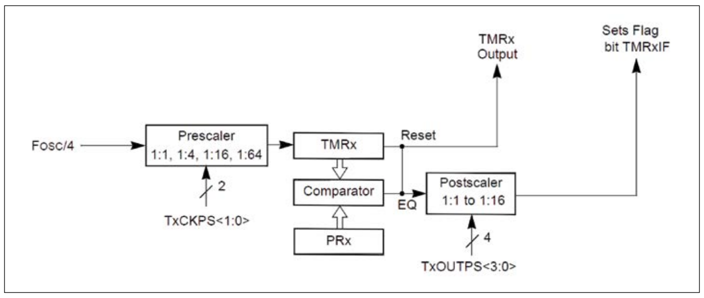


Figure 5. Block diagram of one of the timer modules in the PIC16F1829 microcontroller.

with the value in the PR2 register, and when they agree an output pulse is generated for the postscaler.

- The postscaler is the last counter in the chain. It can be configured for a maximum count of 1 to 16. On each overflow it sets the TMR2IF flag, which, when enabled, triggers an interrupt.

To set things up for driving the display we need to consider four registers. The first is the T2CON register, whose bit layout is shown in **Table 2**. As you can see, the two least significant bits of T2CON specify the prescaler division ratio according to the information given in **Table 3**. The bit TMR2ON enables or disables the timer. The last four bits, T2OUTPS, specify the division ratio of the postscaler according to the values shown in **Table 4**.
The last three registers are most simply explained by looking at the firmware code itself. Initializing the interrupt and Timer2 proceeds according to the steps shown in **Listing 2**: first, in step 1, we set the prescaler division ratio to 64 (T2CKPS<1:0> = 11) and the postscaler division ratio to 1 (T2OUTPS<3:0> = 0000); then TMR2 is cleared, so that it starts counting up from zero; and finally, the value 02h is written to register PR2. An interrupt will now be generated each time TMR2 reaches the value 2. The postscaler has in effect been disabled by setting its division ratio to 1.
The second step is to configure the interrupts. Setting bit TMR2IE in register PIE1 to 1 enables the interrupt from Timer2. Now we simply have to set the global interrupt enable flags, bits 6 and 7 of the INTCON register, to 1.
Finally, step 3 starts Timer2 running.

Now to the interrupt service routine (ISR) in **Listing 3**. The first thing to note from the listing is that the routine does not begin by saving registers such as W or the STATUS register. In part 3 of the first series of this course [2] we looked at how the W and STATUS registers of the PIC12F674 microcon-

Table 2. T2CON: Timer 2 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
–	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>	

Listing 2.

```
;Step 1: timer setup
timer2_init movlw B'00000011'
;T2CON, TMR2, and PR2 are in BANK0 => no bank
;selection required
   movwf T2CON
   clrf  TMR2
   movlw D'002'
   movwf PR2   ;overflow after 2 pulses
;
;Step 2: Interrupt setup
   banksel  PIE1
   clrf  PIE1
   clrf  PIE2
   bsf   PIE1,TMR2IE ;enable TMR2 interrupt
   clrf  BSR   ;select BANK0
   movlw B'11000000'
   movwf INTCON
;
;Step 3: start TIMER2
   bsf   T2CON,D'002'   ;start TIMER2
   return
```

Listing 3.

```
;-------------------------
;Interrupt service routine
;-------------------------
;Step 1
ir_main  clrf  BSR
;bit GIE of INTCON is cleared in HW
;
;Step 2
   movf  v_curr_p,0
   xorlw D'000'
   btfsc STATUS,Z
   call  ishow_digit0   ;LD0 active
;
   movf  v_curr_p,0
   xorlw D'001'
   btfsc STATUS,Z
   call  ishow_digit1   ;LD1 active
;
   movf  v_curr_p,0
   xorlw D'002'
   btfsc STATUS,Z
   call  ishow_digit2   ;LD2 active
;
   movf  v_curr_p,0
   xorlw D'003'
   btfsc STATUS,Z
   call  ishow_digit3   ;LD3 active
;
;Step 3
irs_fin_prep   incf  v_curr_p,1
   movf  v_curr_p,0
   xorlw D'004'
   btfsc STATUS,Z
   clrf  v_curr_p ;start new round
;
;Step 4: finalize the interrupt service routine
irs_fin  clrf  TMR2
   bcf   PIR1,TMR2IF ;clear source of interrupt
   retfie      ;sets bit GIE of INTCON too
```

troller could be saved, but here that is not necessary, as all the microcontrollers in this family perform context switching completely automatically. When an interrupt is triggered the hardware saves the contents of the registers W, STATUS, BSR and FSR (whose function we will describe later) in so-called 'shadow' registers. When the '`retfie`' ('return from interrupt') instruction is executed the previous contents of these registers are automatically restored. Splendid!
In the first step of interrupt processing the BSR is set to zero, so that subsequent register accesses are directed to bank 0.
In step 2 there are four almost identical code fragments, which check which display digit is to be lit by looking at the value stored in the variable `v_curr_p`. Each four-instruction fragment exclusive-ORs the value in the variable with a different value from 0 to 3: for example, if the value in `v_curr_p` is 2 then executing the instruction `xorlw D'002'` will set the Z bit in the STATUS register to 1 (because the exclusive-OR of two equal operands is zero), and therefore the instruction `call ishow_digit2` will be executed. This code is a good example of how to compare two values in assembler.
In step 3 the variable `v_curr_p` is incremented, so that when the next interrupt happens the next display digit will be shown. The code then checks whether `v_curr_p` has reached 4: if so it means that the digit counter has overflowed and it must be reset to zero. The next round can then begin, starting again from the first display digit.
Step 4 completes the processing required in the interrupt routine.

BRW

The PIC12F675 knows a total of 35 instructions. The newer family we are using here offers an enhanced instruction set with a total of 49 instructions. One of the new entries is the `brw` (relative branch with W) instruction, which is useful for saving code space and execution time in an interrupt service routine. The instruction causes a jump but takes no parameters. Its syntax is thus very simple:

```
brw
```

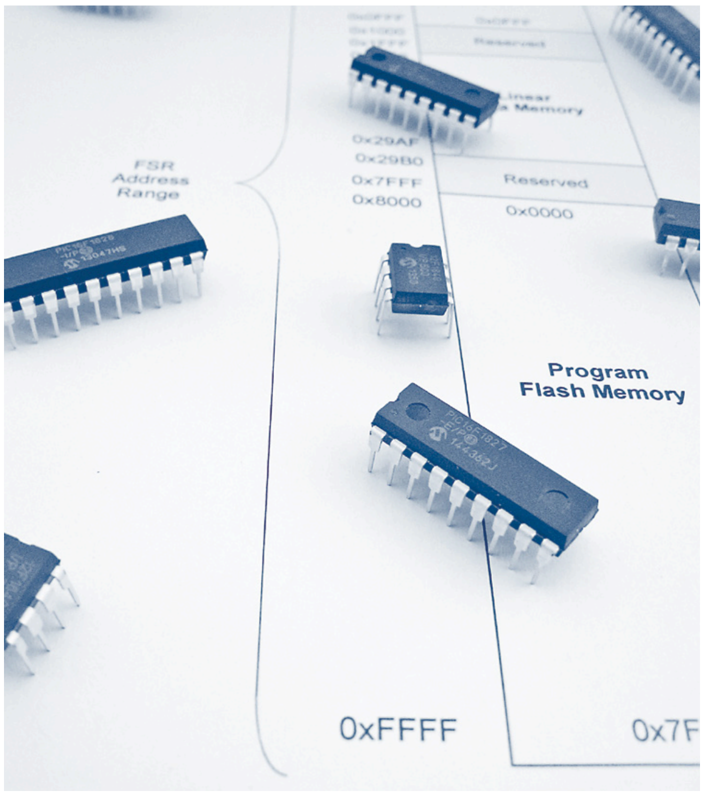
The effect of the instruction is to add the contents of the W register to the PC (program counter). As a result program execution passes to a new address, the displacement of the new address from the instruction itself being given by the value in the W register. The instruction is designed to be used in cases where it is wanted to choose one of several code execution paths depending on the value in a variable. So, for example, one can execute one of four different code fragments according to the contents of W. If W is zero then path 1 is executed; if W is one then path 2 is executed, and so on: see **Listing 4**.

Table 3.
Prescaler configuration

T2CKPS<1:0>	Divisor
00	1
01	4
10	16
11	64

Tabelle 4.
Postscaler configuration

T2OUTPS<3:0>	Divisor
0000	1
0001	2
0010	3
...	...
1111	16



In the listing the W register is first set to the value 02h and then the `brw` instruction is executed. When the value in W is added to the PC the next instruction to be executed will not be `goto path1` but rather `goto path3`. Execution will now branch to the label `path3`.
We can now use the `brw` instruction to simplify the code of our interrupt service routine in an elegant way to branch to one of the four `ishow_digitx` subroutines depending on the value of `v_curr_p`. In **Listing 5** you can see that there is no change needed to steps 1, 3 and 4; step 2, however, has been split up. In step 2a we call a subroutine to use our new technique, and in step 2b the `brw` instruction is used to branch to one of the four routines to light the display digits depending on the contents of `v_curr_p`: if `v_curr_p` is zero then `ishow_digit0`

Listing 4.

```
   movlw B'00000010' ;W = 02h
   brw
   goto  path1
   goto  path2
   goto  path3
   goto  path4
;
;other code here...
;
path1 nop
;
path2 nop
;
path3 nop
;
path4 nop
```

is executed, if v_curr_p is one then ishow_digit1 is executed, and so on. Instead of 4 x 4 instructions we now have just seven. The individual subroutines ishow_digit0 to ishow_digit3 are practically identical to one another. The only difference is in which of the outputs RB4 to RB7 is set to logic 0 to enable the appropriate digit of the display. Otherwise the code in the subroutines should be self-explanatory, though we should just mention that the code makes use of a display buffer (the variables v_pos0 to v_pos3) to hold the data to be displayed. These four variables act as the point of interface between the main application code and the display driver in the ISR. The main application code stores the data to be displayed in the buffer and the display driver reads them out and displays them. So that the application does not have to worry about encoding digits into the patterns of segments to appear on the LED displays, we also handle that conversion function. If a variable in the buffer holds the value zero, then '0' will appear on the corresponding digit of the display. In the example code eleven segment patterns are defined, giving the digits from 0 to 9 and '-', the last of which is represented by the value 0Ah.

In order that the display driver code can show the correct patterns it needs its character set defined in some kind of table. In the case of the digit '1', for example, we want to light segments B and C of the display. From the circuit diagram, this means that outputs RC1 and RC2 will need to be taken high, while RC0 and RC3 to RC7 will need to be taken low. Again, we can make use of the BRW function to implement the required table, with a little help from the retlw instruction as described in the next section.

RETLW

The retlw instruction is a variant of the return instruction. The latter serves to end a subroutine and return to the calling code. The retlw (return with literal in W) instruction does the same, but also takes a parameter. This parameter, a constant, is placed in the W register as part of the return operation. The syntax is as follows.

```
   retlw k
```

As **Listing 6** shows, a combination of the brw and retlw instructions gives us a simple implementation of the character set

Listing 5.

```
;--------------------
;ISR with command bwr
;--------------------
;Step 1
ir_main  clrf  BSR
; bit GIE of INTCON is cleared in HW
;
;Step 2a
   movf  v_curr_p,0
   call  ishow_digits
;
;Step 3
irs_fin_prep   incf  v_curr_p,1
   movf  v_curr_p,0
   xorlw D'004'
   btfsc STATUS,Z
   clrf  v_curr_p ;start new round
;
;Step 4: finalize ISR
irs_fin  clrf  TMR2
   bcf   PIR1,TMR2IF ;clear source of interrupt
   retfie      ; set bit GIE of INTCON too
;
;Step 2b
ishow_digits    brw
   goto  ishow_digit0
   goto  ishow_digit1
   goto  ishow_digit2
   goto  ishow_digit3
```

Listing 6.

```
wm_char_set brw
   retlw B'00111111' ;"0"
   retlw B'00000110' ;"1"
   retlw B'01011011' ;"2"
   retlw B'01001111' ;"3"
   retlw B'01100110' ;"4"
   retlw B'01101101' ;"5"
   retlw B'01111101' ;"6"
   retlw B'00100111' ;"7"
   retlw B'01111111' ;"8"
   retlw B'01101111' ;"9"
   retlw B'01000000' ;"-"
```

Listing 7.

```
demo_loop   nop
   call  dr2
   incf  v_pos3,1
   movf  v_pos3,0
   xorlw H'0A'
   btfss STATUS,Z
   goto  demo_loop
;
   clrf  v_pos3
   incf  v_pos2,1
   movf  v_pos2,0
   xorlw H'0A'
   btfss STATUS,Z
   goto  demo_loop
;
   clrf  v_pos2
   incf  v_pos1,1
   movf  v_pos1,0
   xorlw H'0A'
   btfss STATUS,Z
   goto  demo_loop
;
   clrf  v_pos1
   incf  v_pos0,1
   movf  v_pos0,0
   xorlw H'0A'
   btfss STATUS,Z
   goto  demo_loop
;
   clrf  v_pos0
   goto  demo_loop
```

look-up table. When the subroutine `wm_char_set` is called with the W register holding, for example, the value 07h, the `brw` instruction will cause a branch to the line of code where the '7' character is defined. This line then returns a value corresponding to the bit pattern required to display the digit '7'.

Counting from 0000 to 9999

It is now high time we looked at a practical example of how to use all the above theory. We have everything in place to show the contents of the display buffer: to test the code we can write a simple main application in the form of a counter to generate changing data. The count will run from 0000 to 9999 and then restart at 0000. The display driver will be responsible for continuously displaying the current counter state.
Our main application consists of an infinite loop which implements the counter function and writes its state into the display buffer variables. The infinite loop `demo_loop` in **Listing 7** contains the incrementing counter. After each counter state change there is a small delay (implemented by `call dr2`) so that it does not increment too fast to be visible. To explore the behavior of the display driver it is possible to make small changes to the code and observe the effect. For example, you could try increasing the period of Timer2 and see how that affects the display. If you change the postscaler division ratio from 1 to 16 and initialize the register PR2 to FFh rather than 02h, then the sequential pattern of activation of the individual digits will become clear to the eye.

Coming up

This example concludes the first part of our second assembler crash course series. All the code is available for download from the web pages accompanying this article at [3]. In the next installment we will take a look at indirect addressing, including another practical demonstration, as well as some basic information on serial communication using the I^{2}C and SPI buses. If you have any questions on the code in this article, please feel free to contact the author at miroslav.cina@t-online.de. ⏮

(150518)

Web Links

[1] PIC1XF18XX datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/40001413E.pdf

[2] Third part of the first course: http://www.elektormagazine.com/150393

[3] Software download: http://www.elektormagazine.com/150518

Advertisement

USB Add USB to your next project. It's easier than you might think!

DLP-USB1232H: USB 2.0 UART/FIFO

HIGH-SPEED 480Mb/s

- Multipurpose: **7** interfaces
- Royalty-free, robust USB drivers
- No in-depth knowledge of USB required
- Standard 18-pin DIP interface; 0.6x1.26-inch footprint

DLP-IO8-G
8-Channel Data Acquisition

Only $29.95!

- 8 I/Os: Digital I/O
  Analog In
  Temperature
- USB Port Powered
- Single-Byte Commands

DLP-IOR4
4-Channel Relay Cable
DLP-TH1b
Temp/Humidity Cable
DLP-RFID1
HF RFID Reader/Writer
DLP-FPGA
USB-to-Xilinx FPGA Module

DLP Design

www.dlpdesign.com

The Easiest Way to Design Custom
Front Panels & Enclosures

You design it
to your specifications using our FREE CAD software, Front Panel Designer

We machine it
and ship to you a professionally finished product, no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

FRONT PANEL EXPRESS

FrontPanelExpress.com

Analog Delay Lines

Peculiar Parts, the series

By **Neil Gruending** (Canada)

It isn't always necessary to add a propagation delay to a signal but sometimes it's the only way to make a circuit operate properly. A classic example is the trigger circuitry in an analog oscilloscope. The circuit can typically take 60 ns to detect a trigger event so any signal sent to the CRT must be delayed by the same amount to can be displayed correctly.

Delaying a digital signal is easy, but delaying an analog signal is a little more challenging. One way is to digitize it with an analog to digital converter (ADC), add the delay and then go back to analog with a digital to analog converter (DAC). Or you could use one of the hybrid ICs that can sample an analog signal using techniques like charge bucket brigades that don't require digitizing it. But since this is *Peculiar Parts* I thought it would be interesting to look at **analog** delay lines that don't use any ICs at all.

Truly analog delay lines use wave propagation to create the delay. In an ideal case a signal will propagate with a velocity *v* and will arrive at a length *v* at a time *l*/*v* later. For a signal travelling at the speed of light, that means a delay of about 3 ns per meter travelled. Longer delays can be obtained by winding a shielded conductor around an insulating core which can slow down the signal to up to 280 ns per meter depending on the construction. Really longer delay can be obtained by using lumped L and C elements.

Older Tektronix analog oscilloscopes are great examples of the different ways to construct delay lines. The photos printed here show a few instances. The first Tek scope to use a delay line was the 513D which was a large L-C network. The Tek 517 used 15 m of RG-63U cable, and the 519 used a coil of low-loss air dielectric semi-rigid coax. Other oscilloscopes like the famous 7000 series used a special twin-lead cable for its delay line.

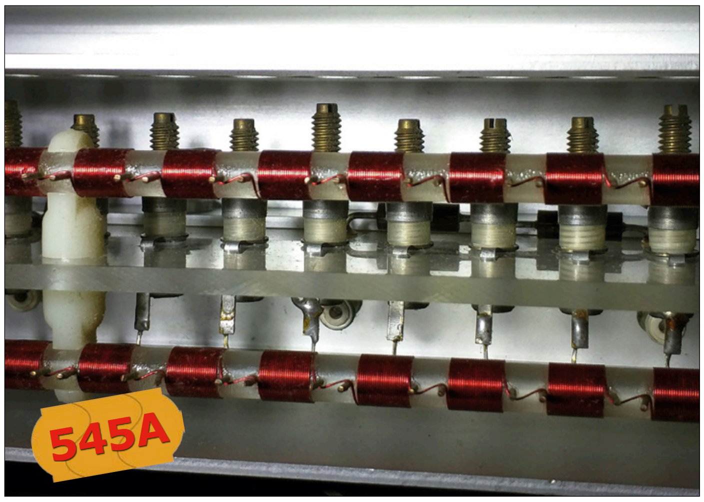
Analog delay lines are also designed to optimize their frequency and pulse response. For example the Tek 513D needed to use trimmer capacitors to adjust the frequency response of each element in the L-C delay line which was especially important for the scope's pulse response. As oscilloscopes got faster, the delay line step performance also needed to be faster which necessitated using coax except that it can't be easily adjusted for optimal performance. It turns out that the solution is to precompensate the signal before it goes through the delay line with external circuitry. Tektronix scopes used a special circuit with loosely coupled inductors which is called T-coil compensation which was one of their secrets behind their performance [1].

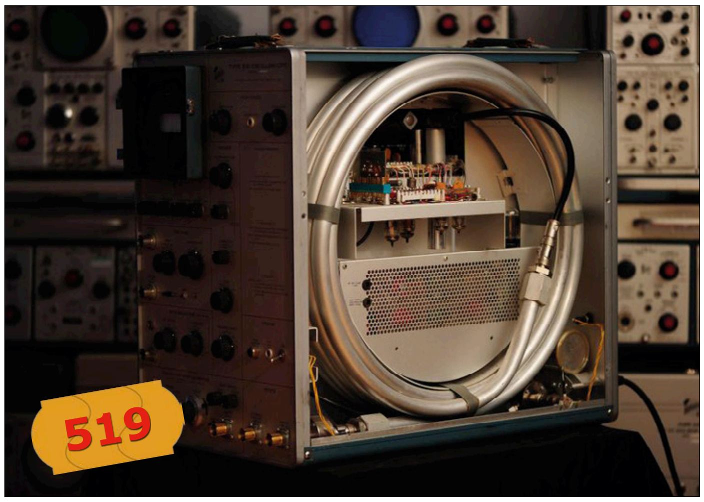
So next time you see a coil of coax delay line used in a piece of equipment, I hope you consider some of the nuances of its simplicity. ◄

Photos courtesy Kurt Harlem, w140.com (150812)

Web Link

[1] *The Art and Science of Analog Circuit Design*, edited by Jim Williams



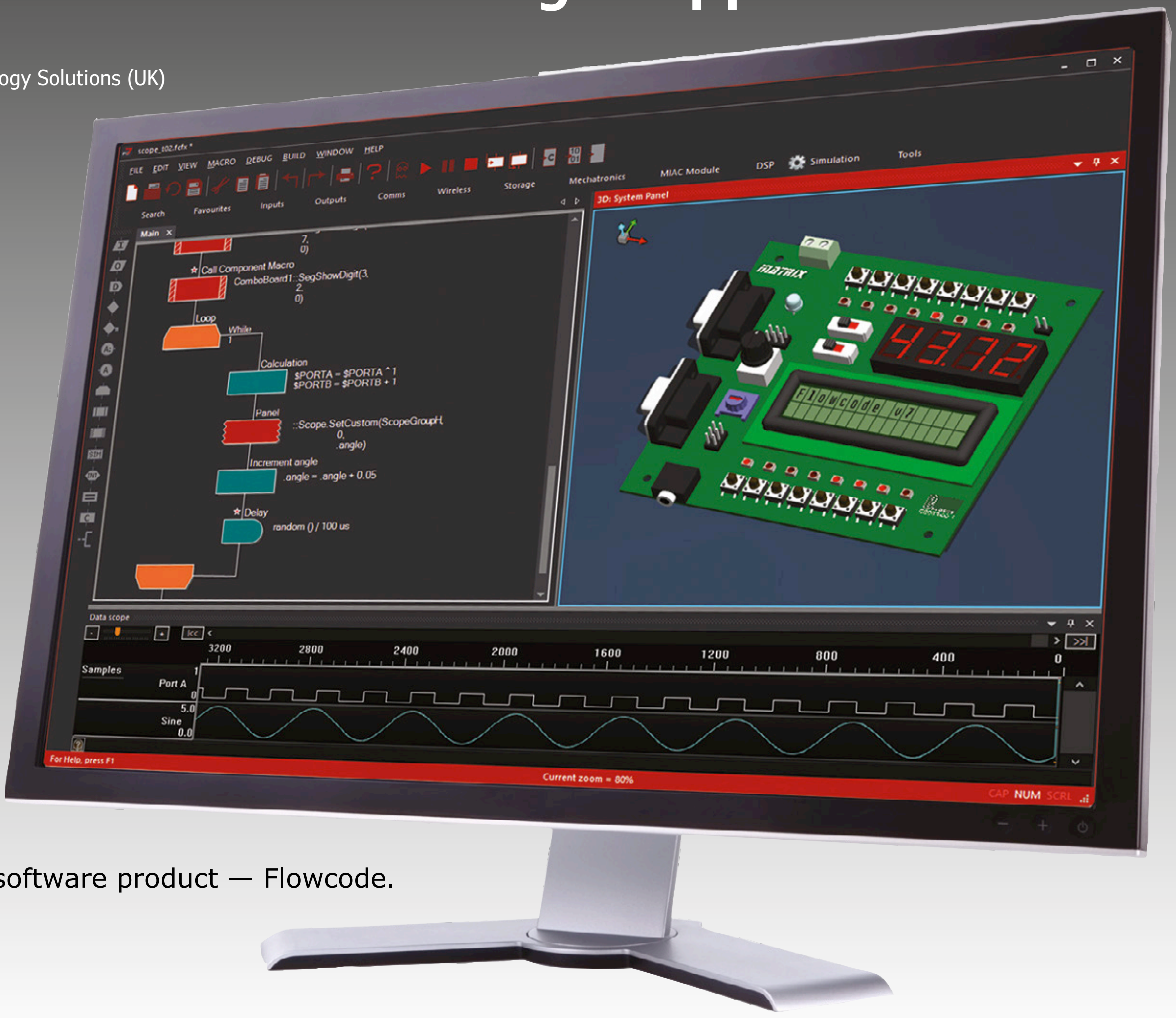


Please contribute your Peculiar Parts article, email neil@gruending.net

FLOWCODE7
Smarter programming

The software tool to make things happen

By **Liam Walton**, Matrix Technology Solutions (UK)



Whatever sized company you work in, one thing remains constant; change. Here at Matrix our main business is the development and supply of microcontroller based development products: at the moment this market is changing at an amazing rate. In this article we review how the market is changing and how this is directing the development of our key software product — Flowcode.

Single-chip microcontrollers have been used in consumer and industrial products for perhaps 40 years now, but in the last few years the manufactured volumes have grown to the point where the features/cost curve is really hard to understand. Simple 8-pin microcontrollers are now available for a few cents. Complex 32-bit microcontrollers with over 40 pins are now available for a few dollars and just cents more than the 8-bit equivalent pin count package. Chip packaging costs are more significant here than silicon costs as the number of transistors per square inch has soared.

Free hardware, free software

Semiconductor companies have recognized that the future innovators are more likely to come from the maker and hobbyist movements and now sell development kits for $15 or less. Here at Matrix we recently bought an ST ARM development kit with integrated color 6 inch LCD screen for just $45. We recently bought a fully specified Bluetooth module for just $3: direct from China. With zero import duty and shrinking domestic market Chinese companies have established slick direct-to-consumer channels with customers in Europe via E-bay. So you might say that system development hardware is now free. (How does the RPi foundation produce and sell a board for $5?)

Similarly, Arduino and RPi have set the bar high for free development software. The Arduino IDE is simple, capable and effective.

How has this affected us? This summer we're launching version 7 of Flowcode — a graphical programming environment for microcontroller based systems (**Figure 1**). Flowcode is selling more copies than ever before: Arduino and the Maker movement have attracted many new customers to the market, but they struggle

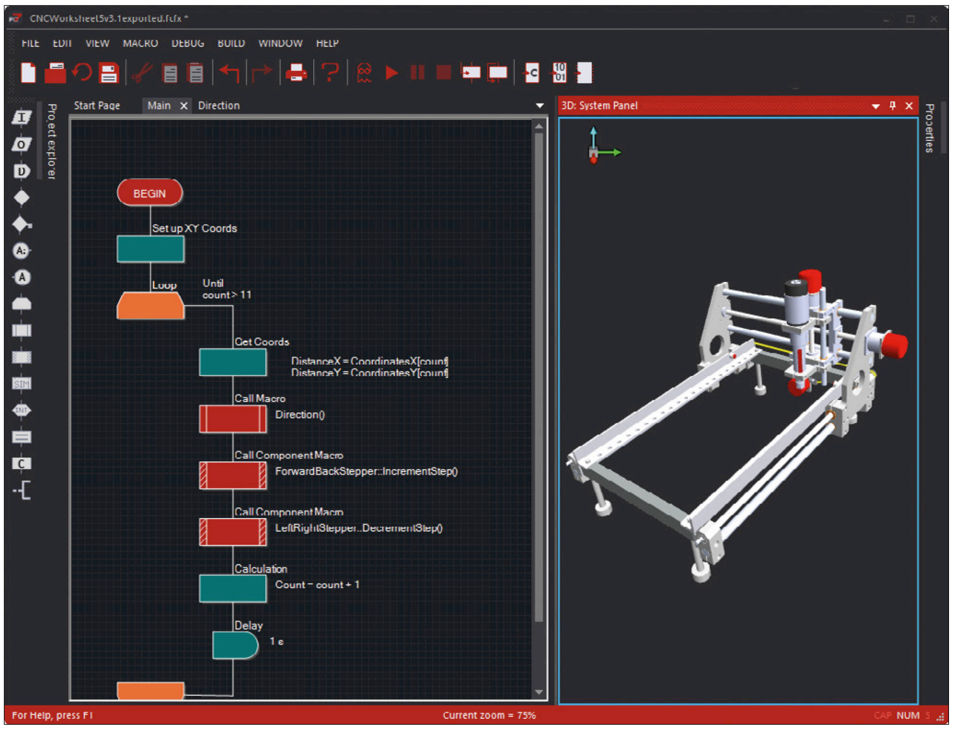


Figure 1. Welcome to Flowcode 7!

with developing in C and are attracted to Flowcode's simplicity. Flowcode is compatible with many microcontrollers: new in V7 is support for Microchip's powerful PIC32 series of 32-bit processors: incredible processing power for little extra cost, and all programs written in Flowcode will directly transfer to this new platform and can take advantage of 32-bit processor power for mathematical and processor power hungry functions like speech generation. Flowcode programs will seamlessly transfer between microcontroller types allowing users to take advantage of many hardware platforms.

Of course we still sell our own hardware: it is more expensive than that from the big semiconductor companies as our volumes are smaller and we need to make a profit: but we have included time saving debug and instrumentation circuitry which we call 'Ghost'. Ghost includes a data recorder, oscilloscope, logic analyzer, packet decoder, in-circuit and in-system debug. Whilst development hardware and software are free, time is not. Flowcode and Matrix hardware saves time.

Not electronics, not mechanics: mechatronics and robotics

You might not have noticed it but robotics is evolving from car manufacturers' factories and starting to have quite an impact on our lives. Number plate recognition and car park barriers, automatic passport recognition access control, and of course 3D printers to name but a few. Robotics might not (yet) have reached i-robot like humanoid functionality but as 3D printers evolve to solve the mechanical problems we will find them an increasing part of our lives.

Perhaps mirroring this trend in industry, in engineering education the big trend of the last few years has been the merger of the two major engineering disciplines: Electrical and Mechanical engineering. Certainly this is true of France's well respected *Lycees Technique* school system, and the gold standard in engineering education here in the UK: the *BTEC National* in engineering. This follows industry's demand for more rounded engineers who now need skills across engineering disciplines. Following on from this has been the acknowledgement that engineers are more likely to have a tablet computer in their hands than a soldering iron or a spanner: so all engineers are now taught programming skills from an early stage — particularly the programming of microcontrollers (as opposed to programming PCs) as these little devices will be at the heart of every engineering system.

How has this affected us? Version 6 of Flowcode included simulation of both 2-dimensional and 3-dimensional systems. This confused many customers. Why would you need to simulate a microcontroller's behavior in 3D? The answer here is that engineers are not designing a microcontroller: they are designing a system with a microcontroller in it. Now if we can get development engineers to simulate the mechanics and the electron-

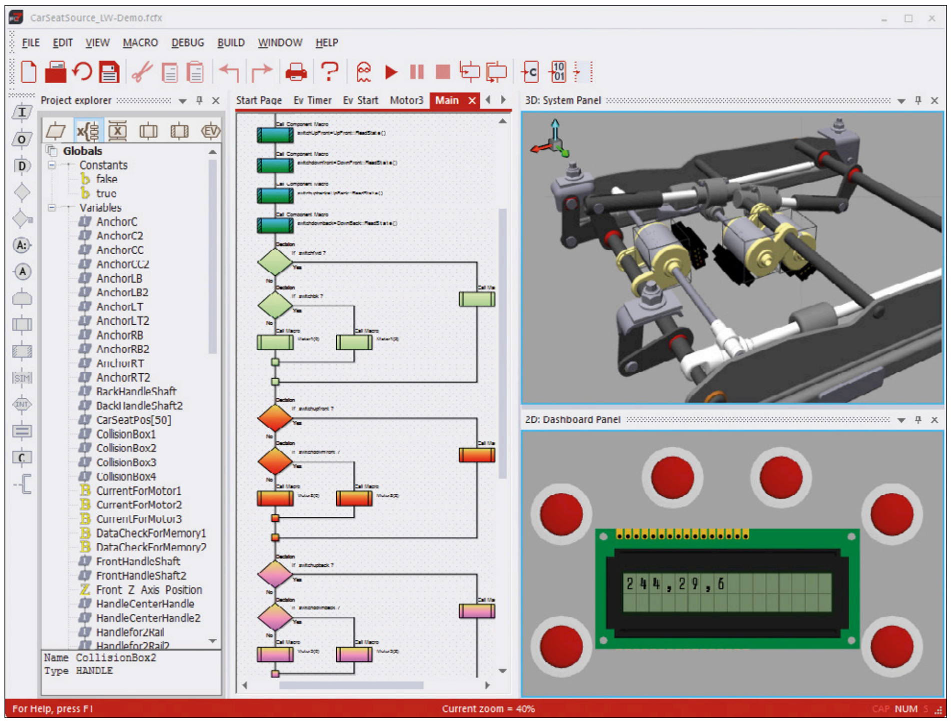


Figure 2. The "Flowcode" approach to designing the electronics in a car seat.

ics of a system together then it will make the design process easier and shorten the design cycle — saving time and money. Similarly if we can provide students with an environment where they can simulate electromechanical, or chemical, or production engineering systems together then it provides a richer learning experience. To enhance this further Flowcode 7 has new, improved 3D simulation capabilities alongside the ability for users to integrate with 3rd party CAD packages such as Solidworks and DesignSpark Mechanical. **Figure 2** shows how automotive engineers can now use applications and examples built into Flowcode to characterize the electronic elements of a car seat and learn how microcontrollers work.

ADAS and vehicle technology

The biggest thing to hit us in the next few years will be self-driving vehicles using Advanced Driver Assistance Systems (ADAS). This will be the biggest robotics explosion ever to hit us. By 2025 (at the latest) analysts believe that we will have driverless vehicles on our roads. In fact car manufacturers — along with Google and others — have had working prototypes on the road for a few years now. The interesting thing about this for us is just how many customers we have who already use Flowcode for developing technology for vehicles: not for the main vehicle brand ECU systems, but for all sorts of affiliated products: electric vehicle displays, campervan habitation systems, remote relays, test gear etc. Crucially, these systems consist of networked microcontrollers — using CAN bus as the principle communication standard. *How has this affected us?* The key point here comes back to developing systems rather than just microcontroller circuits: in this case, systems with multiple microcontrollers. Flowcode 7 has loads of features in it for developing systems: it is possible to simulate more than one microcontroller at a time — in fact you can have up to 10 instances of Flowcode on a PC or network of PCs all communicating with each other to simulate the system. System protocols can be based on spreadsheets: so you can design one program, have many microcontrollers, but the data for each microcontroller's behavior can be external. This makes design of multi-microcontroller systems really easy. Lastly the 'injector' feature of Flowcode 7 makes test and debug really easy — you can set up a simulation of the signals a system would receive and make sure your program responds correctly to each incoming communication transfer.

Conclusion

Here, we have just examined three changes that we think about and how we are reacting to them. A few others worthy of mention are:

- Internet of Things continues to grow at a pace and Flowcode 7 is ready with Bluetooth and Wi-fi/Internet comms systems built in.
- Touchscreen PCs and tablets are widespread — so Flowcode icons have become larger and fat-finger friendly.
- Screen real estate continues to be in short supply so Flowcode 7 minimizes system space take up.
- Microchip templates — Flowcode gives users the ability to easily program pre-developed templates of popular microchip development kits including the PICkit low pin count demo and the PICkit 44-pin demo board.

Flowcode 7 is also packed with new features that just make things easier: new fast Microchip XC compilers (in the PIC versions), code profiling (**Figure 3**) visually displays the frequency with which icons are accessed during simulation and debug, Offline help for fast help access, new graphical user interface, and more.

Flowcode 7 is available from the Elektor Store website www.elektor.com/matrix from late June 2016. ⏮

(160050-I)

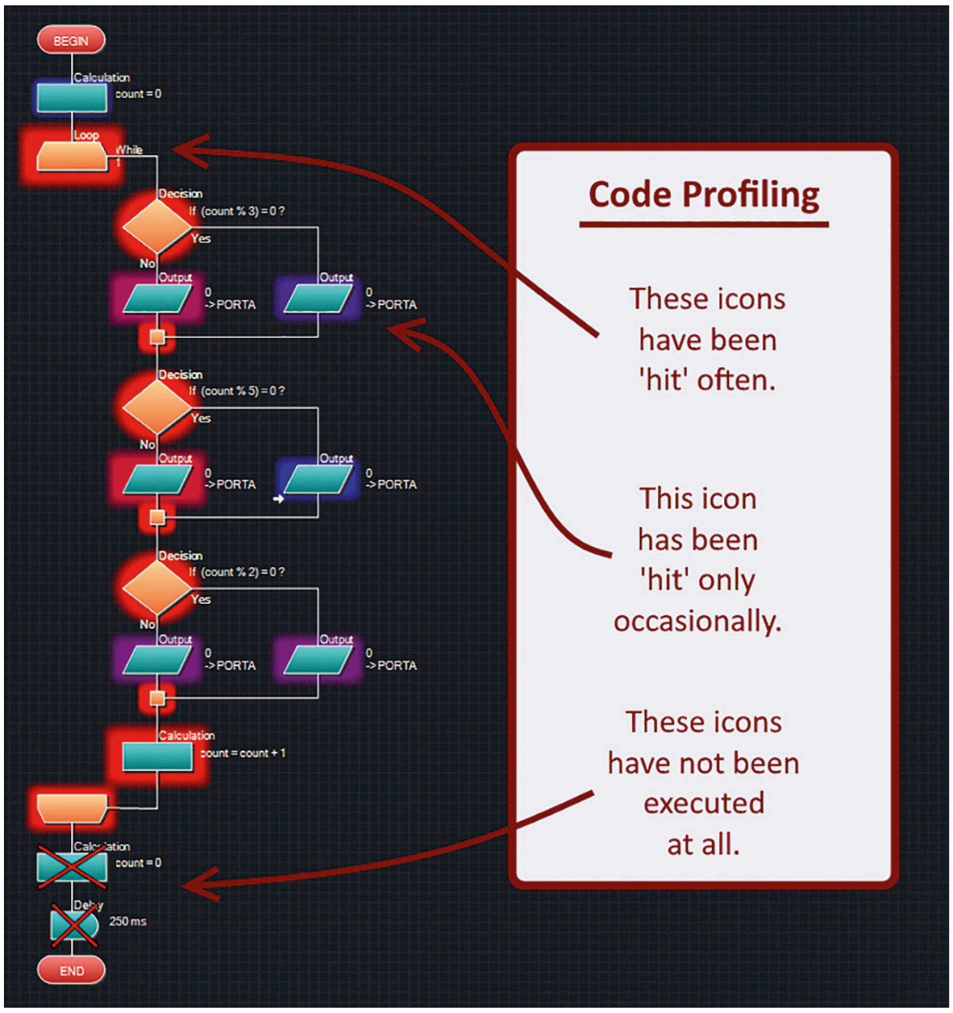


Figure 3. New in Flowcode 7: code profiling.

EAGLE Tips & Tricks (4)

Eagle High-Speed design

By **Neil Gruending** (Canada)

This time we cover net classes and design rules for high-speed designs.

Did you know that Eagle can be used for high-speed designs? Let’s take a look at how to use net classes and design rules to route length-matched differential pairs.

Net classes

A net class in a CAD program like Eagle is a way to group nets together so that you can create design rules and constraints for them as a group. In Eagle, you define net classes using the Edit → Net classes menu which opens the Net classes window like in **Figure 1**.

Every net class has a number (0 to 15) and a Name to make it easy to find. The Width, Drill and Clearance parameters define the track width, via size and trace clearance respectively to use while routing any nets in the net class. In this example, the net class ‘default’ will use a track width of 0.15 mm, a 0.3-mm via and a track to track clearance of 0.15 mm. You can also define the clearance between any net classes by clicking on “>>” to display the clearance matrix for all of the net classes. Net classes are really important for high-speed signals like differential traces because they typically have different width and clearance rules for impedance matching. The clearance matrix is also a good way to make sure that there’s enough clearance between high-speed nets and the rest of the design to avoid cross talk.

You associate a net to a net class by using the Info tool and right clicking anywhere on the net. Then choose Properties to open the Properties window like in **Figure 2** where you can choose the net class for the net.

Design rules

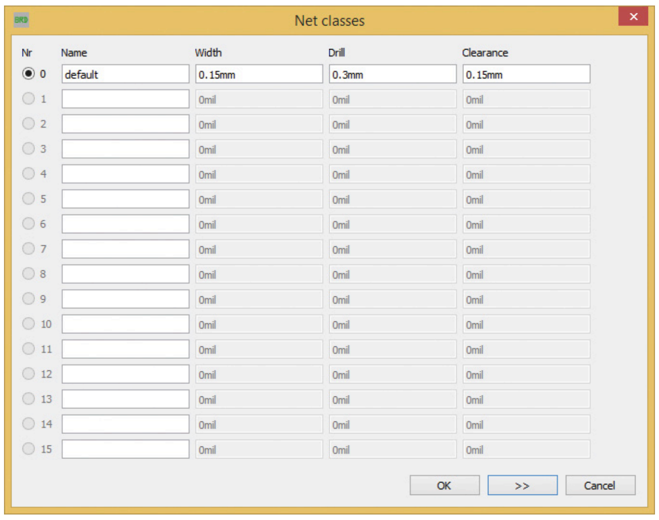
Eagle also has design rules that are applied to a design in addition to the net class rules. The design rules define all of the physical aspects of the board like the layer stackup, clearances and minimum widths. The miscellaneous rules shown in **Figure 3** has a couple of rules that affect differential signal routing. This is where you specify the maximum difference between differential signal track lengths and the gap factor to use when meandering tracks. A meandering track is when you add extra curves to a trace to increase its length. Eagle multiplies the differential trace spacing by the gap factor to calculate the spacing between the added curves.

The Eagle documentation has more information about how to set up the other design rules.

Differential pairs

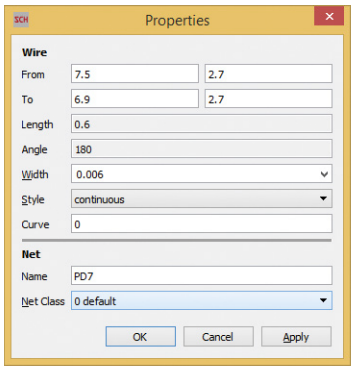
Now let’s experiment with differential pair routing. A differential pair in Eagle is any two nets that share the same base name followed by _P and _N like USB_P and USB_N. Normally the positive net would have the _P suffix and the negative net would have the _N suffix. Now when you click on a differential pair to route it, Eagle will route both of the signals together like in **Figure 4** using the associated net class rules for wire and via size.

But how does Eagle know where to start running the differential traces in parallel? How about when the device package pins air further apart than the desired trace to trace distance? Eagle deals with this by using the point where you click on the


Net classes

Nr	Name	Width	Drill	Clearance
● 0	default	0.15mm	0.3mm	0.15mm
○ 1		0mil	0mil	0mil
○ 2		0mil	0mil	0mil
○ 3		0mil	0mil	0mil
○ 4		0mil	0mil	0mil
○ 5		0mil	0mil	0mil
○ 6		0mil	0mil	0mil
○ 7		0mil	0mil	0mil
○ 8		0mil	0mil	0mil
○ 9		0mil	0mil	0mil
○ 10		0mil	0mil	0mil
○ 11		0mil	0mil	0mil
○ 12		0mil	0mil	0mil
○ 13		0mil	0mil	0mil
○ 14		0mil	0mil	0mil
○ 15		0mil	0mil	0mil

Figure 1. Net classes window.


Properties

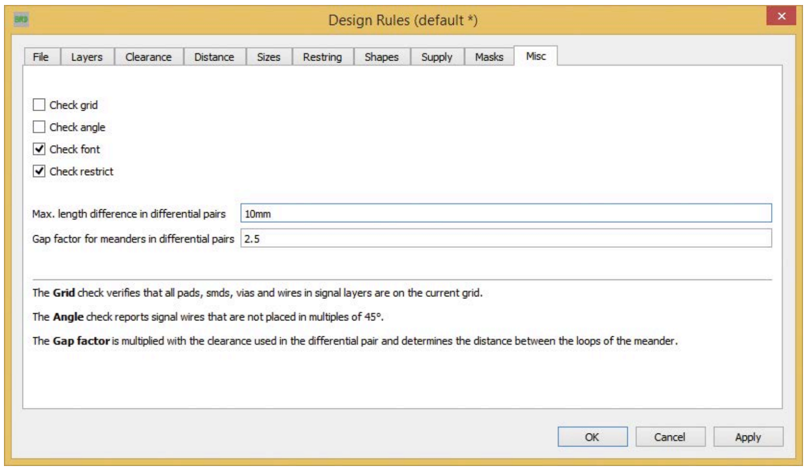
Wire

From	7.5	2.7
To	6.9	2.7
Length	0.6	
Angle	180	
Width	0.006	
Style	continuous	
Curve	0	

Net

Name	PD7
Net Class	0 default

Figure 2. Net properties window.



- [ ] Check grid
- [ ] Check angle
- [x] Check font
- [x] Check restrict

Max. length difference in differential pairs: 10mm
Gap factor for meanders in differential pairs: 2.5

The **Grid** check verifies that all pads, smds, vias and wires in signal layers are on the current grid.
The **Angle** check reports signal wires that are not placed in multiples of 45°.
The **Gap factor** is multiplied with the clearance used in the differential pair and determines the distance between the loops of the meander.

Figure 3. Design Rules window.

connection airwire as the starting point for routing the traces in parallel. Eagle will also draw traces from the closest pads to the starting point using the current wire bend style.

Sometimes the generated traces can cross over each other or violate other design rules so it's important to choose your starting point carefully. It's also a good idea to run a design rule check (DRC) afterwards as well just to make sure everything is ok.

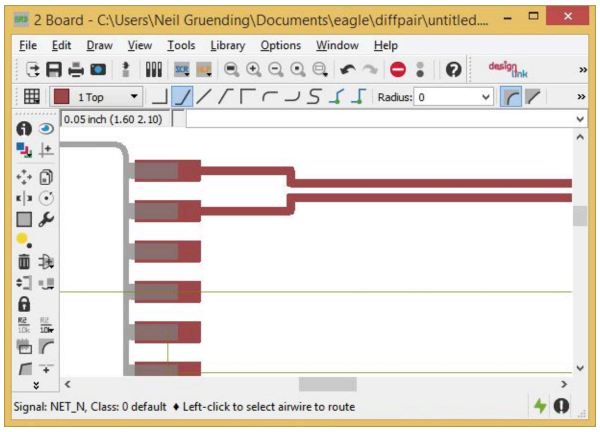
Another critical requirement for routing differential traces is to try and maintain the same lengths for both nets. This is where the Meander tool is useful. Once the traces are routed, you can use the meander tool to select individual traces to extend by inserting a meandering trace like in **Figure 5**.

The Meander tool works by clicking on the trace you want to extend and then it will increase the size of the added meandering trace as you move the mouse away from the trace. As you adjust the meander size a very helpful indicator will pop up to show you the trace lengths and the percentage difference between the differential pair lengths. It's also possible to use the Meander tool as many times as necessary to get the desired length.

Another interesting feature of the Meander tool is that you can use it to control the overall length of a trace. After you select the Meander tool you can enter the trace length you want using the command line and then the tool will meander traces as needed to achieve the correct length. If the trace is part of a differential pair then Eagle will meander both traces together. Otherwise only the one trace will be adjusted.

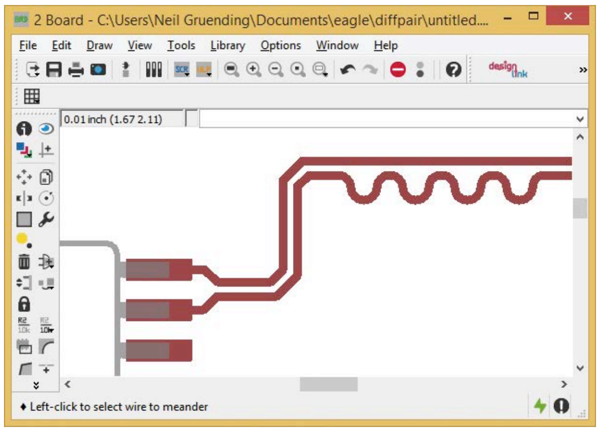
Differential traces in Eagle require a little work to set up but the Meander tool makes it easy to use them in your next design. Give them a try! ◀

(150535)


2 Board - C:\Users\Neil Gruending\Documents\eagle\diffpair\untitled....
File Edit Draw View Tools Library Options Window Help
1 Top
Radius: 0
0.05 inch (1.60 2.10)

Signal: NET_N, Class: 0 default ♦ Left-click to select airwire to route

Figure 4. Routing a differential trace.


2 Board - C:\Users\Neil Gruending\Documents\eagle\diffpair\untitled....
File Edit Draw View Tools Library Options Window Help
0.01 inch (1.67 2.11)

♦ Left-click to select wire to meander

Figure 5. Meander tool routing.

New Life for an NFC Tag (1)

getting right to the guts

By **Patrick Gueulle** (France)

Now that the competition in edition 1/ 2016 is over, it's time to look for new applications for the NFC ST25TA02K tag which was given to you as a key for participating! For example (but this will certainly only be a starting point...), changing the now-defunct URL that had been pre-programmed into it. This is a great opportunity to get to know about reading and writing to these interesting contactless smart cards that are NFC "Type 4" tags.

As near-field communication (NFC) is only one specific (and promising) way of cleverly combining existing contactless technologies (RFID), practically any contactless smart card could theoretically function as an NFC tag.
But in order to promote as wide interoperability as possible (particularly with smartphones), the NFC Forum has defined standardized "Types" that it is strongly recommended to comply with.

This has not, however, stopped certain semiconductor manufacturers from "doing their own thing", in order to push their own products, sometimes numbering them in a rather fanciful way.

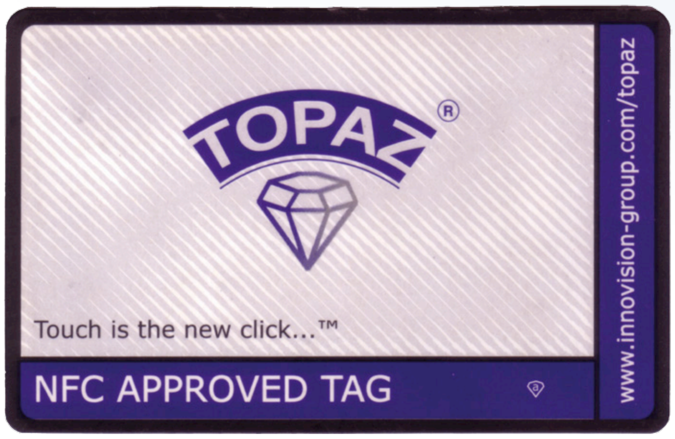


Figure 1. Topaz card from Innovision complying with the NFC Forum "Type 1" specifications.

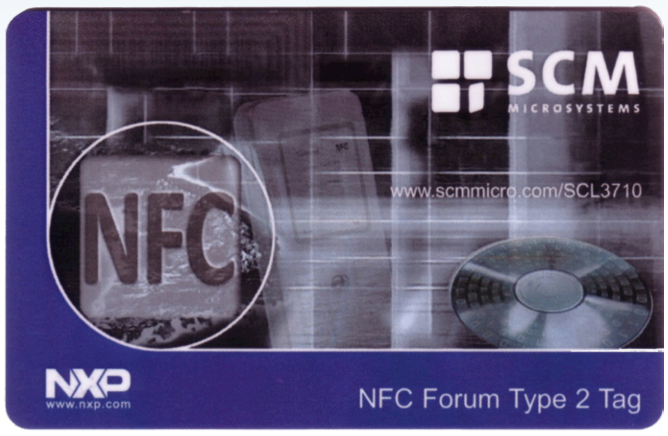


Figure 2. Mifare Ultralight from NXP, easy to configure as an NFC "Type 2" tag.

From *Type 1* to *Type 4*

One of the first “native” NFC tags sold was the Topaz chip from Innovision (then Broadcom), strictly meeting the NFC Forum “Type 1” specifications. However, with only 96 bytes of usable memory, it was only suitable for very simple applications (**Figure 1**). But we’re now up to 512…

NXP in its turn brought along its Mifare Ultralight chip, easy to configure as an NFC Forum “Type 2” tag. But with a capacity of just 64 bytes, it was even more limited (**Figure 2**)! Nowadays, it’s better to go for the Ultralight C and NTAG families — but do let’s acknowledge one decisive advantage for our two pioneers, which is their derisory price!
Let’s pass over “Type 3” (Felica) which is found practically only in Asia, to take a look at the very popular Mifare Classic (or “Standard”) specially formatted as an NFC tag (sometimes nicknamed “Type 7”).

Despite its much greater memory space (already 1024 bytes for the 1 K model), adopting it for an “open” NFC project is not necessarily such a bright idea (**Figure 3**)…
In point of fact, its communication protocol is partly proprietary, and as a result incompatible with a great many smartphones fitted with NFC chipsets from NXP’s competitors!
“Type 4” uses a completely different approach that seems to us infinitely more attractive. Instead of depending on theoretical “low-level” compatibility with a restricted choice of reading platforms, it recognizes genuine ISO7816-4 (T=CL, contactless) commands like any well-bred microprocessor card.

It’s true that the first “Type 4” tag were developed around DESFire or JCOP cards, whose major drawback is however the

Figure 3. Mifare Classic type tag: 1024 bytes of memory, but limited compatibility.

Some useful reminders

“ISO 14443” brings together a series of standards governing near-field radio communication at 13.56 MHz for identification cards, contactless integrated circuit cards, and proximity cards:

Part 1: Physical characteristics
Part 2: Radio-frequency power and signal interface
Part 3: Initialization and anticollision
Part 4: Transmission protocol

The “A” in our tag’s part number ST25T**A**02K, for example, indicates that it complies with the type A protocol in the 14443-4 standard.
As we have already said in the text, the manufacturers are moving away from the standards. Here are a few cards derived from the ISO 14443 standard:

Mifare Classic: Philips (now NXP) proprietary protocol that meets 14443-1,2,3 (type A), but not 14443-4; with or without CRYPTO1 algorithm.
Mifare Ultralight: variant of Mifare Classic, without CRYPTO1.
LEGIC RF: proprietary standard, but with similarities to ISO 14443. They both use the frequency of 13.56 MHz, for example.
FeliCa: proprietary protocol, developed by Sony.

We also mentioned APDU commands. This acronym stands for “**A**pplication **P**rotocol **D**ata **U**nit” – this is the message exchanged between a smart card and a smart card reader. It is standardized and described in the ISO 7816-4 standard.

If we take a look at the application select command, it breaks down as follows:

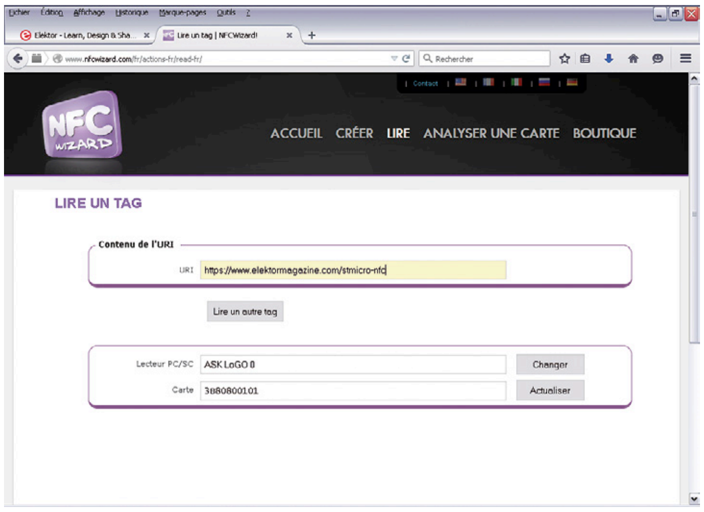
C-APDU of the NDEF Tag Application Select Command

Name	CLA	INS	P1	P2	Lc	Data	Le
–	0x00	0xA4	0x04	0x00	0x07	0xD27600 00850101	0x00
Class byte							
Select instruction code							
P1 field							
P2 field							
Number of bytes of data							
Application ID							
Le field							

Table taken from STMicroelectronics documentation.

The meaning of the fields is as follows:
CLA: instruction class, i.e. the type of command, e.g. *interindustry* or *proprietary*
INS: instruction code (read, write, etc.)
P1 and **P2**: instruction parameters, e.g. position within a file
Lc: number (Nc) of bytes sent by the command
Data: data proper
Le: maximum number (Ne) of bytes expected in the reply, if there is one

Figure 4. The MIFARE DESFire card from NXP is one of the first "Type 4" tags.


Fichier Édition Affichage Historique Marque-pages Outils ?
Elektor - Learn, Design & Sha...
Lire un tag | NFCWizard!
www.nfcwizard.com/fr/actions-fr/read-fr/
Rechercher
Contact
NFC WIZARD
ACCUEIL CRÉER LIRE ANALYSER UNE CARTE BOUTIQUE
LIRE UN TAG
Lire un autre tag
Changer
Actualiser

Contenu de l'URI

URI https://www.elektormagazine.com/stmicro-nfc

Lecteur PC/SC ASK LoGO 0

Carte 3B80800101

Figure 5. Reading the tag contents using a contactless PC/SC reader and online software.

much higher price (**Figure 4**). This is fully justified for "active content" tags that include firmware capable of applying complex (or even cryptographic) processing to the data they carry. And this even makes it possible to give a contactless BasicCard (ZC7.5) genuine NFC functions (see our experimental source code in Elektor January/February 2015 [2]), benefiting too from a huge memory capacity.

While remaining strictly "Type 4" compatible, the ST25TA02K tag from STMicroelectronics does not contains a microprocessor and so remains very cheap (though still has 256 bytes of EEPROM memory). In relative terms, this concept is reminiscent of the CryptoMemory from Atmel, a very original synchronous "contact" smart card (with no microprocessor and so using hard-wired logic), but which accepted asynchronous card commands (APDU) T=0. At the outset, it was a brilliant idea from Gemplus (now Gemalto), who had moreover patented it prior to launching (before our very eyes...) its "GemClub-Memo" card at the CARTES 1998 show in my country.

Let's get talking

We already know that the ST25TA02K tag is supported by a number of applications for (certain) smartphones, which let us do quite a lot of things... but within insurmountable limits. Another approach, even more attractive, consists in using a PC/SC contactless card reader connected to a computer. The make and model don't much matter, as long as it is installed with the right driver (or even with a generic driver if it is also CCID compatible, like the very popular ACR122).

Anything interesting starts with a very small number of very common commands, including in the front line 'Select'. The commands are detailed in the 50-some page documentation (DM00179392.PDF) STMicroelectronics makes available to us on their website [1].

Before anything else (in other words, just after the tag has been presented to the reader), it is necessary to select the application called "NDEF" (NFC Data Exchange Format):

`00A4040007D2760000850101`.

One variant (not really documented) of this command (`00A4040007D27600008501`**`00`**) is supposed to make the tag temporarily compatible with version 1.0 mapping instead of version 2.0, the most commonly used these days.
Once this is done, several files are available to be selected:

- a "Capability Container" (CC) that gives information about the characteristics of the tag,
- a system file specific to STMicroelectronics,
- and above all the (NDEF) file containing the tag's "payload".

Let's skip through those first steps and take a look directly at this one, because this is mainly where we're going to be able to get involved.

Let's fix the URL!

The select command is `00A4000C020001` because the ID for this file is 0001h; it would be E103h for the CC and E101h for the system file).

The tag recognizes two read commands (`ReadBinary = B0`), distinguished from each other by their "ISO class" bytes: `00h` for normal reading and `A2h` for so-called "extended" reading. The difference is that a `00 B0` read command will return an error report (`6282h`) if we attempt to read beyond the end of the "NDEF message", while `A2 B0` lets us read the file right to the end (up to the limit of 255 bytes).
Luckily, the first two bytes of the NDEF file indicate the length of its contents, and we can find this out using the command `00B0000002`.

In the case of our competition tag, the result is `00 24`, telling us there are 36 bytes (24h) to be read afterwards.
So let's read them (with a 2-byte offset, i.e. 02h) using a `00B0000224` command, or else read the file right to the end using the command `A2B00002FE` so as to see the unused (and of course recoverable) memory space.

In either case, we will see appear the NDEF message containing the URL used for taking part in the competition [2], coded as follows:

```
D1 01 20 55 02 65 6C 65 6B 74 6F 72 6D 61 67 61 7A 69
6E 65 2E 63 6F 6D 2F 73 74 6D 69 63 72 6F 2D 6E 66 63
```

From the sixth byte (from 65h to 63h), we find the text *elektormagazine.com/stmicro-nfc* expressed in ASCII. Just before this, byte 02h is a standardized abbreviation of *https://www*. 01h on the other hand means *http://www*. And there are plenty of other variants, certain of which are potentially very useful!

Byte 55h indicates (capital letter U) that what follows is a URL (or more precisely a URI), while 20h specifies its length, including the abbreviation (in fact, 4 less than the length byte 24h previously read, i.e. here 32 bytes).

We don't need to know much more than this to be able to modify the URL "cleanly" — bearing in mind that any anomaly would cause the tag to be rejected when used with a smartphone. We can even try a little experiment: converting it into http://www.elektormagazine.com (always up to date, this one!) by judiciously tweaking… just three bytes. Two `UpdateBinary` (`00 D6`) commands are all it takes: the first (`00D6000601 01`) changes the abbreviation https to http, while the second (`00D6000005001 8D10114`) modifies the two length bytes.

Since this means that the end of the file is shifted to just after .com, the rest of the URL (/stmicro-nfc) will no longer be taken into account. However, it will remain readable via an "extended" `ReadBinary` command (`A2B00002FE`), which covers the whole of the memory space. It's just the same as the way texts deleted on a SIM card can be recovered by forensic science experts!

And it would even be possible to backtrack completely by running the two commands `00D600060102` and `00D600000500024D10120`!

You can check that these operations have worked by reading the tag online, on http://www.nfcwizard.com/fr/actions-fr/read-fr/ using any contactless PC/SC reader (**Figure 5**). Didn't we tell you it's possible to do without Android?
Waiting for better in a second episode… ◄

(150805-I)

Web Links

[1] Command documentation, STMicroelectronics: www.st.com/web/en/resource/technical/document/datasheet/DM00179392.pdf

[2] STMicroelectronics NFC competition (now closed): www.elektormagazine.com/stmicro-nfc

[3] Executable of the program in ZCBasic: www.elektormagazine.com/150805

Advertisement

mouser.com

The Newest Products for Your Newest Designs®

The widest selection of the newest products.

Over 4 Million products from over 600 manufacturers.

MOUSER ELECTRONICS

Authorised distributor of semiconductors and electronic components for design engineers.

electronica FAST FORWARD START-UP AWARD

PUT YOUR INNOVATION IN THE SPOTLIGHT powered by elektor APPLY NOW

Get your business idea off to a flying start on the international marketplace, we are on the lookout for innovative Ideas, Projects and Start-ups worldwide!
The 'electronica fast forward Start-up Award powered by Elektor' initiative brings together, for the first time the worldwide electronics marketplace, innovative technologies and an international media presence to guarantee that your Start-up idea comes to the attention of the movers and shakers of industry.

The call goes out to creative thinkers, developers and Start-ups the world over. You have until July 15, 2016 to send your ideas to us at www.elektor.com/electronica-startup-award.

A Jury made up of Elektor Magazine editors and engineers from Elektor Labs will sift through and evaluate all the entries. Senders of the best entries in the categories of Ideas, Projects and Start-ups will be invited to showcase their ideas and take part in the final at *electronica* – the world's leading trade fair for components, systems and electronic applications held in Munich Germany.
As a participant in one of the award categories of 'Idea', 'Prototype' and 'Start-up' you will get the unique opportunity to make use of the electronica fast forward Start-up Platform powered by Elektor to establish worldwide contacts.

At *electronica* you will benefit from personal business advice, get the opportunity to make important business contacts and attend a wide range of exhibition events. At the exhibition we will take your Start-up idea and provide you with a coherent business plan. We will also allow you to showcase your creative ideas to an international audience of visitors and future customers.
From the finalists attending the trade fair an international jury will select three winners. Should you be one of these winners you will be in receipt of an 'electronica fast forward Start-up Award' which we will be presenting on the November 11, 2016 at *electronica*. In total the three winners will benefit from international PR, consulting and marketing services with an estimated value in excess of 150,000 euros and will benefit from ongoing endorsement via the international Elektor network of over 250,000 subscribers.
The overall winner will be awarded an international Elektor cross-over marketing campaign worth 75,000 euros together with their own stand at *electronica* in 2018. Second place will be awarded an Elektor Media campaign valued at 50,000 euros and third place will benefit from media exposure through Elektor valued at over 25,000 euros.
Curious? For more information on terms and conditions, and the application forms, go to

www.elektor.com/electronica-startup-award.

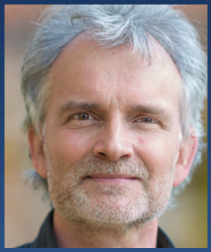
Intelligent IoT Gateway Starter Kit

From ADLINK comes a real-time M2M automation solution featuring the MXE-202i intelligent gateway, based on Intel® IoT Gateway Technology, and Vortex Data Distribution Services (DDS). Motion tracking triggers a motion sensor connected to the MXE-202i, which ingests the data that is then distributed to the robotic devices via Vortex DDS in order to activate the arms. The MXE-200i Series provides an intelligent, robust embedded system supporting wide application development and easy service deployment in industrial automation, while the Vortex intelligent data sharing platform enables easy distribution of data between devices and from device to cloud.
The associated IIoT Gateway Starter Kit serves as a complete connection solution for reduced development time and quick deployment for every application environment. The starter kit contains the MXE-202i intelligent IoT gateway, EdgePro IoT device & sensor management application, light sensor, siren output, Modbus TCP module and accessories. IoT Gateway Starter Kit and EdgePro application benefits also include easy configuration with a user-friendly administrator interface and dashboards.

www.adlinktech.com **(150026-1)**

The ElektorBusiness section in Elektor Magazine accommodates articles, news items and other contributions from companies and institutions active in electronics. Publication is at the discretion of the Editor. Contributions to: newsdesk@elektor.com

Jan Buiting,
your ElektorBusiness Editor

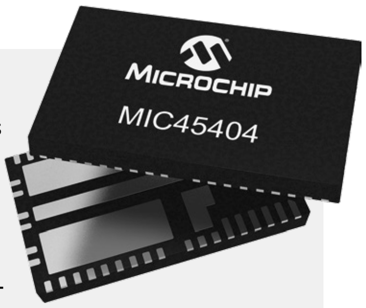


How Price Philip's mailbox was hacked

The archive of a hacking case that made computing legal history in the UK and changed all our lives has been deposited at The National Museum of Computing (TNMOC) by Robert Schifreen, the 'white hat' at the center of a 1980's controversy.
The archive tells the story of Schifreen's two-year legal ordeal following his quite open hack of Prestel, BT's pre-internet public online system, and his demonstration of his ability to access Prince Philip's Prestel mailbox. With no anti-hacking law in existence at the time, the archive gives details of the passage of what turned out to be in effect a test case through three courts ending in Schifreen's acquittal in the House of Lords in 1987.
Included in the archive are Schifreen's 1980's hacking password book, transcripts of his interviews with police, legal correspondence, the Jury Bundle and a substantial number of press cuttings. In presenting the archive, Robert Schifreen explained the context of 1980's hacking to an audience at TNMOC. In 1985, the Internet did not exist, home computing was beginning to take off, Prestel, one of the first public online services had become available, but there was no real awareness of the need for computing security and no law explicitly against computer hacking.
Schifreen, aged 22 at the time, was one of a group of sociable, skilled and inquisitive hackers who, without malicious intent, collected user names and passwords and investigated computer databases not supposedly open to the public. He explained: "Hackers in those days never started until 6 pm because it was so expensive to go online with a dial-up connection before that. But 6 pm was a significant start-time because the Prestel security staff had gone home and weren't there to deal with automated messages telling them that there had been three unsuccessful attempts at a log-on to Prestel. I could read the messages, delete them to cover my tracks before security arrived for work next morning. In effect I was a Prestel System Manager. I even managed to hack Prince Philip's Prestel Mailbox and was quite open about it."

www.tnmoc.org **(160026-4)**

5 amps from 2-mm profile power module

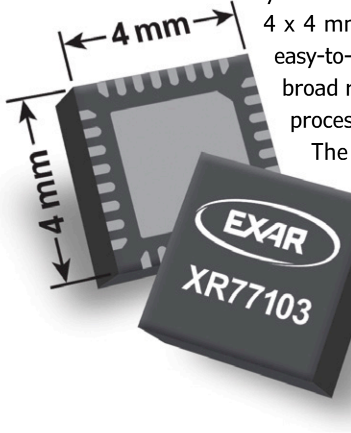


With its slim 10 x 6 x 2 mm package Microchip's MIC45404 integrated switching power module delivers point-of-load power conversion in telecom, industrial and solid-state drive (SSD) applications.
The thermally-enhanced package integrates a controller, MOSFETs, feedback path and a PWM switching regulator. The MIC45404 also integrates one of the thinnest inductors as well as a bootstrap capacitor and high-frequency input capacitor mounted on a thermally-enhanced lead frame. This helps to eliminate unexpected electromagnetic interference (EMI) from external passives in addition to simplifying the board layout.
Development with the DC-to-DC integrated switching power module is supported by the MIC45404YMP-EV evaluation kit.

www.microchip.com/EUMIC45404 **(160026-5)**

3-channel universal PMIC for low-power FPGAs and SoCs

Exar Corporation's XR77103 is a universal PMIC with three integrated synchronous MOSFET power stages. In a tiny, 4 x 4 mm IC the device is said to deliver an easy-to-use power management solution for a broad range of FPGAs, SoCs, DSPs and video processors.



The XR77103 features an I²C interface allowing customers to control output voltage (from 0.8V to 6V), switching frequency (from 300 kHz to 2.2 MHz), power sequencing, and current limit. The XR77103 is supported by a new release of PowerArchitect™ 4 design and configuration software.
The XR77103 operates from a 4.5 V to 14 V input supply and all three outputs are designed for 2-A load currents with peak currents up to 3 A. Since the device employs a current mode control architecture, outputs can be easily paralleled to provide up to a total of 5 A allowing the XR77103 to power a range of low power processors. A selectable Pulse Skipping Mode (PSM) results in improved efficiency at light loads, a key feature in meeting standby energy requirements or extending battery life.
The XR77103ELB, XR77103ELB-A0R5 and XR77103ELB-A1R0 are available in RoHS compliant, green/halogen free, space-saving 4x4 QFN packages.

www.exar.com/products/power-management/universal-pmics **(160026-3)**

Futuristic Fantastic Batteries

must charge in seconds, last months

We've seen a plethora of battery discoveries coming out of universities and research institutions all over the world. Tech companies and car manufacturers are pumping big money into battery development. Still there's nothing "new" in our phones and everyone's waiting for the killer replacement to good old lithium-ion.

Compiled by **Jan Buiting**, Editor-in-Chief

No worries, the wonder battery may be here sooner than you think, and it may be any one of the technologies on parade in this article, or another from a totally unexpected corner (watch MIT and Fraunhofer). Please be cautioned though, this article contains some "extremely forward looking views", some of which were already featured in recent editions of Elektor's weekly e-zine.

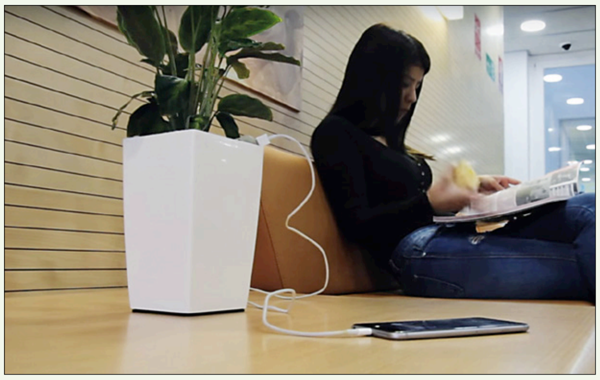


Figure 1. Bioo: electricity from plant photosynthesis.

It's soo eco: Bioo plant charger

The Bioo already exists and can be bought now [1]. It's a plant pot that utilizes photosynthesis to charge your tablet or smartphone. Bioo (Figure 1) is capable of supplying two to three charges per day at 3.5 V / 0.5 A via a USB port cleverly disguised as a small rock. The pot uses organic materials that react with the water and organic matter from the plant's photosynthesis process.

The resulting reaction is claimed to generate enough power to charge gadgets. Thinking Big, Bioo Forest might power cities with 100% green energy and automatically provide the best reason ever to protect plants and trees.

Golden: nanowire batteries

Developed at the University of California, Irvine, nanowire batteries that can withstand plenty of recharging could mean the battery that does not die was just discovered.

A thousand times thinner than a human hair, nanowires pose a great possibility for future batteries. Sadly they always broke down when recharging but now, gold nanowires in a gel electrolyte avoid that. These batteries were recharged over 200 Ktimes in three months and showed no degradation at all. This could be ideal for future EVs, spacecraft and phones that will never need new batteries.

Lightweight: fuel cell for phones & drones

Lighter fuel cells could mean phones only need to charge once a week and drones stay airborne for over an hour (Figure 2). Porous stainless steel with thin-film electrolyte and electrodes of minimal heat capacity are the ingredients. The result is a battery that's more durable and longer lasting than lithium-ion.
Further development for phones, drones and even electric cars is expected soon and with the hub of the research in South Korea we might even see it in the next Samsung Galaxy S8 smartphone.

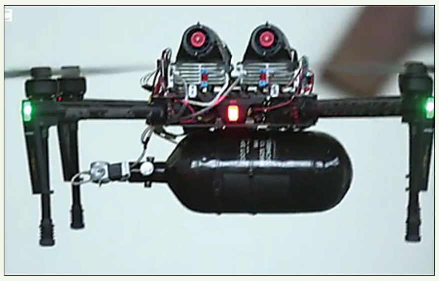


Figure 2. xperimental drone flying on fuel cell power. (BBC)

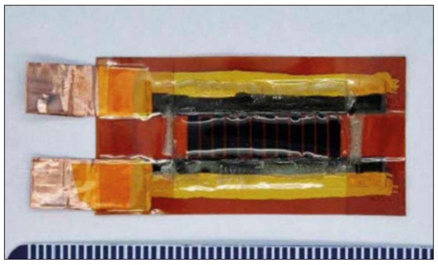


Figure 3. Microsupercapacitor prototype. (Rice University)

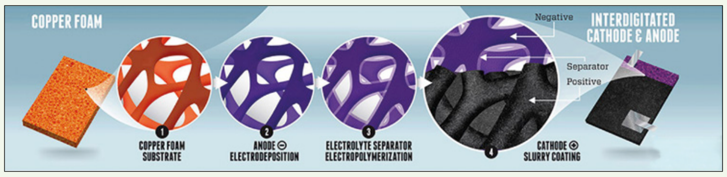
Laser-made: microsupercapacitors

From Rice University comes a breakthrough in microsupercapacitors (Figure 3). Currently they are expensive to make but that could change at the drop of a hat. By using lasers to burn electrode patterns into sheets of plastic, manufacturing costs and effort drop massively.

The result is a battery that can charge 50 times faster than current batteries and discharge even slower than current supercapacitors. They're even tough, able to work after being bent over 10,000 times in testing.

Foamy: foam batteries

With 3D on their minds Prieto was the first company to create a working battery that uses a copper foam substrate (**Figure 5**). Such batteries will not only be safer (thanks to nonflammable electrolyte) but they will also offer longer life and faster charging.
Not forgetting five times higher density, low manufacturing cost and smaller size than current products. That's why Prieto aims to place its batteries into wearables first. The batteries can be upscaled though for use in phones and possibly cars in the future.


COPPER FOAM
1 COPPER FOAM SUBSTRATE
2 ANODE ⊖ ELECTRODEPOSITION
3 ELECTROLYTE SEPARATOR ELECTROPOLYMERIZATION
4 CATHODE ⊕ SLURRY COATING
Negative
Separator
Positive
INTERDIGITATED CATHODE & ANODE
Figure 5. Prieto's copper-foam-substrate battery technology. (Prieto)

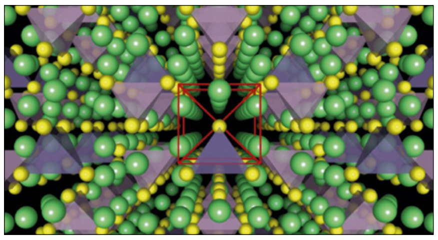


Figure 6. No drip, no filling, no spilling — solid-state batteries last thousands of charge cycles. (MIT)

Rev up: solid-state batteries

Scientists at MIT, working with Samsung, have discovered solid-state batteries that are better than current lithium-ion efforts. These batteries (**Figure 6**) should be safer, last longer and offer more power. Current lithium-ion batteries rely on an electrolyte liquid to transport charged particles between the two electrodes. It's this liquid that can be flammable and which degrades the battery, limiting life.

According to the MIT report these new batteries could be charged for hundreds of thousands of cycles before degrading. They could also provide a 20 to 30% improvement in power density meaning that much more charge for whatever they are powering. And they aren't flammable so they're ideal for electric cars.

Eggnog: nano 'yolk' 3x capacity, 6 mins charge

Also from some great minds at MIT stems a battery that triples the capacity of current offerings yet charges to full in just 6 minutes, that's close to the average concentration span of a teenager. It also does not degrade rapidly over time meaning it should last a long while.
The bonus here is that production is inexpensive and easy to scale. Figure 7 shows the student-friendly "diagram".

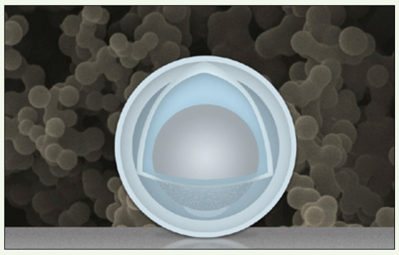
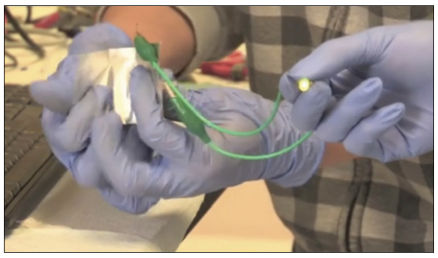


Figure 7. MIT's idea of a nano 'yolk' battery. (MIT)



Faster still: aluminum graphite

Possibly challenged by MIT, scientists at Stanford University create an aluminum graphite battery that could replenish to full in a smartphone in just a minute (**Figure 8**).Their batteries are flexible, long lasting and charge ridiculously fast.
The only issue is they hold about half the power of a current lithium battery, but with charging to full in just a minute that's not too much of a problem, is it?

Figure 8. Aluminum-graphite battery charges in 1 minute — they say. (Stanford University)

Runs on water: Alfa battery lasts 14 days

The Alfa battery with 40 times the capacity of lithium-ion is a breakthrough in aluminum-air technology. You recharge the battery by simply topping it up with water, be it salty or normal.
The new battery (**Figure 9**) is claimed to last 14 days by its creators Fuji Pigment, and will be out later this year. Fuji expect to see these batteries to appear in cars first. Hopefully mobiles will be next in line.
While the alu-air battery with 8.1 kW/kg (claimed) capacity dwarfs Lithium-Ion with its 0.12 – 0.2 kWh/kg it's still lithium-air that comes out on top with 11.4 kWh/kg.

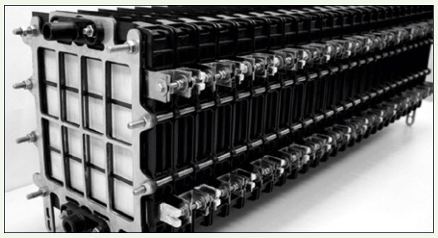


Figure 9. The Alfa battery is based on aluminum-air technology. (Fuji Pigment)

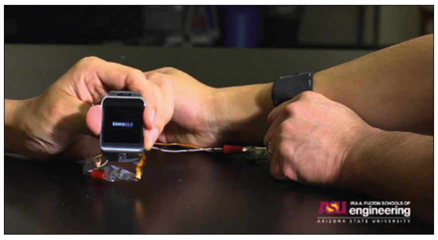


Figure 10. Arizona State University's flexible strap battery is Kirigami-derived.

Ultra-adaptive: flexible battery

A team at Arizona State University have come up with a flexible battery using the ancient Japanese art of Kirigami. A flexible strap battery (**Figure 10**) should allow smartwatches to be smaller and last longer on a charge.

Scientists managed to power a Samsung Gear 2 using a flexible band with the batteries inside. This was stretchy enough to move from the wrist to the biceps, and move with flexing, while still powering the smartwatch. With this technology we can look forward to thinner smartwatches plus clothing with brains and power built in, soon.

Paper-like but tough: foldable battery

Bendable gadgets are possible with the Jenax J.Flex battery (**Figure 11**). This paper-like battery can fold and is water resistant, making it ideal for embedding in clothing or wearables. And beyond that, foldable tablets that you could fit into your pocket just like a phone or a piece of paper!
The battery has already been created and has even been safety tested, including being folded over 200 Ktimes without losing performance.

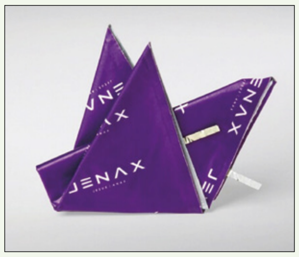
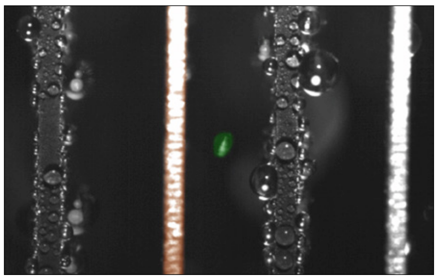


Figure 11. The Jenax foldable battery holds a great promise for wearables.



Phew: power from water dew

Still in the embryonic stage at MIT this futuristic device uses interleaved flat metal plates to produce power from water dew in the air (**Figure 12**). Initial tests have produced small amounts of power, at 15 picowatts, with a promise though to upscale to at least 1 microwatt. Even that is minute and only when time is not issue, and with plenty of dew water, then a charger the size of a coolbox lid might just charge a phone in 12 hours.

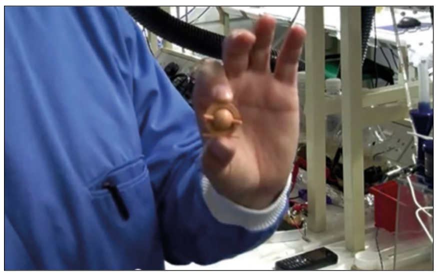
Figure 12. A dew powered phone could be reality one day.

1800 km range: aluminum-air for EVs

As part of test program, a small car (**Figure 13**) managed to go 1800 km (approx. 1100 miles) on a charge put into a high-power aluminum-air battery. The technology uses oxygen to fill its cathode, making it far lighter than liquid-filled Li-ion batteries. Alu-air batteries drain, turning the metal into aluminum hydroxide which can then be recycled to make new batteries. As a “small disadvantage”, the developers say, “users have to swap out batteries every few months”. Luckily the new battery is much lighter and cheaper than current products.

Figure 13. The Alu-air battery has both potential and down sides for electrical vehicles. This little car reportedly went 1800 km on a single charge.

Gotta go: urine-powered batteries



With further funding in the pipeline from The Bill Gates Foundation, Bristol Robotic Laboratory have embarked on a battery that can be powered by urine (**Figure 14**). Reportedly it’s efficient enough to charge a smartphone. Using a Microbial Fuel Cell micro-organisms take the urine, break it down and output electricity — to put it simply. On a scale large enough to charge a smartphone there are several cells into which the urine is passed via tubes. The unit creates electricity and also expels a broken down version of the waste making it safer to dispose of.

Figure 14. With a Microbial fuel cell, micro-organisms can turn urine into electrical power, well drink to that.

Not unlike rhubarb: the organic battery

Depending on the time-to-market of a recent MIT discovery, organic batteries can be added to the list of power sources of the future that are truly sustainable.
A proof-of-concept organic flow battery (**Figure 15**) was found to cost only $27 per kilowatt-hour compared to metal batteries at $700 per kilowatt-hour — nearly a 97 per cent saving. Using quinone molecules comparable to those found in rhubarb, a battery was made that is not only as efficient as metal but that could also be made on a huge scale.

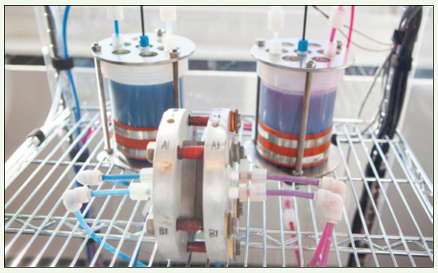


Figure 15. If you haven’t heard of quinone molecules, well in good numbers they make a fine organic battery.

Forget lithium: sodium-ion batteries

Scientists in Japan propose new batteries based on sodium, one of the most common materials on the planet rather than rare lithium — and they’ll be up to seven times more efficient than conventional batteries.
Research into Na-ion batteries (**Figure 16** — artist’s impression) has been going on since the eighties in an attempt to find a cheaper alternative to lithium. Salt is the sixth most common element on the planet and not subject to “supply issues” as with lithium. Some e-prophets proclaim lithium will soon become too rare and expensive for the glut of battery-powered to appear on the roads. Commercializing Na-ion batteries is expected to kick off for smartphones, cars and other devices in the next 5 to 10 years.

Figure 16. Given time the sodium-ion battery might take over from lithium-Ion. (artist’s impression)

Conclusion

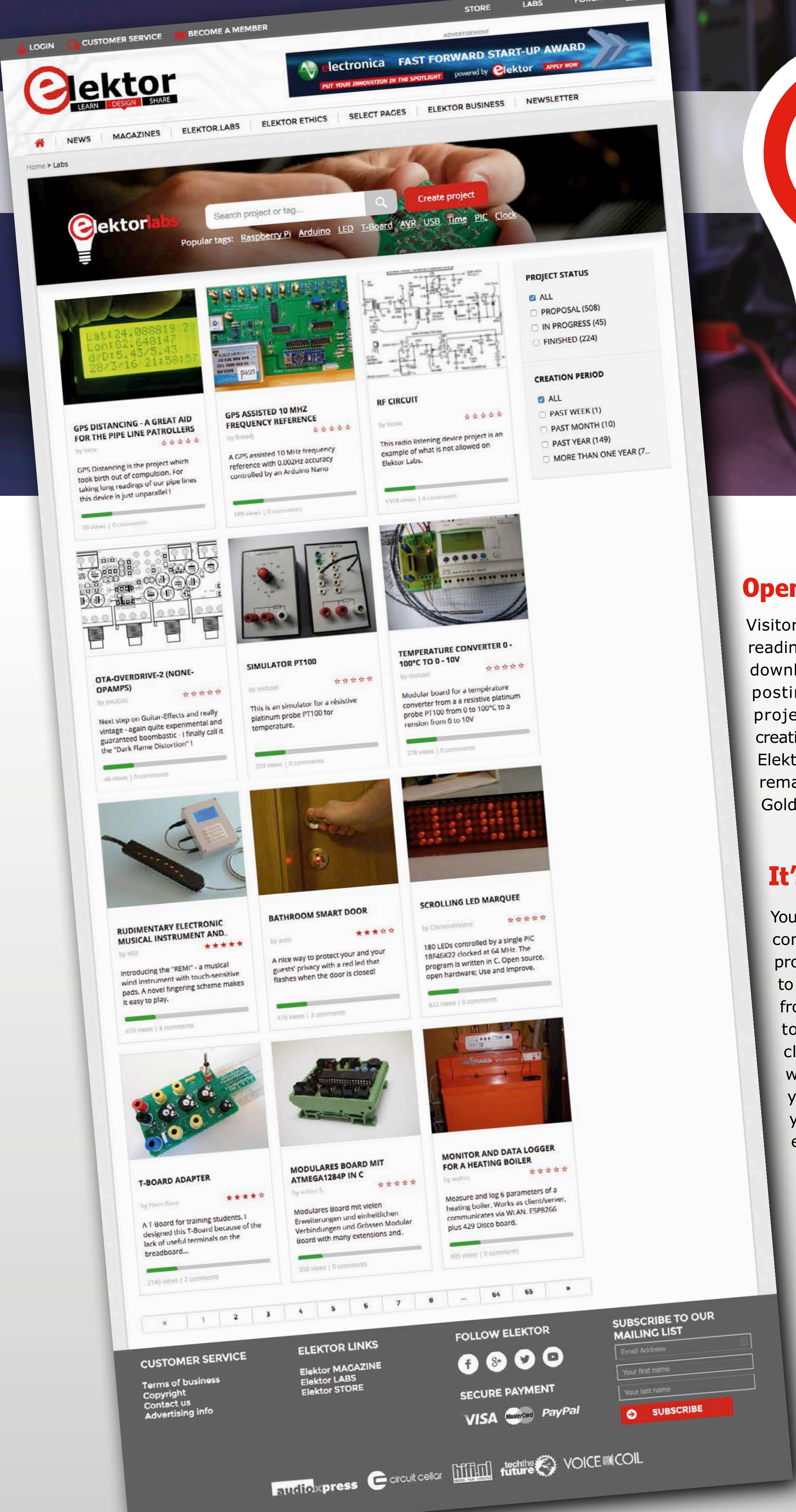
The above is only a small selection of battery and battery-related technologies mostly in development in laboratories at the time of writing, with a few happy examples of real-life products like Upp [2]. The existence of the latter does not mean wide acceptance by the consumer yet — you are unlikely to see a Bioo or a fuel cell in an airport departure lounge and it will be plain old AC power and lithium-ion ruling the roost there for some time to come.
Battery technology is very much alive and has great appeal to non-technical audiences, mostly due to the source of the electrical energy (after conversion), including unexpected ones like potatoes, human skin, human muscle flexing, street noise, onions, peptides, and mental activity. ⏮

(160013)

Web Links

[1] Bioo: www.bioo.tech/

[2] Upp: www.beupp.com/



LOGIN CUSTOMER SERVICE BECOME A MEMBER STORE LABS

ADVERTISEMENT

electronica FAST FORWARD START-UP AWARD powered by elektor APPLY NOW PUT YOUR INNOVATION IN THE SPOTLIGHT

elektor LEARN DESIGN SHARE

NEWS | MAGAZINES | ELEKTOR.LABS | ELEKTOR ETHICS | SELECT PAGES | ELEKTOR BUSINESS | NEWSLETTER

Home > Labs

elektorlabs

Search project or tag...

Create project

Popular tags: Raspberry Pi Arduino LED T-Board AVR USB Time PIC Clock

GPS DISTANCING - A GREAT AID FOR THE PIPE LINE PATROLLERS
by bera
GPS Distancing is the project which took birth out of compulsion. For taking long readings of our pipe lines this device is just unparallel !
20 views | 0 comments

GPS ASSISTED 10 MHZ FREQUENCY REFERENCE
by Breedj
A GPS assisted 10 MHz frequency reference with 0.002Hz accuracy controlled by an Arduino Nano
189 views | 0 comments

RF CIRCUIT
by Kosie
This radio listening device project is an example of what is not allowed on Elektor Labs.
1318 views | 4 comments

PROJECT STATUS
- ☑ ALL
- ☐ PROPOSAL (508)
- ☐ IN PROGRESS (45)
- ☐ FINISHED (224)

CREATION PERIOD
- ☑ ALL
- ☐ PAST WEEK (1)
- ☐ PAST MONTH (10)
- ☐ PAST YEAR (149)
- ☐ MORE THAN ONE YEAR (7..

OTA-OVERDRIVE-2 (NONE-OPAMPS)
by joe2030
Next step on Guitar-Effects and really vintage - again quite experimental and guaranteed boombastic - I finally call it the "Dark Flame Distortion" !
46 views | 0 comments

SIMULATOR PT100
by mickael
This is an simulator for a résistive platinum probe PT100 for temperature.
253 views | 0 comments

TEMPERATURE CONVERTER 0 - 100°C TO 0 - 10V
by mickael
Modular board for a température converter from a a resistive platinum probe PT100 from 0 to 100°C to a rension from 0 to 10V
278 views | 0 comments

RUDIMENTARY ELECTRONIC MUSICAL INSTRUMENT AND..
by MJB
Introducing the "REMI" - a musical wind instrument with touch-sensitive pads. A novel fingering scheme makes it easy to play.
619 views | 8 comments

BATHROOM SMART DOOR
by anto
A nice way to protect your and your guests' privacy with a red led that flashes when the door is closed!
479 views | 3 comments

SCROLLING LED MARQUEE
by ClemensValens
180 LEDs controlled by a single PIC 18F46K22 clocked at 64 MHz. The program is written in C. Open source, open hardware; Use and improve.
822 views | 0 comments

T-BOARD ADAPTER
by Hans Bans
A T-Board for training students. I designed this T-Board because of the lack of useful terminals on the breadboard...
2145 views | 2 comments

MODULARES BOARD MIT ATMEGA1284P IN C
by achim S.
Modulares Board mit vielen Erweiterungen und einheitlichen Verbindungen und Grössen Modular Board with many extensions and.
358 views | 0 comments

MONITOR AND DATA LOGGER FOR A HEATING BOILER
by waltro
Measure and log 6 parameters of a heating boiler. Works as client/server, communicates via WLAN. ESP8266 plus 429 Disco board.
405 views | 0 comments

« 1 2 3 4 5 6 7 8 ... 64 65 »

CUSTOMER SERVICE
Terms of business
Copyright
Contact us
Advertising info

ELEKTOR LINKS
Elektor MAGAZINE
Elektor LABS
Elektor STORE

FOLLOW ELEKTOR

SECURE PAYMENT
VISA MasterCard PayPal

SUBSCRIBE TO OUR MAILING LIST
Email Address
Your first name
Your last name
SUBSCRIBE

audioxpress circuit cellar hifi.nl techthefuture VOICE COIL

Open 24/7 to everyone

Visitors to Elektor.Labs are not limited to reading the project writeups, but can also download files, participate in projects by posting comments and even post new projects. The only requirement is the creation of a (free) Elektor ID. Access to the Elektor articles archive and other services remains a privilege of Elektor Green and Gold members.

It's all in your profile

Your (free) Elektor ID with its unique login consisting of user name and password provides links your Elektor.Labs account to your other Elektor accounts. Starting from your profile page you have access to all services granted to your ID. A click or tap on "My Labs Project(s)" will display a list of the projects that you manage and/or follow. From here you can open a project for viewing or editing.

Project editing

Updating a project is one through a *project comment*. This maintains a clear overview of each change or update to a project, enabling you to stay up to date with minimum effort. The main project page, or "Project Description" as it called from now on, can only be edited in draft

mode; once the project is published, it can only be modified in *draft mode*. Once published, only *project managers* can add *project comments*. Comments from followers and viewers are displayed next to the *project comments*.

Creating a new project…

… is easy. You start by clicking the red “Add project” button (at the top of the Elektor.Labs homepage) and then entering a title (choose it well!), a *teaser* and a description. The *teaser* is what people see on the project overview pages, so keep it clear and concise (not exceeding 150 characters).
Uploading a project picture is highly recommended. If you don’t have one yet, don’t worry, you can add it later. It is also possible to add downloadable files (photos, software, CAD files, etc.) to your project by clicking the *Attachments* button. *Attachments* can be assigned a type so that they will be displayed in certain areas.

Drafts

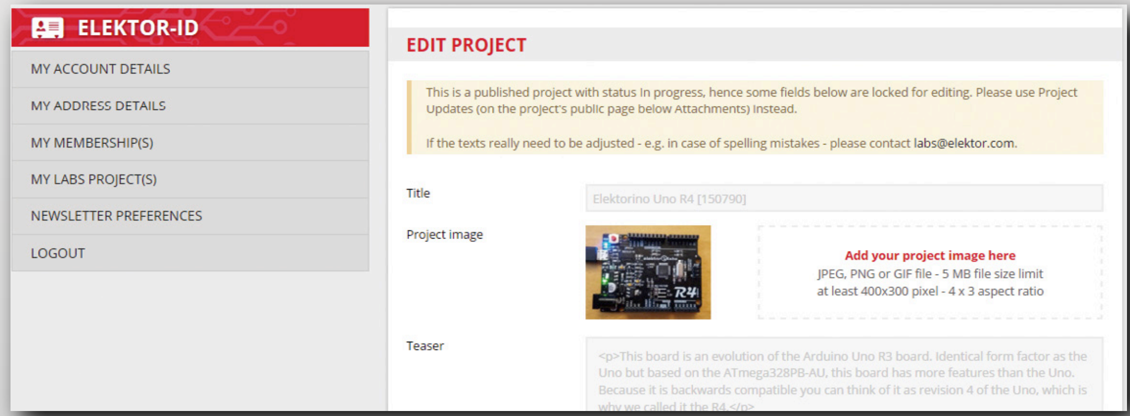
Projects in *draft mode* are not visible to the rest of the world and can be changed indefinitely. They can be changed without restriction or even deleted. *Draft mode* is useful if you want to present a completed project in one *post*. Simply preview and modify the project description until you are happy with it, then publish it. But beware, there is no way back; your project is now visible to the world and beyond.

Searching and filtering

Having lots of projects and information is nice, but when you cannot search through it in an efficient manner, it all remains pretty useless. That’s why we have added several search and filter tools. A good keyword search complemented by tags and multiple sorting options enable you to quickly find what you are looking for. ⏮

(160021)

www.elektormagazine.com/labs



EDIT PROJECT

This is a published project with status In progress, hence some fields below are locked for editing. Please use Project Updates (on the project's public page below Attachments) instead.

If the texts really need to be adjusted - e.g. in case of spelling mistakes - please contact labs@elektor.com.

Title: Elektorino Uno R4 [150790]

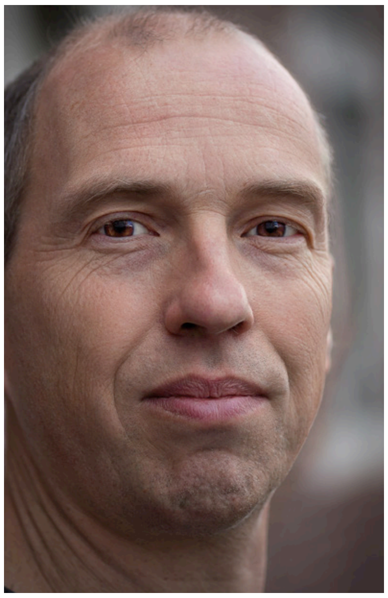
Project image

Add your project image here
JPEG, PNG or GIF file - 5 MB file size limit
at least 400x300 pixel - 4 x 3 aspect ratio

Teaser: <p>This board is an evolution of the Arduino Uno R3 board. Identical form factor as the Uno but based on the ATmega328PB-AU, this board has more features than the Uno. Because it is backwards compatible you can think of it as revision 4 of the Uno, which is why we called it the R4.</p>

Welcome to the DESIGN section

By **Clemens Valens,** Elektor Labs



André-Marie Ampère (1775 – 1836)

never went to school. Born in Lyon, France, André-Marie Ampère, AM to his friends, figured it out all by himself in the well-garnished family library. At the age of 12, AM becomes interested in mathematics and, a year later, publishes his first paper. During the French Revolution, AM, sixteen years old, proposes a new decimal measurement system. When he is 18 his father is guillotined and the family ruined. AM turns to studying botany, inventing scientific instruments and observing the stars; he learns Greek and Italian and creates a universal language. He studies Latin poetry and even writes a tragedy criticizing Christopher Columbus. Some people might say that his attention was dispersed, but not AM, who went on to study chemistry in general and carbon oxide in particular. When, at the age of about 25, AM finally goes to school, he goes as a teacher, not as a student.

In 1820 the Danish physicist Hans Christian Ørsted discovered that a magnetic needle is deflected by a nearby electric current. At that time AM lived in Paris and was working on speculative philosophy and its application to other fields of science. Even without Twitter, Ørsted's discovery went viral and AM learned about it at the beginning of September 1820. Inspired by the exciting news AM dropped everything and set off to repeat Ørsted's experiments. He realized quickly that a wire carrying a current behaves like a magnet. From this he concluded that two of these wires in parallel should attract or repel each other depending on whether the currents flow in the same direction or not. Also he deduced that such a wire would be influenced by the Earth's magnetic field, exactly like a compass needle. About a week later AM presented his first findings. AM was on a roll and after another week of frantic work presented a second paper in which he coined the term *electrical current*. In the weeks that followed he invented techniques for measuring such currents and for magnetizing steel. By the end of 1820 AM wrapped it all up in a paper and the field of electrodynamics was born.

During the following years AM continued to experiment with electricity and magnetism and came up with the hypothesis that electrical current consists of some sort of electrodynamic molecule being pushed through a conductor. In 1827 he published his magnum opus, *Memoir on the Mathematical Theory of Electrodynamic Phenomena, Uniquely Deduced from Experience*, which is considered to be the founding treatise of electrodynamics.

Being an SI base unit, today Ampère's name is mainly connected to electrical current, but his scientific oeuvre goes way beyond that. He has made important contributions to mathematics and physics, like playing a major role in the discoveries of chlorine, fluorine and iodine. André-Marie Ampère was one of those truly great minds that gave science a big push forward. ⏮

(160015)

MAXQ to the Rescue

A multimeter gets a new lease on life

A while ago I picked up a second-hand Philips system multimeter on Marktplaats (the Dutch equivalent of eBay) with a minor defect for next to nothing. The display was no longer legible, but I thought it would be easy to repair. That turned out to be a major error in judgement — no matter what I tried, there was no way to rescue the LCD. I therefore needed a different solution using modern components.

By **Fons Janssen**,
Maxim Integrated (Netherlands)

The nice thing about test equipment from the 1980s and 1990s is that it is mainly built with standard components and the service documents are often available online. After studying the service manual for my meter, a Philips PM2535 [1], I decided it must be possible to connect a different display module.

The standard display architecture

The LCD module in the meter is controlled by a Philips PCF8576 LCD driver IC. It receives its data from the main processor over an I²C bus. The idea was to tap this data stream, decode it, and then display the data on a different module.
The PCF8576 can drive up to 160 segments on an LCD module. The service manual describes exactly how the display segments are linked to the memory in the

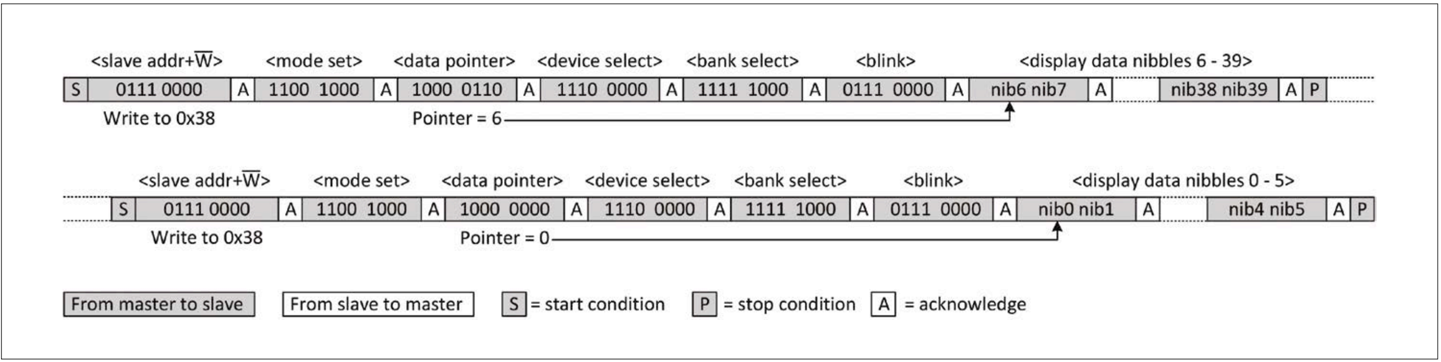

<slave addr+W̄> <mode set> <data pointer> <device select> <bank select> <blink> <display data nibbles 6 - 39>
S 0111 0000 A 1100 1000 A 1000 0110 A 1110 0000 A 1111 1000 A 0111 0000 A nib6 nib7 A nib38 nib39 A P
Write to 0x38
Pointer = 6
<slave addr+W̄> <mode set> <data pointer> <device select> <bank select> <blink> <display data nibbles 0 - 5>
S 0111 0000 A 1100 1000 A 1000 0000 A 1110 0000 A 1111 1000 A 0111 0000 A nib0 nib1 A nib4 nib5 A P
Write to 0x38
Pointer = 0
From master to slave
From slave to master
S = start condition
P = stop condition
A = acknowledge

Figure 1. The I²C data stream to the display driver.

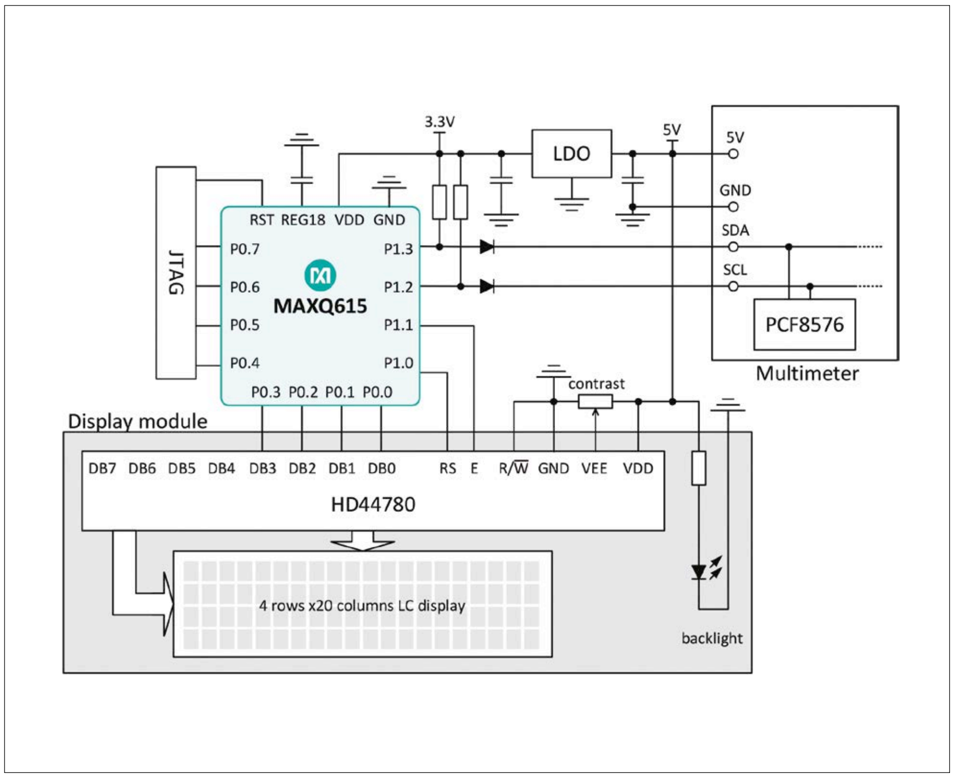


Figure 2. The hardware for driving the new display. The display module integrates an HD44780 display driver, a 4-line by 20-character LCD and a backlight. It is connected to the multimeter by four lines: 5 V and GND for power; SDA and SCL for the data link.

The original LCD was replaced by an HD44780-based display with 4 lines of 20 characters.

driver IC, which consists of a matrix of 40 rows of 4-bit nibbles. Each bit in the memory determines whether the corresponding segment is visible or hidden.

The data sheet for the PCF8576 [2] describes the protocol for using I²C commands to fill the display memory. First one or more command bytes are sent, followed by data bytes. With the aid of a logic analyzer, it quickly became apparent that the data is written to the display driver in two sessions. The first session is for nibbles 6 to 39, and the second session is for nibbles 0 to 5. These sessions are repeated over and over to continuously refresh the display. The data stream is shown schematically in **Figure 1**.

The <mode set>, <device select>, <bank select> and <blink> commands are configuration commands for the LCD driver. They always have the same values and are not important for our purposes. However, the <data pointer> command is important because it indicates where the data will be written in the memory. The commands are followed by the data bytes, with each byte containing two data nibbles. They are automatically written to the right locations in the display memory.

The new display

I chose an HD44780-based display module with four rows of 20 characters as a replacement for the original module. It has just enough room to display all of the original data. A microcontroller acting as an I²C slave device can be used to tap the data stream to the PCF8576. It can also decode this data and write it to the new display module. For this task I chose a MAXQ615 from Maxim Integrated. Along with an I²C port, it has exactly enough I/O pins to drive the display module in 4-bit mode.

Figure 2 shows a simplified schematic diagram of the hardware. It is built on a MAXQ615 evaluation board, which has a perfboard prototyping area where the additional components are mounted.

The MAXQ615 operates at 3.3 V and is therefore not compatible with the 5 V logic levels used in the meter. Since the microcontroller only has to receive signals, this problem can be solved easily by giving the SDA and SDL pins their own pull-up resistors and isolation diodes. When the I²C master pulls these lines low, the diodes conduct and the levels on the microcontroller inputs are also low. When the I²C master sets these lines high, the diodes are reverse biased and the level is limited to 3.3 V. When the MAXQ615 pulls the lines low (to acknowledge a command, for example), the diodes are also reverse biased. This prevents the MAXQ615 from “talking back” and disturbing communications between the main processor and the PCF8576.

The firmware

The MAXQ615 has a hardware-based I²C interface, so the firmware only needs to initialize a few registers to enable I²C functionality and configure the IC as a slave device.

In the main routine the microcontroller waits for data on the I²C interface. When the I²C hardware recognizes its own slave address (the same as that of the PDF8576), the firmware knows that the next byte will be a command byte. All following command bytes are ignored except the data pointer command, which is used to determine where to store the data. The data bytes come after the last command byte in the string. The data bytes are stored in an array of 20 bytes (two nibbles per byte). At the byte level this array is a copy of the display memory in the PCF8576, with the associated pointer forming half of the data pointer at the nibble level.
As previously mentioned, the data is sent in two sessions. After both of these data blocks have been received, the data can be processed. The content for the new display is held in a string of 80 characters. The firmware scrolls through the received data and determines what has to be shown on the display and where it

has to be shown. That is fairly easy for simple on/off segments, but it is more complicated for 7-segment and 16-segment characters. The bits belonging to a 7-segment character are grouped into one byte in the array, while the bits for a 16-segment character are grouped into two bytes. The characters can be derived from the byte values in a straightforward manner by using a look-up table. Once the final character has been determined, the display string is copied to the new LCD module in a single operation. The entire data processing architecture is shown in schematic form in **Figure 3**.

The firmware for the MAXQ615 is available on the Elektor website [3], although it's unlikely that many readers will have the same meter type with the same defect.

Installation

To make room for the new display, the original display was sawn out of the circuit board and moved to a different place in the enclosure, along with the MAXQ615 evaluation board. Then the new display was glued into the freed-up space.

As you can see from the photos of the meter with the original defective display (**Figure 4**) and the new display (intro photo), the replacement display is not only legible but also distinctly easier to read thanks to the backlighting.

What's next?

Now that the meter is again fully operational, it's time to think about potential improvements. Perhaps the reference source could be upgraded? So stay tuned – there may be a follow-up article.

(150766)

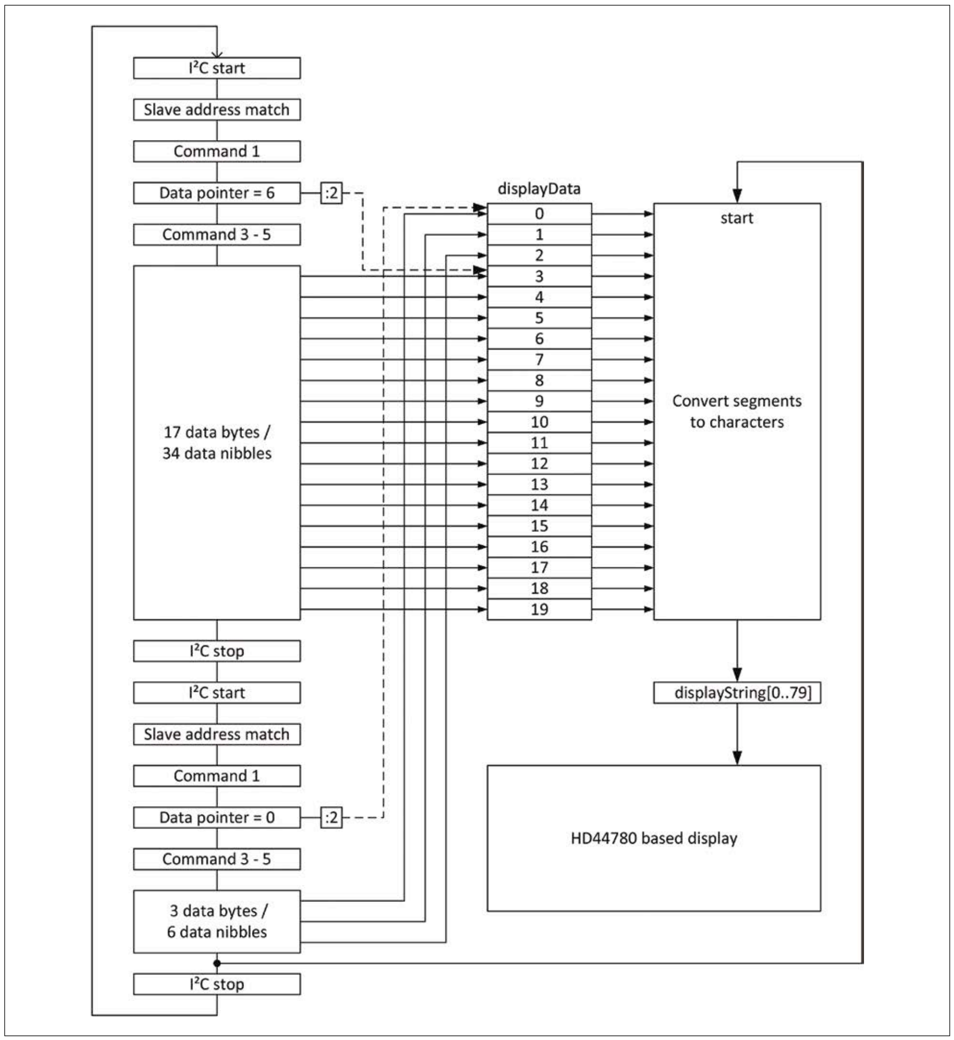


Figure 3. Schematic representation of the data processing architecture.

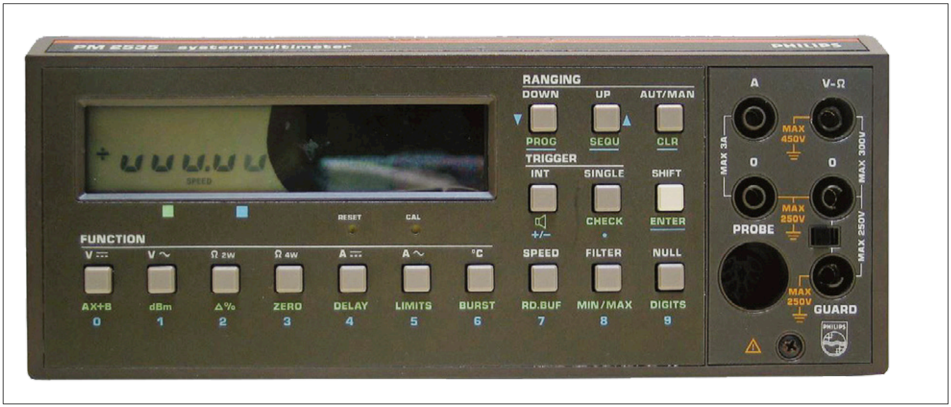


Figure 4. The meter with the defective display, which was beyond repair.

Web Links

[1] Philips System Multimeter Service Manual PM2534-PM2535, No. 4822 872 35313 900205: www.download-service-manuals.com/download.php?file=Philips-6930.pdf

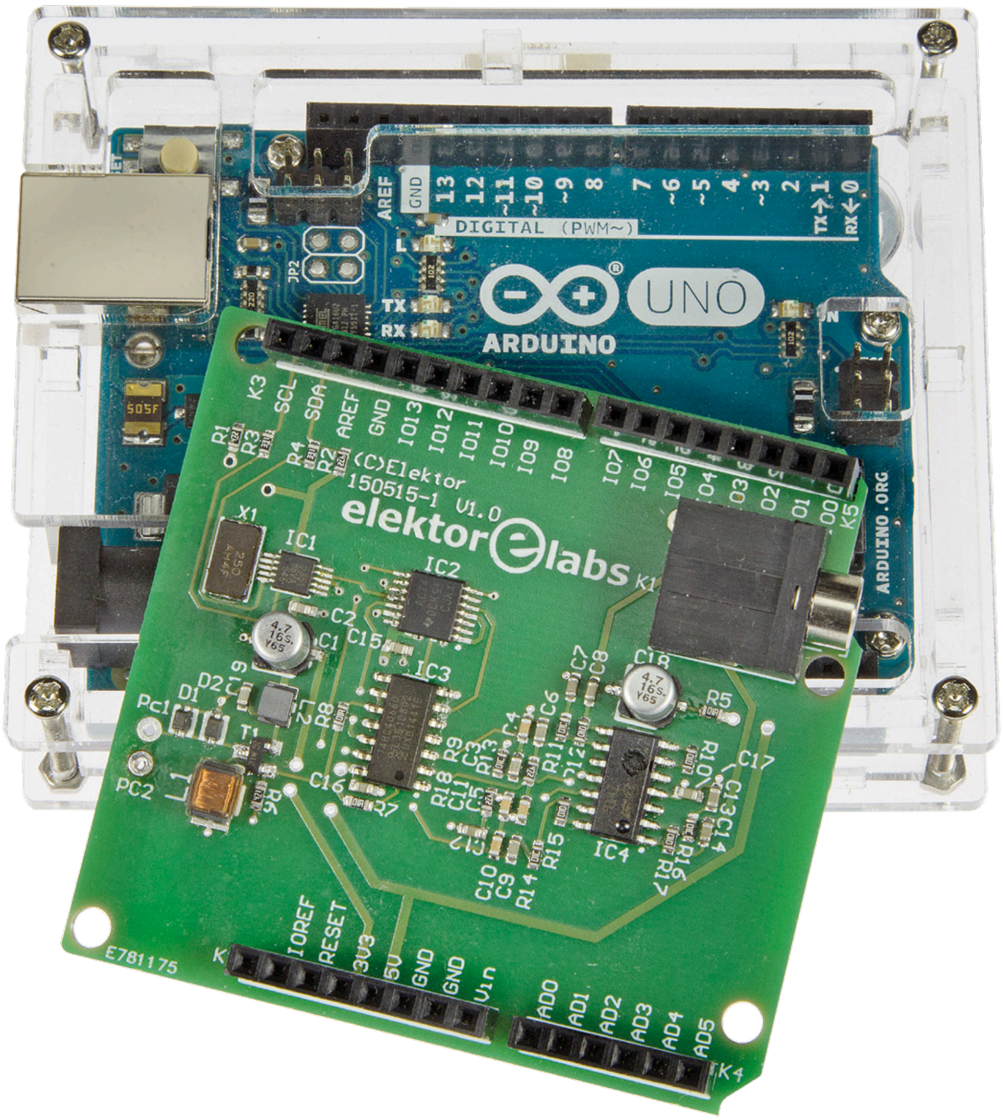
[2] PCF8576 data sheet: www.nxp.com/documents/data_sheet/PCF8576.pdf

[3] Software: www.elektormagazine.com/150766

Elektor SDR Reloaded

SDR Shield for the Arduino

By **Burkhard Kainka** (Germany)



A Software Defined Radio is a universal tool in RF technology circles, one that can also be put to use for making measurements. The characteristics of the receiver are defined in software, which now gives us the opportunity to use an Arduino Shield as a front-end.

Technical Characteristics

- Supply voltage: 5 V and 3.3 V as for Arduino
- Frequency range: 150 kHz up to 30 MHz
- Sensitivity: 1 µV
- Total amplification: 40 dB
- Maximum antenna signal level: 10 mV
- Dynamic range: 80 dB

Even though broadcast services are deserting the AM domains in the long, medium and short wavebands, there is still plenty of interest to be found surfing the radio waves with a home-constructed receiver. Now more than ever you might say, because many distant stations now come up far more clearly because they are no longer swamped by stronger signals. In fact it is often so quiet on the short waves that it's easy to imagine your receiver has gone deaf. On some bands it is the radio amateurs who produce the strongest signals. And there is always something new to find, from pirate radio stations through SSB radiotelephony to the new digital modes. That just has to make you curious!

Elektor has already published many radio and receiver projects. A Software Defined Radio with USB interface was introduced as long ago as 2007 [1]. In the meantime much thought has been devoted to conceiving updates for this design. However, the PLL chip we used originally is no longer made, making it necessary to find a new solution. This has arrived in the form of the Silicon Lab SI5351 chip, a CMOS clock generator from 8 kHz to 160 MHz with I^{2}C bus.

First investigations revolved around a break-out board from Adafruit. The available sample software was written for the Arduino, so our first steps were undertaken with the Arduino. The new VFO was simply hooked up to the old SDR PCB and proved its suitability (**Figure 1**).

And then there came an idea: why not simply build the entire receiver as an Arduino Shield? This decided the power supply requirements, using the USB interface already available on the PC. The Arduino would look after controlling the VFO and could be addressed in plain language so to speak (6030 kHz please). And what is perhaps even more exciting, this even gives you a real chance to build a totally standalone receiver. Operation could be migrated from the PC to the Arduino relatively simply. And who knows, perhaps one day the decoding of the IQ signal as well?

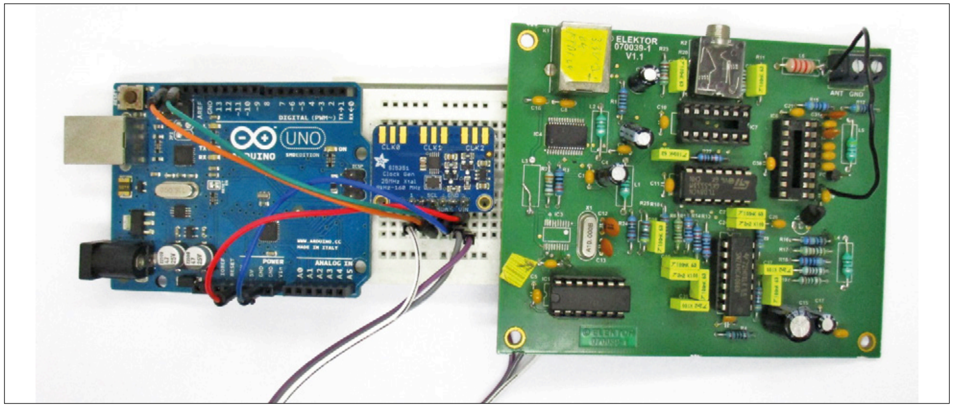


Figure 1. First preliminary test for an SDR2: an SI5351 PLL chip hooked up to an Arduino Uno and the 'old' SDR receiver.

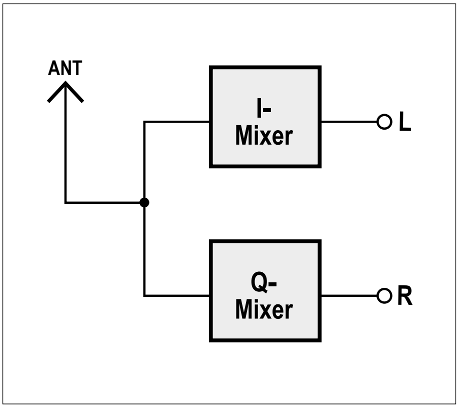


Figure 2. Working principles: the front-end consists of a dual direct mixer with signals shifted 90 degrees out of phase.

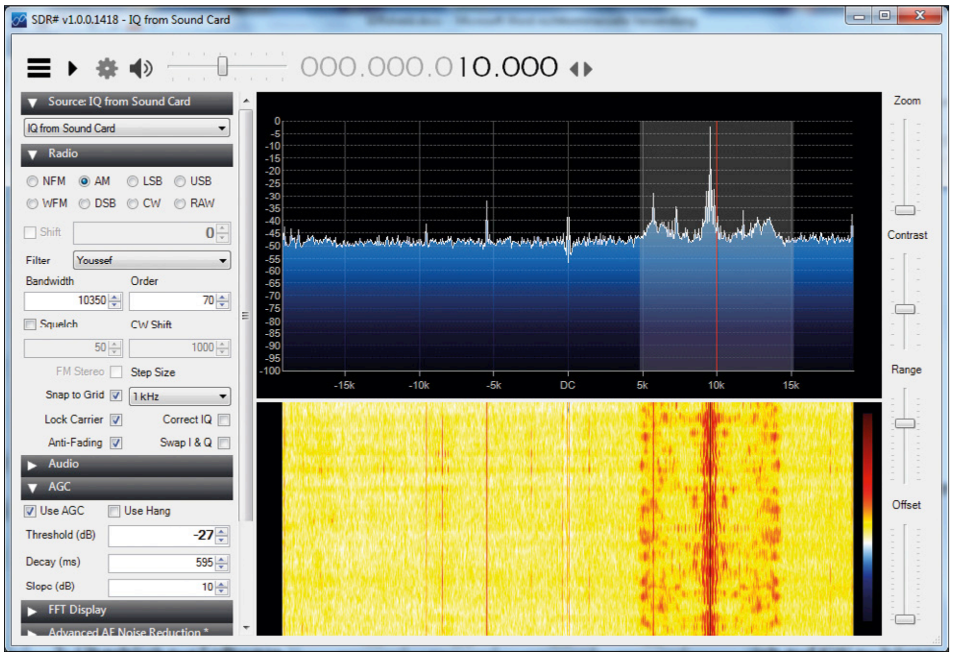
So how does it work?

First, let's go back to basics. What fundamentally is a Software Defined Radio? For quite some time the development of digital electronics left radios entirely untouched. When home computers first became available, most radios were still analog. Then development began to take place, at least for digitizing their tuning. Today's radios are often equipped with a PLL synthesizer that simplifies tuning and guarantees precise conformity to channel spacing. The rest of the circuitry remains analog as previously.

Subsequently digital electronics appeared inside commercial RF equipment and amateur radio gear. Ever more of the analog functions in devices were replaced by software. In most cases a digital signal processor (DSP) with appropriate software operates out of sight from the user and takes care of optimal filter curves, variable bandwidth, signal decoding, interference suppression and much more. The equipment is altogether improved and with less hardware overhead. Further examples of this kind of development can be found in smartphones and other portable end-devices. At the same time it's evident that hobby constructors can no longer keep pace with this technology. In fact things don't need to be so sophisticated, however. All you need is a rapid A-D converter connected direct to the antenna. The entire spectrum is digitalized and then further processed digitally. In fact technology of this kind is available for the entire frequency range from 0 to 30 MHz. It's software alone that filters out specified frequencies and demodulates the desired signal. Regardless of whether we are dealing with AM or DRM broadcast stations or whether we're receiving SSB signals, CW Morse transmissions, teletype (RTTY), weather fax or whatever else, everything is feasible. There is appropriate software for everything. Sure, it must be conceded that with such a large bandwidth, the hardware can be quite expensive, and the further processing necessary for this broad spectrum imposes high demands.

Nevertheless a cunning way around this challenge can be found in the sound cards of modern PCs. Using the 96 kHz sampling rate that's normal today you can already receive the whole frequency range up to 48 kHz. Instead of using a microphone, you simply connect a large coil as an antenna and immediately you're able to receive the VLF band. Down there are plenty of interesting signals, even transmitters installed on submarines.

If you want to use the sound card for higher frequencies, you must first downconvert the signals. The process is akin to a superhet with a lower intermediate frequency (IF). The PC handles the IF stages, filtering, automatic gain control (AGC) and demodulation. In principle a simple direct mixer with a diode ring mixer or the well-known NE612 would be adequate for this. Only a stable variable oscillator (VFO) would be needed in addition. For special applications you could



SDR# v1.0.0.1418 - IQ from Sound Card

000.000.010.000

Source: IQ from Sound Card
IQ from Sound Card
Radio
NFM [✓ AM] LSB USB
WFM DSB CW RAW
Shift 0
Filter Youssef
Bandwidth 10350
Order 70
Squelch 50
CW Shift 1000
FM Stereo
Step Size 1 kHz
Snap to Grid ✓
Lock Carrier ✓
Correct IQ
Anti-Fading ✓
Swap I & Q
AGC
Use AGC ✓
Use Hang
Threshold (dB) -27
Decay (ms) 595
Slope (dB) 10

Figure 3. The SDR# program receiving an AM signal.

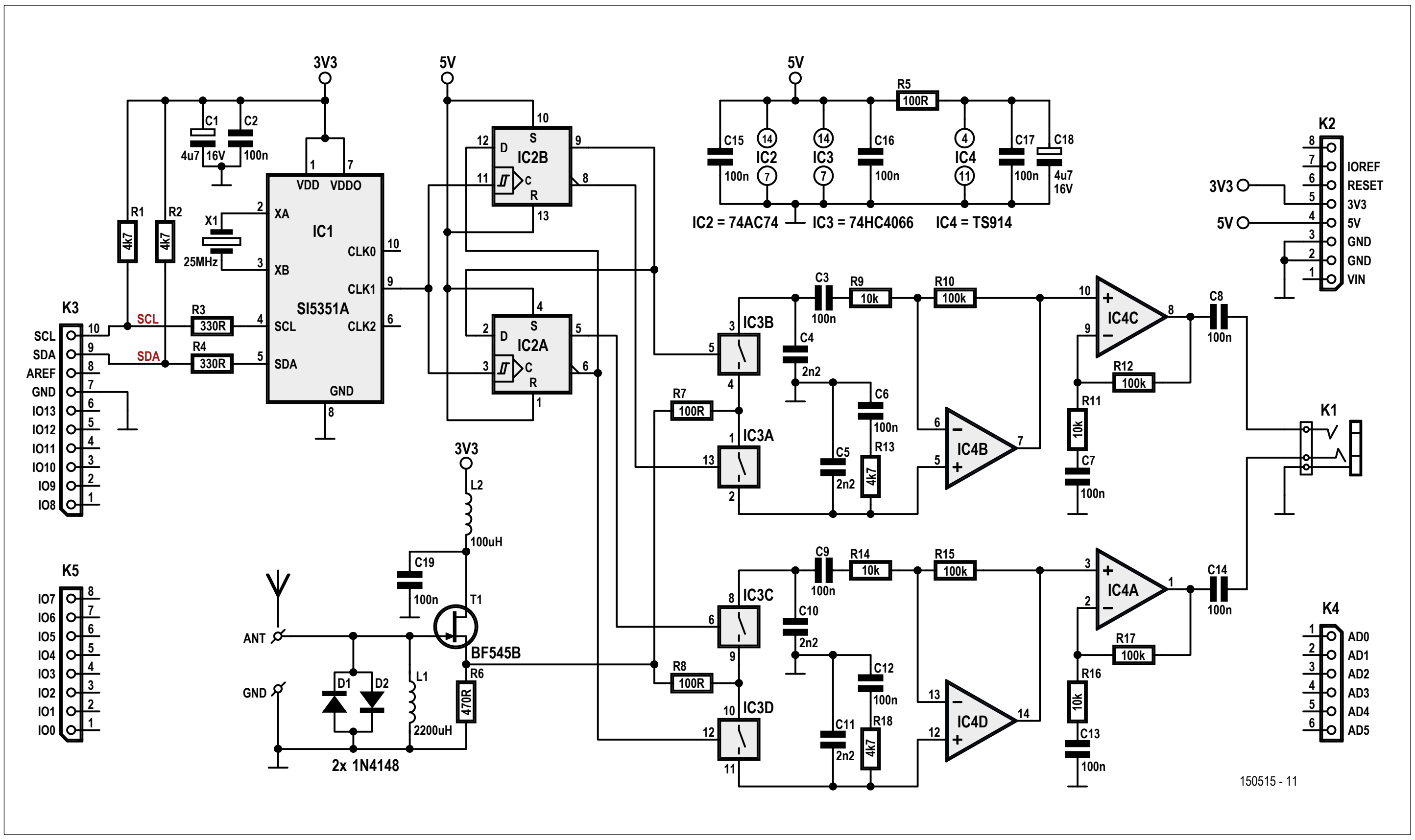


Figure 4. Schematic of the new SDR receiver.

use a crystal oscillator. But if you want to be able to tune an entire band, it ought to be a DDS generator or a PLL module. Simply stated, an IQ mixer is a matter of a dual direct mixer with two signals phase-shifted by 90 degrees. The oscillator signal is always adjacent to the reception frequency. The output signals therefore lie within the AF range, mostly between 0 kHz and 24 kHz. The two signals are designated I and Q (see **Figure 2**). These are applied direct to the

Component List

Resistors
R1,R2,R13,R18 = 4.7kΩ 1%, 0.1W, SMD 0603
R3,R4 = 330Ω 1%, 0.1W, SMD 0603
R5,R7,R8 = 100Ω 1%, 0.1W, SMD 0603
R6 = 470 Ω 1%, 0.1W, SMD 0603
R9,R11,R14,R16 = 10kΩ 1%, 0.1W, SMD 0603
R10,R12,R15,R17 = 100kΩ, 0.1W, SMD 0603

Capacitors
C1,C18 = 4.7µF 16V, SMD case B
C2,C3,C6,C7,C8,C9,C12,C13,C14,C15,C16,C17,C19 = 100nF 50V, X7R, SMD 0603
C4,C5,C10,C11 = 2.2nF 50V, X7R, SMD 0603

Inductors
L1 = 2200µH (Fastron L-1812AF)
L2 = 100µH (Murata LQH32CN101K23L)

Semiconductors
D1,D2 = 1N4148WS, SOD-323
T1 = BF545B, SOT-23
IC1 = SI5351A-B-GT, MSOP-10
IC2 = SN74AC74PW, TSSOP-14
IC3 = 74HC4066, SOIC-14
IC4 = TI914IDT, SOIC-14

Miscellaneous
K1 = stereo jack socket, 3.5mm, PCB mount
K2,K3,K4,K5 = connector set, Arduino compatible (1 pc. 6-pin, 2 pcs. 8-pin, 1 pc. 10-pin)
X1 = 25MHz quartz crystal (Abracon ABM7)

PCB # 150515-1
Or
PCB with preassembled SMD parts: 150515-91

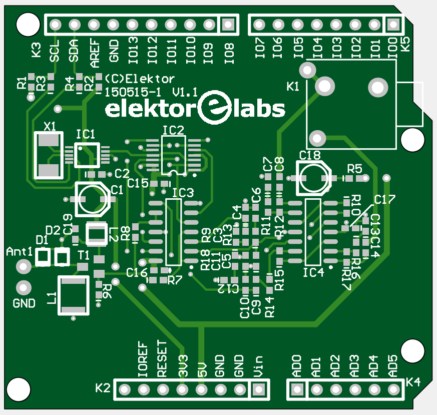
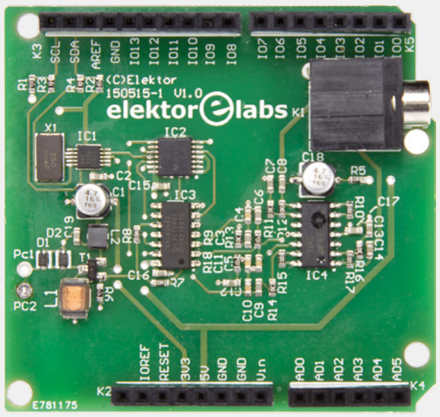



Figure 5. Double-sided PCB for the SDR3 executed as an Arduino Shield.

left and right channels of the sound card input. The rest is dealt with in software. A simple mixer would mix the range below and above the oscillator frequency into the same range, in which the dreaded image frequency problem would arise. By carrying out dual mixing and phase shifting, the software is able to cancel out and eliminate the image frequency, however. In this way a range between –24 kHz and +24 kHz can be received if the sound card has a sampling rate of 48 kHz. **Figure 3** shows what a program like SDR# makes out of this (see also the text panel *SDR software*).

Circuit

A glance at the schematic in **Figure 4** shows the individual building blocks. The SI5351 PLL generator (IC1) delivers the oscillator signal with the 4x receive frequency to the 74AC74 divider (IC2B). This divides the frequency by four and delivers the signals phase-shifted by 90 degrees to the 74HC4066 mixer (IC3). This analog switch is wired as a changeover switch and applies the RF signal alternately to the inverting and non-inverting inputs of the TS914 op-amp (IC4B/IC4D). In this way the signal is mixed down into the AF region. After some modest filtering and amplification (IC4C/IC4A), the signal reaches the audio output. The RF input stage creates a source follower using the BF545B JFET (T1), the SMD equivalent of the BF245B. Anyone familiar with the old Elektor SDR will see a certain simplification in the signal path. On its RF input it had several switchable lowpass filters. The new design has a wideband input and is protected against over-voltage by two diodes. This is completely adequate for shortwave reception with a wire antenna. The additional over-voltage protection was born from experience with the first SDR; in a thunderstorm the input stages could be damaged. For specific purposes you can also use additional external filters and a preamp. In the old version the AF amplification could be adjusted in three steps. This time around there is only the middle level, which proved itself to be fine for general use. So everything has become a bit simpler and now works well with the Shield.

For first trials you simply need to connect up a wire antenna. Some wire with a length of three meters hung from the ceiling will be fine. If this is impossible a longer piece of wire lying anywhere in the room should work. Admittedly indoor antennas suffer greater interference levels and advice on how to make optimized antennas will be printed soon in Elektor.

Construction

The PCB (**Figure 5**) is designed as an Arduino Shield, enabling it to be plugged straight into an Arduino Uno. As the SI5351 is available only as a miniature 10-pin SMD package, we took the decision to design the complete circuit with SMDs and to offer the ready-assembled PCB in the Elektor Shop [3]. Beyond this your only other action is to solder the four Arduino-compatible female headers onto the PCB. Anyone who would prefer to build the PCB completely unassisted can download the PCB layout at [3] or you can buy the PCB on its own from the Elektor Shop.

Setting the frequency

The Arduino, in conjunction with the SDR Shield, serves as an interface between antenna and PC. Its sole task is to tune the VFO and to do this, a PC program tells it which frequency is currently desired. Data from the PC reaches the Arduino via the USB cable. The downconverted signal of interest is then sent for further processing along a stereo cable to the sound card input. You could certainly attempt to relocate the control function to the Arduino as well, perhaps even some simple signal processing but this would be hard labor for a small system. For the moment it is sufficient for the Arduino to receive commands from the PC and adjust the VFO.

How you deal with the Arduino is not a topic for this article. Using the Arduino IDE is presupposed. First of all a suitable Arduino program is loaded. Exactly what happens in the software will be explained presently. However, if you're in a rush for practical results, you can skip this information and simply load the software [3].

The critical task lies in persuading the SI5351 to generate an appropriate frequency. The IC has two internal PLLs and three outputs (see block diagram, **Figure 6**). Here only PLL A and the output CLK1 are used. The concept makes use of the Adafruit Library, which makes the whole business delightfully simple. Before you let loose, you must download the Library from [2] and integrate it.

The SI5351 has a 25 MHz crystal oscillator and two PLLs that can be set between 600 MHz and 900 MHz. The PLL dividers operate with fractional division ratios, so it is in fact possible to achieve almost any desired resolution. The following multisynth divider also uses fractional division ratios. This gives you two ways to generate the desired frequency:

- You can set the PLL on a fixed frequency e.g. 900MHz and then divide

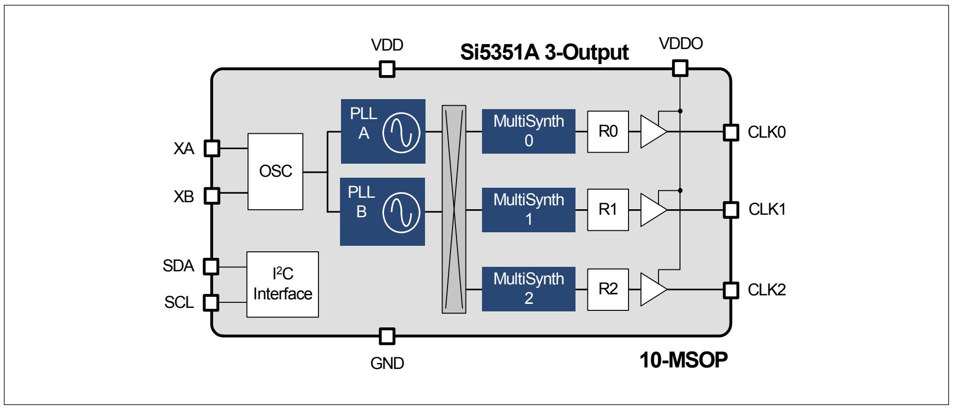
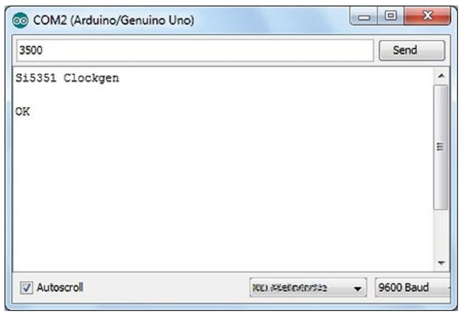


Figure 6. Block diagram from the data sheet for the SI5351A.



```
3500
```

```
Si5351 Clockgen

OK
```

Figure 7. Activating the clock generator with the Arduino terminal.

down with fractional numbers.

- You can adjust the PLL in small steps and then divide by integers to reach the final frequency.

First, here is method A. The VFO frequency is four times the mixer frequency, which is 12 kHz below the receive frequency. The program is arranged to receive the radio frequency in kHz and implement in text format. In order to receive 3500 kHz the SI5351 must produce at output 1 of the SI5351 must produce an output frequency of 4 × (3500-12) = 13952 kHz. The PLL divider is set to 36 (25 MHz × 36 = 900 MHz) and the multisynth divider 900000/13952 = 64.506. Using this method we can get down to 1 MHz. For even smaller frequencies the additional R_DIV divider is set to 16. **Listing 1** indicates the relevant

Listing 1. Program for fixed-tuned PLL.

```
//SI5351_vfo  PLL fixed at 900 MHz (si5351vfo2.zip)

#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <Adafruit_SI5351.h>

Adafruit_SI5351 clockgen = Adafruit_SI5351();
void setup(void)

{
  Serial.begin(9600);
  Serial.println("Si5351 Clockgen"); Serial.println("");


  /* Initialise the sensor */
  if (clockgen.begin() != ERROR_NONE)
  {
    Serial.print("Error");
    while(1);
  }
  Serial.println("OK");
  clockgen.enableOutputs(true);
  clockgen.setupPLL(SI5351_PLL_A, 36, 0, 1000);  //900
MHz
  setfreq (6000);
}

void setfreq (unsigned long freq)
{
  unsigned long f2;
  unsigned long f3;
  unsigned long f4;
  unsigned long f5;
  unsigned long div2;
  unsigned int Divider2;
  unsigned int rdiv;

   if (freq > 0)    {
    f2=(freq-12)*4;
    if (f2<1000) {
      rdiv = 16;
      f2 = f2 * 16;
      }
    else  {
      rdiv = 1;
      }
    div2 = 900000000/f2;
    f4 = div2/1000;
    f5=div2-(f4*1000);
    clockgen.setupMultisynth(1, SI5351_PLL_A, f4, f5,
1000);
    if (rdiv == 16) {
      clockgen.setupRdiv(1, SI5351_R_DIV_16);
      }
    if (rdiv == 1) {
      clockgen.setupRdiv(1, SI5351_R_DIV_1);
      }
   }
}

void loop(void)
{
  unsigned long freq;
  if (Serial.available()) {
   freq = Serial.parseInt();
   setfreq (freq);
   }
}
```

software for the Arduino; **Figure 7** shows the control panel on the Arduino terminal. Method A has the advantage that the VFO can be adjusted more or less continuously, i.e. there is no interruption when a change of frequency occurs.
On the other method B promises greater phase accuracy, adequate even for DRM. Against this, every frequency change is accompanied by a brief interruption of about a millisecond, which appears an interference signal on the SDR. The method requires calculation of the optimal fractional division (**Listing 2**) in order to keep the PLL constantly in the range 600 MHz to 900 MHz.

Both programs can be controlled by any Terminal program of your choice. However, for really convenient operation a VB program was written in Visual Studio 2015 (SDRShield.zip, downloadable at [3]). This sends the desired frequency to the Arduino in text format (e.g. 3500) at 9600 Baud. The slider control (see **Figure 8**) operates in 9 kHz steps in a range up to 1.6 MHz and beyond that with 5 kHz resolution. Additionally you can enter a desired frequency direct or click the 'band' buttons at the beginning of the individual broadcast or amateur radio bands. The first time you do this, take care that you have selected the correct COM Port.

SDR software

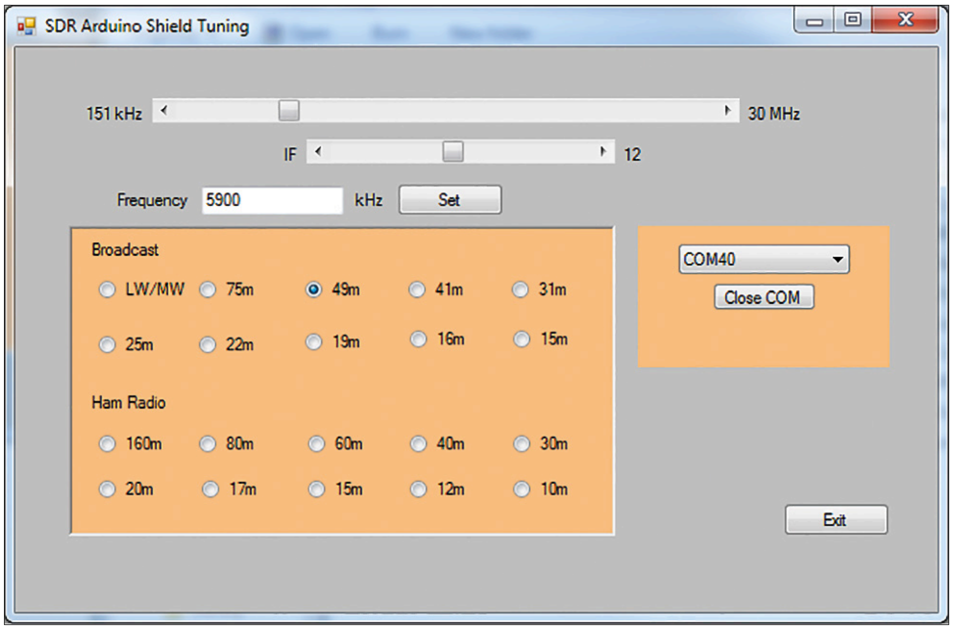
Here is a survey of the SDR software used. Practically all the programs used with the old Elektor SDR still work fine.

- *SDRadio* is still a good choice;
- *SoDoRa* can also decode DRM;
- *DREAM* still works but does not make use of the IQ signal and uses the receiver like a direct mixer;
- *HDSDR* is a current and very powerful program;
- *SDRSharp (SDR#)* is distinguished by simple operation and good on-screen representation.

In a follow-up article we shall discuss these individual programs and their possibilities in detail.

First results with reception

If you have no better antenna to hand, for your first tests you can simply connect a one-to-three meter length of wire to the antenna input. This will enable you to receive all broadcast stations on all shortwave bands without any problem. Experience shows that there is more happening at night than during the daytime. And during the evening we find the main focus is on the lower bands, between 75 m and 41 m. Even amateur radio stations can be pulled in with just a short wire aerial. Normally you'll have the best luck in the 40 meter band, where you can hear some CW and SSB stations. You can select the appropriate operational mode in the SDR software, plus audio volume, bandwidth, ALC settings and much more. With the correct settings you can often get better results than with an expensive analog receiver of the older kind.
A fundamental characteristic of all switching mixers is that signals on uneven (odd) multiples of the base frequency can also



Frequency 5900 kHz
[✓ 49m]
COM40

Figure 8. A short VB program ensures ease of operation.

SDR software

The current stars in the SDR software firmament are SDR# [4] and HDSDR [5]. Both programs follow the new trend for ever higher frequencies and can be driven using simple DVB-T [6] dongles. This is a good choice if you wish to poke around on the VHF and UHF bands. There have also been attempts to use this kind of hardware below 30 MHz. You can, for example, use an up-mixer that shifts every frequency 50 MHz higher. You do of course then have a multiple conversion superhet along with its well-known problems, such as countless phantom and spurious signals together with reduced dynamic range. A dedicated SDR for frequencies up to 30 MHz uses single conversion only and in that way manages to deliver very clean reception without 'birdy' whistles.

On your PC you have two programs running, namely the tuning program and the SDR software. Each SDR program has its own method of operation but the basic steps are nevertheless similar in each case. First you need to establish that the correct input is in use. For this you need to select the sound card and activate the chosen input (Line In). Next you boot up the SDR software. You'll know you have selected the correct input when you see a significant rise in the noise floor, which should increase still more after connecting the antenna. Most sound cards need to have their volume control throttled back, as the receiver can deliver up to a volt of output signal.

Listing 2. Program for variable PLL.

```
//SI5351_vfo, variable PLL (si5351vfo3.zip)

#include <Adafruit_Sensor.h>

#include <Wire.h>
#include <Adafruit_SI5351.h>

Adafruit_SI5351 clockgen = Adafruit_SI5351();
void setup(void)

{
  Serial.begin(9600);
  Serial.println("Si5351 VFO"); Serial.println("");

  if (clockgen.begin() != ERROR_NONE)
  {
    Serial.print("Error");
    while(1);
  }
  Serial.println("OK");
  clockgen.enableOutputs(true);
  setfreq (6000);
}

void setfreq (unsigned long freq)
{
  unsigned long f2;
  unsigned long f3;
  unsigned long f4;
  unsigned long f5;
  unsigned int Divider2;
  unsigned int rdiv;

   if (freq > 0)
   {
   f2=(freq-12)*4;
   // f2=freq;
    if (f2>120000) {
      f2=120000;
    }
    if (f2<800) {
       rdiv = 16;
       f2 = f2 * 16;
    }
    else  {
       clockgen.setupRdiv(1, SI5351_R_DIV_1);
       rdiv = 1;
    }
    if (f2 >= 100000) {
      Divider2 = 6;
     }
     if (f2 < 90000) {
      Divider2 = 10;
     }
     if (f2 < 60000) {
      Divider2 = 15;
     }
     if (f2 < 50000) {
      Divider2 = 18;
     }
     if (f2 < 45000) {
      Divider2 = 20;
     }
     if (f2 < 30000) {
      Divider2 = 30;
     }
     if (f2 < 20000) {
      Divider2 = 45;
     }
     if (f2 < 15000) {
      Divider2 = 60;
     }
     if (f2 < 10000) {
      Divider2 = 90;
     }
     if (f2 < 6000) {
      Divider2 = 150;
     }
    if (f2 < 4000) {
      Divider2 = 220;
     }
    if (f2 < 2700) {
      Divider2 = 330;
     }
    if (f2 < 1800) {
      Divider2 = 500;
     }
    if (f2 < 1500) {
      Divider2 = 600;
     }
    if (f2 < 1000) {
      Divider2 = 900;
     }
    f2=f2*Divider2;
    f2=f2*1000/25;
    f3=f2 /1000;
    f4 = f3/1000;
    f5=f3-(f4*1000);
    clockgen.setupPLL(SI5351_PLL_A, f4, f5, 1000);
    clockgen.setupMultisynth(1, SI5351_PLL_A, Divider2,
      0, 2);
    if (rdiv == 16) {
      clockgen.setupRdiv(1, SI5351_R_DIV_16);
    }
  }
}

void loop(void)
{
  unsigned long freq;
  if (Serial.available()) {
   freq = Serial.parseInt();
   setfreq (freq);
  }
}
```

be downconverted. If you want to receive a signal on 1 MHz, other signals on 3 MHz, 5 MHz, 7 MHz and so on can disturb your reception. For this reason people often use switchable lowpass filters. The SDR Shield doesn't include one of these, so it makes sense to use an antenna that's selective. Even so, things work astonishingly well with a wideband wire antenna. The reason for this is that at specific times of the day strong signals dominate on various bands and get through unscathed. An exception to this is reception on long and medium wave, which can be desensitized by signals in the short wave region. You can eliminate this problem by using a ferrite rod antenna with a rotary tuning capacitor.
The theme of antennas, filters and preamplifiers needs to be examined in closer detail. This involves not merely large signal voltages but also the achievable signal-to-noise ratio. There's nothing better than a long wire antenna, erected as far as possible from your house for this. But because this is not always possible, we must look for compromises. And in this respect the magnetic loop antenna is the clear winner. These enable you to have yourself a relatively small and unobtrusive antenna indoors. More on this later.
In your first trials with this receiver one particular question is bound to arise: won't the Arduino itself interfere with reception? It is after all in very close range. In fact great care was taken when laying out the PCB to achieve a high degree of decoupling. This includes a continuous ground plane on the underside of the PCB, whilst the 5 V and 3.3 V supply voltages are decoupled with L-C filters. In actual fact these measures are extremely effective and under normal conditions you won't notice anything from the Arduino.

Eavesdropping with the Arduino

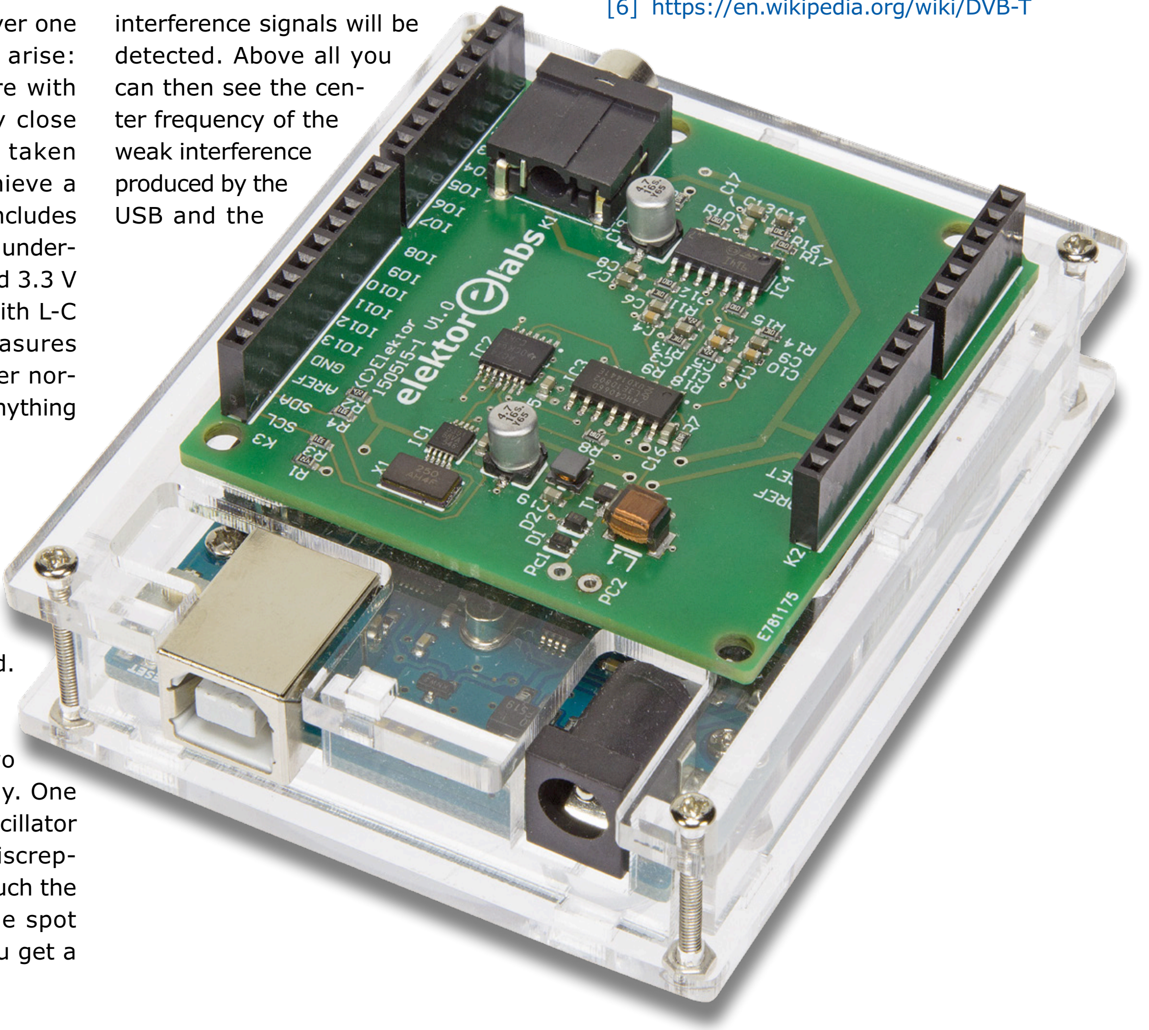
What you might at least 'receive', however, is the 16 MHz clock oscillator. This will occur when you have no antenna at all connected. The Shield can then demonstrate its ability to function as a test device. Actually there are two oscillators running simultaneously. One of these is the 16 MHz crystal oscillator on the Uno's USB chip with a discrepancy of less than 1 kHz. If you touch the underside of the Uno PCB at the spot where the crystal is soldered, you get a slight amount of detuning. You know then that it was the signal in question. With a short piece of wire on the antenna input the signal will grow stronger, as will the noise floor. And we can exploit this fact: signals that arrive via the antenna input have good image frequency suppression, whereas it's different for those that creep into the signal path via the supply voltage. The latter exhibit twice the frequency but are significantly weaker.

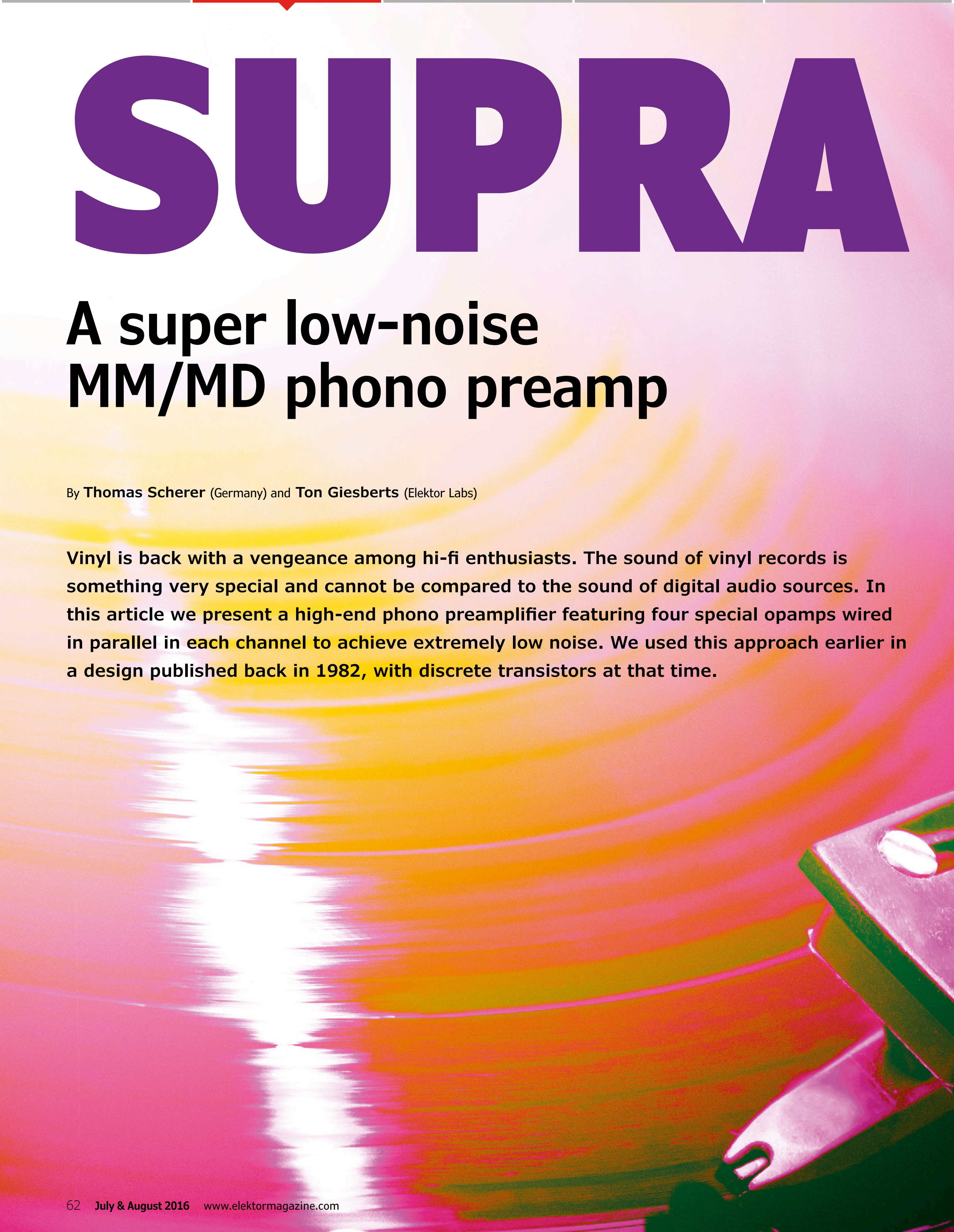
The clock signal of the Mega328 is another thing to track down. This oscillator uses a ceramic resonator and can exhibit discrepancies of up to 50 kHz. In point of fact a weak signal was found on 15950 kHz with some sideband signals into the bargain, contributed by the controller. Touching the Arduino PCB in the region of the ceramic resonator additionally set off some broad FM modulation and further detuning, which proved that the resonator was temperature-dependent to some degree. Of course it's only when you have an SDR that you can sound out the Arduino so accurately!
Without an antenna connected a SDR will normally crank up the amplification so far that even the smallest interference signals will be detected. Above all you can then see the center frequency of the weak interference produced by the USB and the Arduino. If you wanted to separate out the signals caused by the Arduino from those coming from the USB, you could simply power up the Arduino and after tuning into your desired frequency, unplug the USB cable leaving the rest running.
All internal interference signals are very weak though. As soon as you connect an antenna, the noise floor rises to the extent that all the interference is entirely masked. This shows the high sensitivity of the SDR. Even signals of only one microvolt can be received. Normally this level of sensitivity is entirely unnecessary, because the noise level of the antenna is significantly higher. Using long antennas can even lead to overmodulating the receiver and in these situations you will have to consider using an input attenuator. ⏮

(150515)

Web Links

[1] www.elektormagazine.com/070039
[2] https://github.com/adafruit/Adafruit_SI5351_Library
[3] www.elektormagazine.com/150515
[4] http://airspy.com/download
[5] http://www.hdsdr.de
[6] https://en.wikipedia.org/wiki/DVB-T





SUPRA

A super low-noise MM/MD phono preamp

By **Thomas Scherer** (Germany) and **Ton Giesberts** (Elektor Labs)

Vinyl is back with a vengeance among hi-fi enthusiasts. The sound of vinyl records is something very special and cannot be compared to the sound of digital audio sources. In this article we present a high-end phono preamplifier featuring four special opamps wired in parallel in each channel to achieve extremely low noise. We used this approach earlier in a design published back in 1982, with discrete transistors at that time.

2.0

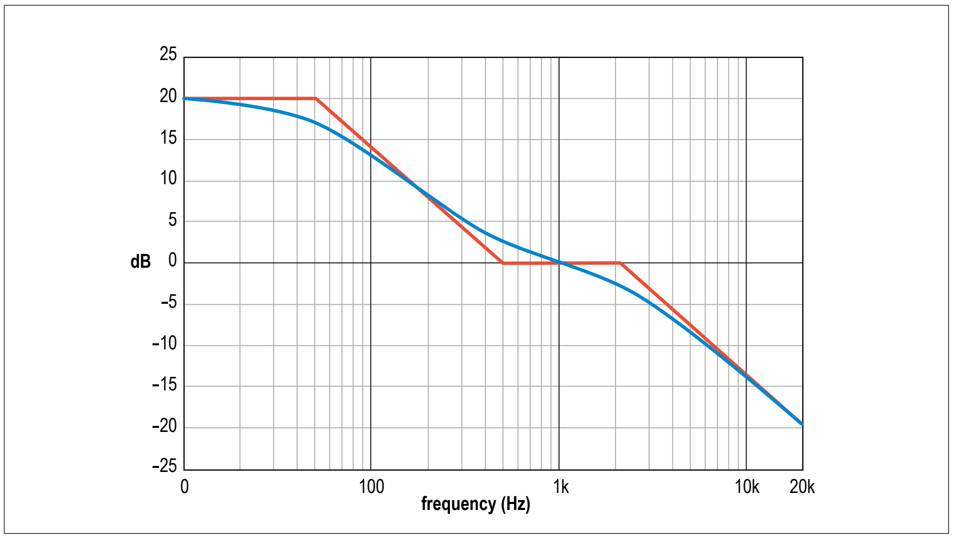


Figure 1. Theoretical and practical RIAA correction curves used for vinyl records.

Vinyl is back with a vengeance!

Audio circuits have been one of our favorite topics ever since the start of Elektor Magazine. Over the years we have developed and published innumerable designs, some of them very special and innovative. One of these was a “super low noise phono preamplifier” published in the 1982 Summer Circuits issue (SUPRA is an acronym of the Dutch name “SUPer RUISArm”, which means “super low-noise”). It had eight inexpensive low-noise transistors types BC550/BC560 connected in parallel in the input stage of each channel. This arrangement reduced the noise by a factor of √8, or 2.82. That was a respectable improvement and particularly beneficial with moving-coil (MC) pickups, which have a very low output signal level. However, the board was rather large with a total of 20 transistors per channel.

Vinyl records, along with record players and phono preamps, are now enjoying a new surge of popularity, so there are lots of hobbyists who would like to build their own phono preamp. On the Web the 1982 preamp is still a hot item and pops up regularly in various audio forums. That gave us the idea of trying to develop a modern version of it, retaining the name SUPRA. Naturally, it would have to be an opamp version of the classic design.

Nowadays there are ICs available with optimized noise characteristics, so it’s not necessary to use discrete components.

For example, the LT1028 has a noise voltage density spec of 0.9 nV/√Hz. If you connect four of these ICs in parallel, the noise is reduced by a factor of 2. That is so low that there is no point in paralleling even more opamps, since the resistors in the circuit and the internal resistance of the pickup already generate more thermal noise. These opamps sell for $5 to $10 each, so the project is not especially low-cost. However, we’re confident that real audio enthusiasts are happy to pay what it takes to get the best possible sound.
Just wiring opamps in parallel does not give you a usable phono preamp. An important part of any phono preamp is the RIAA correction network. It boosts the gain in the bass region and reduces the gain in the treble region according to a curve originally defined in 1954 by the Recording Industry Association of America (RIAA). The opposite curve is used when the record is cut, in order to avoid excessive deflection on the record at low frequencies and improve the signal to noise ratio at high frequencies. The theoretical and practical shapes of the RIAA correction curve are shown in **Figure 1**. The curve is defined by three time constants (3,180, 318 and 75 µs), which correspond to the corner frequencies of 50, 500 and 2,122 Hz.
In the past decades designers have devised all sorts of ways to implement RIAA correction in a circuit as economically as possible, ranging from fully passive to fully active and all intermediate forms. There has always been a lot of discussion about the best configuration for optimal sound quality. Here we opted for an intermediate form consisting of half-active and half-passive correction, as you can see from the circuit description.

The circuit

Despite the paralleled opamps, the schematic diagram in **Figure 2** is fairly compact. Each channel requires four input opamps and an additional dual opamp, which in this case is an LM833. The component values on the schematic are intended for MM/MD pickups with a rated output signal level of approximately 2 mV. The circuit can also be adapted for MC pickups with low output signal levels by adjusting a number of resistor values, which we plan to discuss separately in a future article.

The input impedance of the preamp is determined by R1 (47 kΩ) and C1/C1’ (in this description we only mention the components of one channel). These capacitors in combination with the connecting cable form the load capacitance seen by the pickup, and the ideal value of this capacitance depends on the actual pickup. You can adjust this yourself by mounting extra capacitors in positions C1’ and C26’ if necessary. Bear in mind that the connecting cable from the record player will have a capacitance of 100 to 200 pF. The four input opamps IC1–IC4 are not wired directly in parallel; instead each opamp has its own feedback network (R5/R4 etc.) for a gain of 48. The bandwidth is also limited by a separate network for each opamp. The LT1028 has an open-loop bandwidth (GBW) of 70 MHz. Even with a gain of 48, the closed-loop bandwidth is nearly 1.5 MHz. Since the input sensitivity is very high, it is essential to restrict the bandwidth to what is practically necessary so the preamp does not act like a radio receiver. For this reason, the bandwidth is limited to roughly 150 kHz by a 470-pF capacitor. The 47-Ω resistor in series with this capacitor keeps

the opamp stable (it is internally compensated for a gain of 2). Despite this precaution, the input stage will oscillate when the input is open and no input capacitance is present. You should therefore always terminate the input with a resistor (maximum 1 kΩ) when testing the circuit.
The opamp output signals are summed by resistors R16–R19. Together with C15/C16, they also provide the passive 75 μs correction portion of the RIAA curve, which means that any high-frequency noise present at this point is filtered out. The components are dimensioned with fairly low resistance values to keep the thermal noise of this passive network as low as possible. After all, we don't want to spoil the low-noise output of the parallel opamp input stage with noisy downstream components.

A brief remark about the capacitors used in this circuit: Ceramic capacitors are about the worst possible choice for audio circuits. Here our aim is to have the best possible sound, so we use 1% polystyrene capacitors in all key locations. They are expensive, but unquestionably the best choice for this design. For many of the capacitors there are several footprints on the PCB, so you can also use other capacitor types (MKT or MKP, for example).
The next stage, built around IC5A, provides a gain of 40 and the two remaining time constants of the RIAA correction curve at 3,180 and 318 μs, which are determined by the feedback network R20–R25/C17–C18. These two values are exactly a factor of 10 apart, so the gain drops from 40 to 4. If you want to know how the exact values for the correction networks were calculated, please see the project description on the Elektor.Labs website [1].

Next there is a buffer stage built around IC5B. This opamp has a passive low-pass filter (C19/R26) at the input with a corner frequency of 10 Hz, which suppresses any very low frequency signals from the record player drive mechanism. There is lots of room on the board for this 2.2 μF capacitor, so you can use a polypropylene type (which has better audio charac-

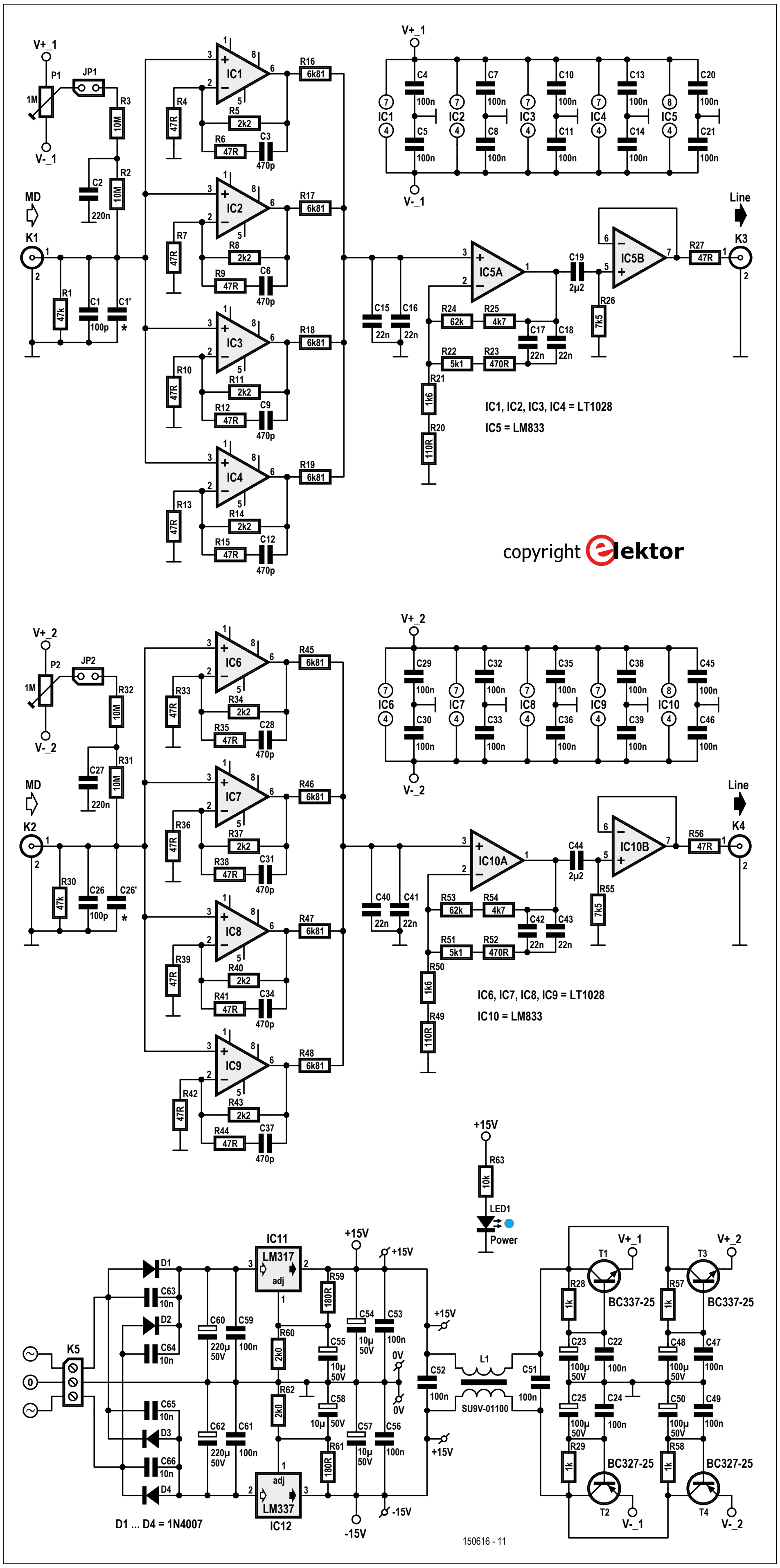


Figure 2. The schematic diagram of the low-noise preamplifier, with four paralleled opamps in the input stage of each channel.

teristics) instead of a standard polyester capacitor. Here as well, it is also possible to mount other types on the board.

That completes the basic description of the preamp circuit, but we have a bit more to say about the input stage. The LT1028 data sheet says that it has internal bias compensation, but the input bias current can still be as much as ±180 nA (or ±90 nA with the A version). With four paralleled opamps, the resulting DC current through the pickup could be as much as ±720 nA. In practice it will be a good deal less, and the input currents of the four opamps will probably largely cancel each other. However, to be on the safe side we added a simple compensation circuit for the input bias current. It consists of trimpot P1 and two high-value resistors (R2/R3) connected in series between the trimpot wiper and decoupling capacitor C2. This circuit can be enabled by fitting a jumper on JP1. If you do not consider this compensation necessary, simply omit the jumper. If you do want to use it, fit the jumper and then measure the input voltage with a high-impedance multimeter with nothing else connected to the input. Adjust P1 to minimize the input voltage. Repeat this procedure for the other channel with JP2 and P2.

Note that P1 and P2 are **not** intended to correct the output offset of the input amplifiers.

Supply voltage regulation

The power supply design is fairly conventional, with an LM317 for the positive supply voltage (+15 V) and an LM337 for the negative supply voltage (–15 V). The Adjust pin is decoupled by an electrolytic capacitor to obtain about 80 dB of ripple suppression. A full-wave bridge rectifier at the input (D1–D4) allows a variety of supply options. For example, you could use a power transformer with a single secondary (18 V / 6 W) connected to the 0 terminal and one of the ~ terminals of K5, but the resulting high ripple level from half-wave rectification makes this the least desirable option. It's better to use a power transformer with two secondary windings, such as the Block type

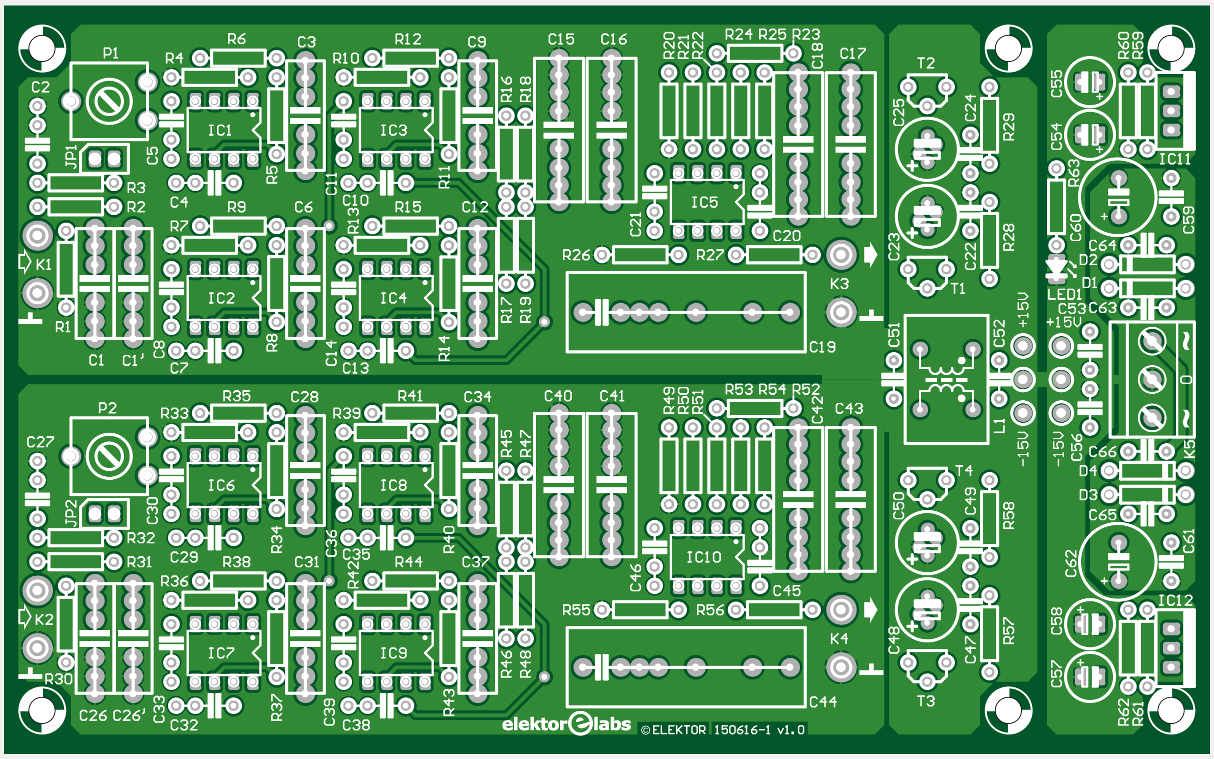
Component List

Resistors

R1,R30 = 47kΩ, 1%, 0.6 W, metal film
R2,R3,R31,R32 = 10MΩ, 5%, 0.25W, carbon film
R4,R6,R7,R9,R10,R12,R13,R15,R27,R33,R35, R36,R38,R39,R41,R42,R44,R56 = 47Ω, 1%, 0.5W, metal film
R5,R8,R11,R14,R34,R37,R40,R43 = 2.2kΩ, 1%, 0.6W, metal film
R16–R19,R45–R48 = 6.81kΩ, 1%, 0.6W, metal film
R20,R49 = 110Ω, 1%, 0.6W, metal film
R21,R50 = 1.6kΩ, 1%, 0.6W, metal film
R22,R51 = 5.1kΩ, 1%, 0.6W, metal film
R23,R52 = 470Ω, 1%, 0.6W, metal film
R24,R53 = 62kΩ, 1%, 0.6W, metal film
R25,R54 = 4.7kΩ, 1%, 0.6W, metal film
R26,R55 = 7.5kΩ, 1%, 0.6W, metal film
R28,R29,R57,R58 = 1kΩ, 1%, 0.6W, metal film
R59,R61 = 180Ω, 1%, 0.6W, metal film
R60,R62 = 2.0kΩ, 1%, 0.6W, metal film
R63 = 10kΩ, 5%, 0.25W, carbon film
P1,P2 = 1MΩ, 20%, 0.15W trimpot, vertical mounting

Capacitors

C1',C26' = nor fitted, see text
C1,C26 = 100pF 160V, 2.5%, axial, polystyrene, max. 12.9 x 5mm
(match C1 and C26 to element in use)
C2,C27 = 220nF 100V, 10%, lead pitch 5 or 7.5mm
C3,C6,C9,C12,C28,C31,C34,C37 = 470pF 160V, 2,5%, axial, polystyrene, max. 12.9 x 5mm, lead pitch 5, 7.5, 10, or 14.6mm
C4,C5,C7,C8,C10,C11,C13,C14,C20,C21,C29 ,C30,C32,C33,C35,C36,C38,C39,C45,C46 = 100nF 50V, 10%, X7R, 0.2'' or 0.34'' lead pitch
C15–C18,C40–C43 = 22nF 63V, 1%, axial, polystyrene, max. 17 x 6,5mm, lead pitch 5, 7.5, 10, 14.6, or 19mm
C19,C44 = 2.1µF 420 V, 10%, polypropylene, lead pitch 5, 7.5, 10, 15, 22.5, or 27.5mm
C22,C24,C47,C49,C51,C52,C53,C56,C59,C61 = 100nF 50V, 10%, X7R, 0.2'' lead pitch
C23,C25,C48,C50 = 100µF 50 V, 20%, diam. max. 8mm, lead pitch 2.5 or 3.5mm



FL6/18. That transformer also has two primary windings, making it suitable for regions with an AC line voltage of 115 V nominal. Another option is to use a separate balanced DC power supply or a small switching power supply, as described further on in this article. In that case the positive and negative supply leads are connected to the two ~ inputs of K5. Thanks to the diodes it does not matter which supply lead is connected to which terminal, as long as the neutral lead is connected correctly.

The output voltages from the two regulators pass through filter F1, which effectively suppresses common-mode interference. It is followed by four low-pass filters built around T1–T4, with a long time constant of 0.1 s (R28/C23 etc.). To ensure that the supply voltages are absolutely clean, the positive and negative supply voltages for each channel are filtered individually by this arrangement.

Power supply options

As previously mentioned, you can choose between linear and switching power supplies.

Linear power supply

For those of you who would rather not use a DC/DC converter to power a phono preamp, we designed a separate power supply board (see the schematic in **Figure 4** and the PCB layout in **Figure 5**) with room for a PCB-mounted transformer from the Block FL series, along with some fuses and terminal blocks. We merged several footprints in the transformer position to allow transformers with different power ratings to be mounted on the board. The type shown in the components list (FL6/18) is more than sufficient for the phono preamp.

This PCB is intended for general purpose use, so there are white label areas next to the fuses where you can mark the actual fuse values. An EMI filter is included on the AC line side to suppress interference as much as possible at this point. The filter consists of a common-mode choke (L1) and two X1 capacitors (C1 and C2).

C54,C55,C57,C58 = 10µF/50V, 20%, diam. 6.3mm max., 0.1'' lead pitch
C60,C62 = 220µF 50V, 20%, 10mm diam. max., 0.2'' lead pitch
C63–C66 = 10nF 50V, 20%, Y5V, 0.2'' lead pitch

Inductor

L1 = SU9V-01100, common-mode-choke 2x10mH, 100mA

Semiconductors

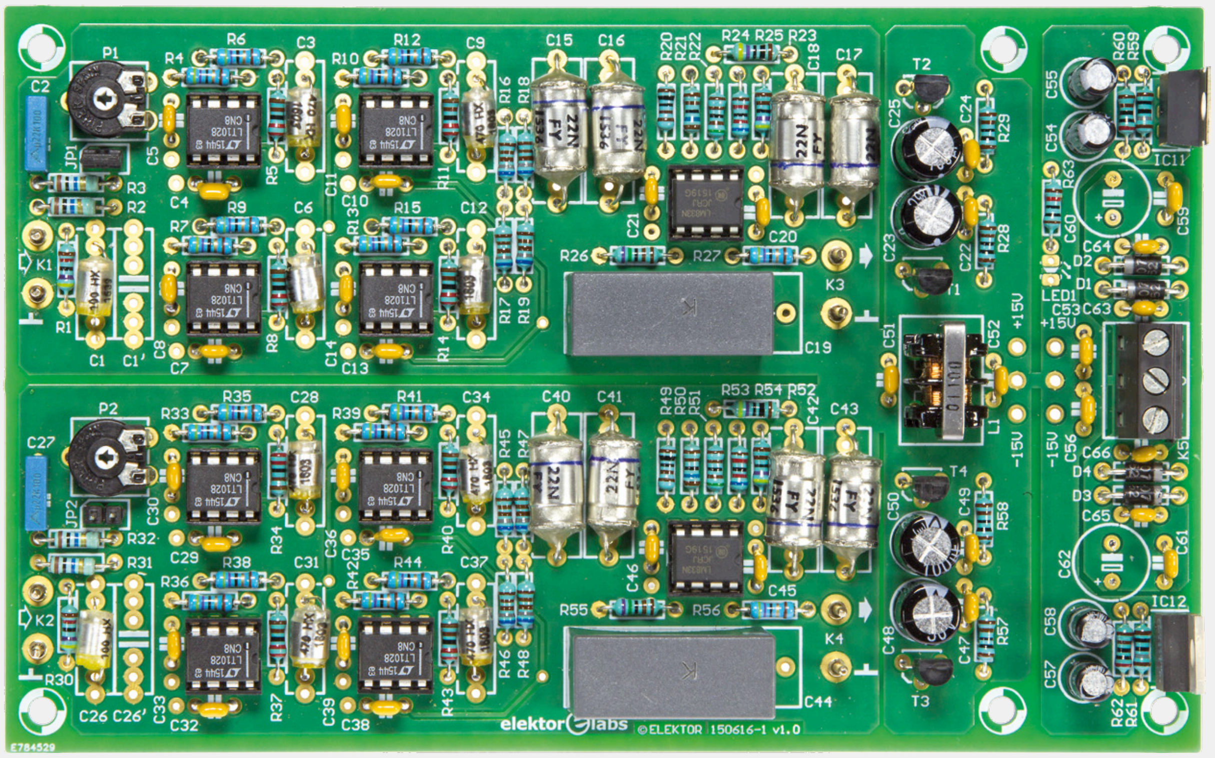
D1,D2,D3,D4 = 1N4007, DO-41
LED1 = high-intensity LED, blue, 3mm, T-1
T1,T3 = BC337-25, TO-92 case
T2,T4 = BC327-25, TO-92 case
IC1..IC4,IC6..IC9 = LT1028CN8, DIP-8 case
IC5,IC10 = LM833NG, DIP-8 case
IC11 = LM317, TO-220 case
IC12 = LM337, TO220 case

Miscellaneous

K1–K4 = solder pin, 1.3mm diam.
K5 = 3-way PCB screw terminal, 0.2'' lead pitch
JP1,JP2 = 2-pin pinheader, 0.1'' pitch
JP1,JP2 = jumper for pinheader
Enclosure, Hammond type 1455N1602
2 pcs Neutrik NYS367-0 cinch socket for panel mounting, black
2 pcs Neutrik NYS367-2 cinch socket for panel mounting, red

PCB # 150616-1

Figure 3. The PCB layout for the preamplifier is generously dimensioned, and all components are leaded (through-hole) types.



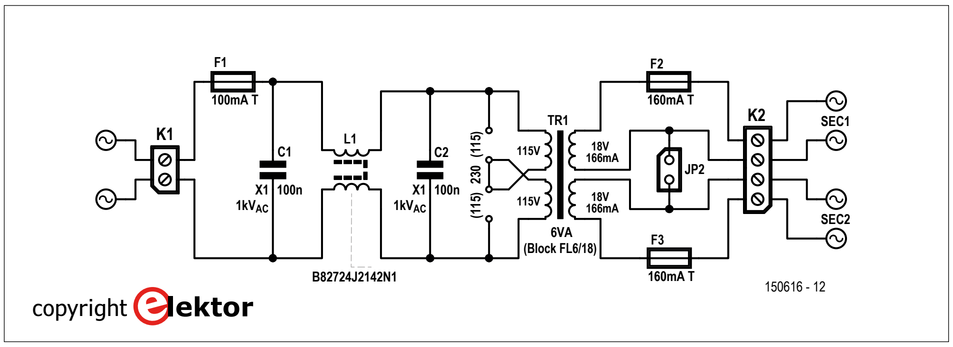


Figure 4. If you want to keep everything linear, you can use a conventional power transformer for the power supply. There's also room for fuses and interference suppression components (choke and capacitors).

The transformer has dual primary and secondary windings. For an AC line voltage of 230 V, fit a single jumper at JP1; for 115 V fit two jumpers. The secondary windings are connected to terminal block K2. These two windings can be connected in series by fitting a jumper on JP2, allowing the board to be used for a wide variety of applications.

We recommend installing the linear supply in a separate enclosure, so that the power transformer can be kept reasonably far away from the sensitive preamp circuit. If you use the same enclosure type for both units, they will make an attractive matched pair. There are various options for interconnecting the two enclosures, either with connectors or hard-wired, and we leave that to your discretion.

Switching power supply

Another option is a compact switching power supply small enough to fit in the same enclosure as the preamp board. For this we took the modular power supply described in the "Brick-by-Brick Power Supply" article in this edition — a small circuit that is suitable for various types of DC/DC converter (**Figure 6**). The JCE0612D24 module from XP Power is a good choice for this project. It delivers ±24 V at 6 W, with an input voltage range of 9 to 18 V. The output voltage is higher

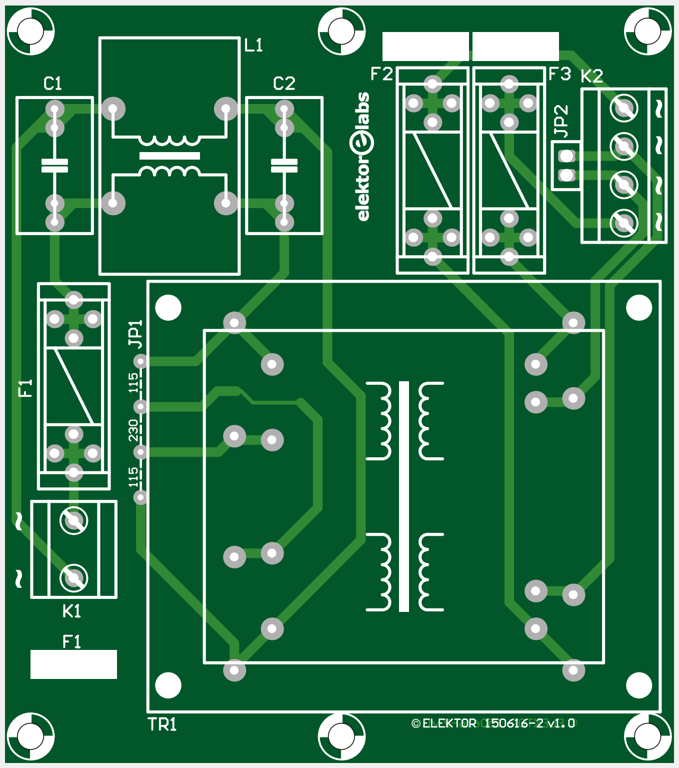
Component List Analog Power Supply

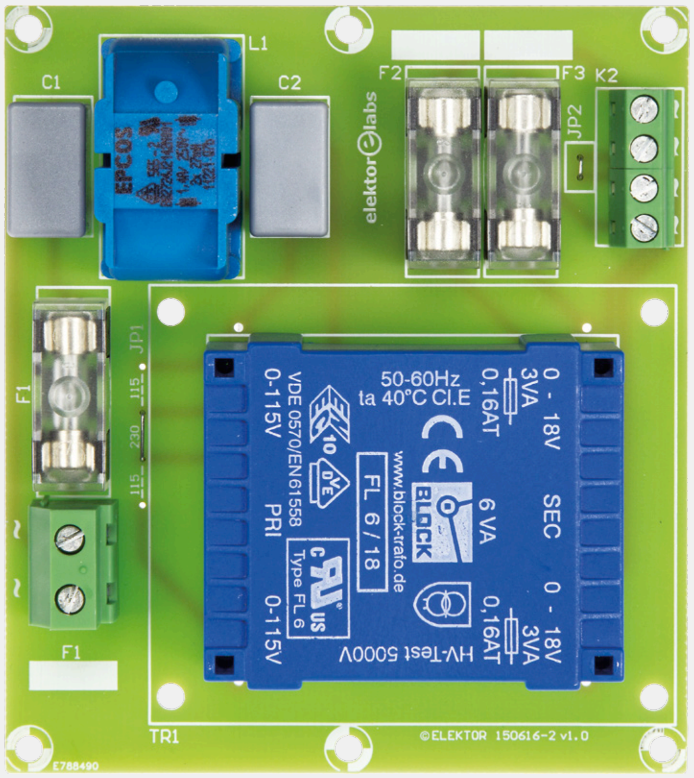
Capacitors
C1,C2 = 100nF 1kV, X1, polypropylene, 10, 12.5, or 15mm lead pitch

Inductor
L1 = B82724J2142N1 common-mode choke, 2x27mH, 1.4A

Miscellaneous
F1,F2,F3 = fuseholder for PCB mounting, 20x5mm, 500V, 10A
F1,F2,F3 = cap for fuseholder, 20x5mm
F1 = 100mAT (slow-blow) glass fuse, 20x5mm
F2,F3 = 160mAT (slow-blow) glass fuse, 20x5mm
K1 = 2-way PCB screw terminal block, 0.3'' pitch, 500V
K2 = 4-way (2x2) PCB screw terminal block, 0.2'' pitch
TR1 = 6VA power transformer, 2x115V primary, 2x18V/166mA secondary, e.g. Block FL6/18
JP1,JP2 = wire link, see text
PCB # 150616-2

Figure 5. The PCB for the linear power supply is designed with a general-purpose layout and can also be used for other applications.





than necessary for this application, but we did not find any currently available modules with balanced output voltages between 15 and 24 V.

If you install this power supply board in the same enclosure as the preamp, you can simply mount a standard power connector to mate with an AC adapter (12 V_{DC} at 0.5 A or more) on the rear of the enclosure.

Note: Capacitors C60 and C62 on the preamp board must be removed if the modular power supply board is used because the maximum rated capacitive load of the DC/DC converter module is 47 µF per output. Also note that the following components on the power supply board are omitted because the auxiliary supply voltage is not needed for this project: K1 (replaced by a wire bridge as shown in the photo of the assembled board in **Figure 7**), C7, C8, C9, D2, D3, MOD1 and K3

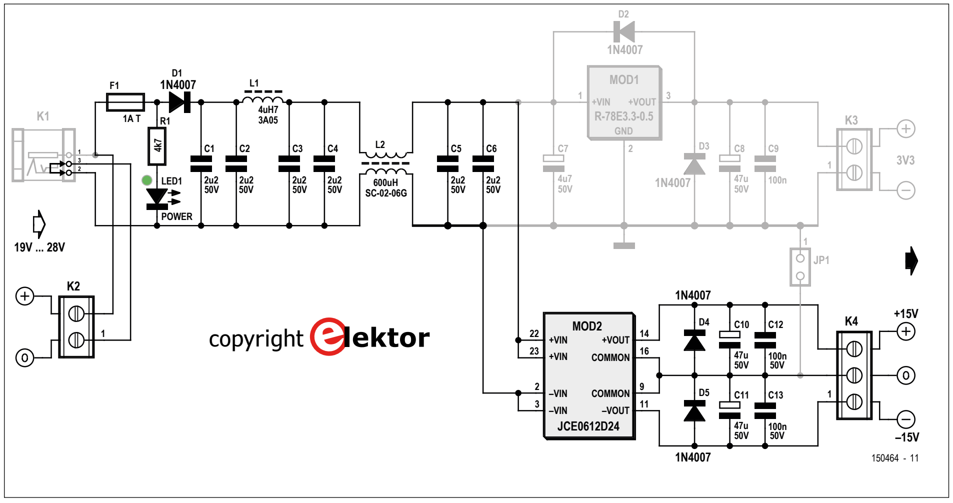


Figure 6. You can also use a small switching power supply, such as the “Brick-by-Brick Power Supply” described elsewhere in this issue. It can be mounted in the same enclosure as the main board (see the components list). The complete schematic diagram of this circuit is reprinted here.

Construction and packaging

Before you start building the preamp, you have to decide which configuration you want to make. In our prototype we opted for the combination of the preamp board and the modular power supply board. They both fit nicely in the Hammond enclosure specified in the components list.

Board assembly should not present any problems even if you don’t have advanced soldering skills, but you should read the article carefully and completely before you start. The input capacitors can be adapted to the pickup you will be using. The manufacturer usually states the preferred value. The opamps may be mounted in suitable high-quality sockets if you want to be able to try different types later on. Some of the components are omitted on the modular power supply board; only the parts shown in the components list are necessary (see also the photo of the assembled board).

The preamp board fits precisely in the bottom slots of the Hammond enclosure. Mount the Cinch sockets and the ground terminal on one of the end panels as shown in **Figure 8**. We used insulated Cinch sockets from Neutrik (see the components list). To avoid ground loops, mount the Cinch sockets with the ground

Component List Modulare Supply

Resistors
R1 = 4.7kΩ, 5%, 0.25W

Capacitors
C1–C6 = 2.2µF 50V, 20%, ceramic Y5V, 0.2’’ pitch
C10,C11 = 47µF 50V, 20%, 2.5mm pitch, diam. 6.3mm max.
C12,C13 = 100nF 50V, 10%, ceramic X7R, 0.2’’ pitch

Inductors
L1 = 4.7µH, 3.05A, 80mΩ, 10%, radial, 5mm pitch, e.g. Epcos B82144B2472K000
L2 = 600µH, 2A, 2x50mΩ, common-mode choke, 17.5x14mm, e.g. Kemet SC-02-06G

Semiconductors
D1,D4,D5 = 1N4007
LED1 = LED, green, 3mm
MOD2 = JCE0612D24 XP Power (6W ±24V)

Miscellaneous
K2 = 2-way PCB screw terminal block, 0.2’’ pitch
K4 = 3-way PCB screw terminal block, 0.2’’ pitch
F1 = fuseholder for PCB mounting, 20x5mm, 500V/10A
F1 = cap for fuseholder, 20x5 mm
F1 = 1 AT (slow-blow) glass fuse, 20x5mm
2 wire links!
PCB #150464-1

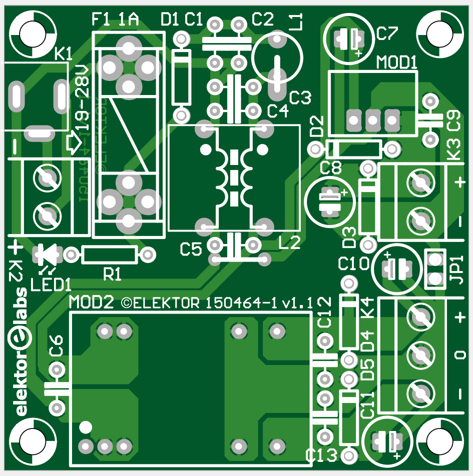
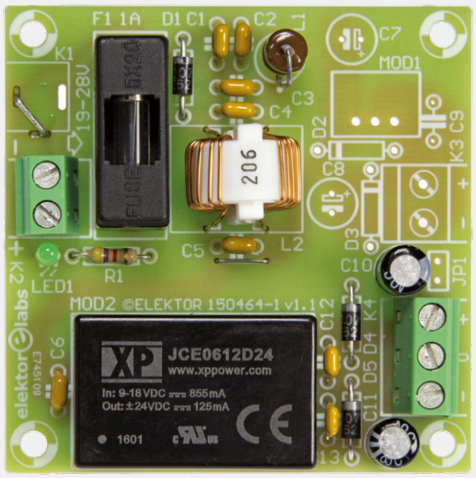



Figure 7. The PCB for the switching power supply. Note that some components can be omitted because the auxiliary supply voltage from this board is not needed.

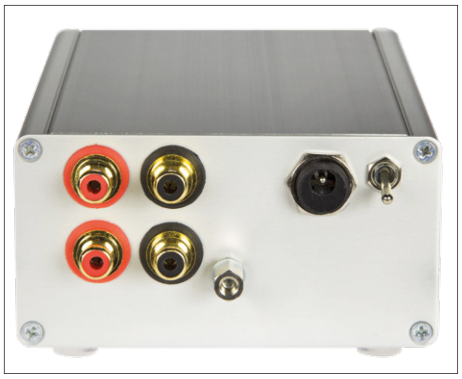


Figure 8. The layout of the connectors and switch on the rear panel.

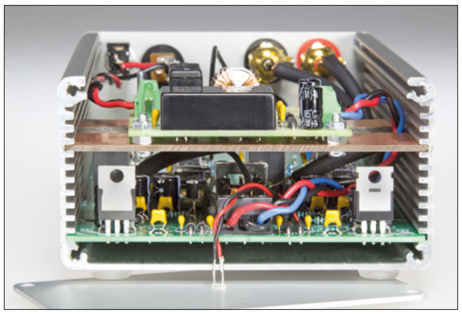


Figure 9. The preamp board and power supply board mounted one above the other in the enclosure.

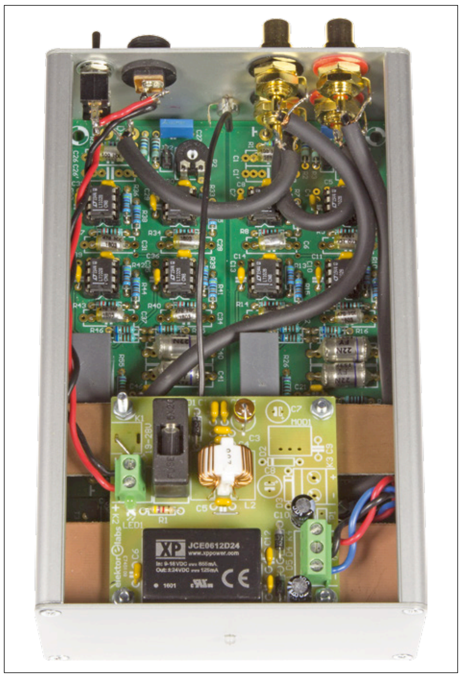


Figure 10: Connecting the wiring is a bit tricky, but there's just enough room.

terminals isolated from the mounting panel. Also make sure that the sockets do not touch any of the components on the board inside the enclosure (there's not much extra space). Position the connectors far enough away from the top edge to allow the top panel to fit properly. We used a small 4-mm screw and nut as the terminal for the ground wire from the record player, with a short internally threaded metal standoff threaded onto the screw. Inside the enclosure the ground screw is connected to the ground pad (0) on the preamp board by a solder lug and a short piece of wire. There is just enough room above the top right corner of the circuit board for the power connector and an on/off switch. On the front panel you can mount a preamp power indicator lamp (LED).

Then you can connect the inputs and output connectors to the board using good-quality shielded cable. We recommend using matched cable lengths for the left and right channels.

You can mount the modular power supply board on a piece of blank PCB material with the same width as the preamp board (dimensions 100 x 62 mm). Then it can be slide into the enclosure right above the preamp board. The photo in **Figure 9** shows how the two boards fit in the enclosure. Now you have to connect the inputs of the power supply board to the on/off switch and the power connector, and the outputs of the power supply board to the supply voltage terminals of the preamp (K2 on the power supply board to K5 on the preamp board). Then connect the LED on the front panel to the corresponding pins on the preamp board with two thin stranded wires.

When you're done with all that, it's time to inspect all the connections and test the circuit. Connect a suitable 12-V AC adapter and check the following points with a multimeter:

- ±24 V on connector K4 of the modular power supply board. Is the green LED lit?
- There should be ±15 V on the preamp board next to C53/C56.
- Check the supply voltages of all opamps; they should be approximately ±14 V.
- Terminate the inputs with 560 Ω resistors and measure the input offset voltages on R1 and R30; they should be 0.0 mV. Depending on the results, you can decide whether or not you need to use the compensation trimpots P1 and P2.
- Measure the output offset voltages of IC1–IC4 and IC6–IC9; they should not exceed a few tens of millivolts.
- Check the output offset voltages of IC5 and IC10; they should not exceed a few millivolts.

If everything is okay, screw the enclosure together. Connect the phono preamp between your record player and your control preamp. Switch on the preamp and allow it to warm up for a few minutes. Now you can sit back and enjoy your favorite LPs with top quality. Happy listening! ◀

(150616)

Web Link

www.elektormagazine.com/labs/supra-20-high-end-preamp-for-record-player-150616-i

Specifications and measurements

- Measurements were made with an input signal level of 2.5 mV (source impedance 560 Ω)
- (1) Measurements made with linear power supply (150616-2);
  (2) Measurements made with switching power supply (150464-1)

SNR

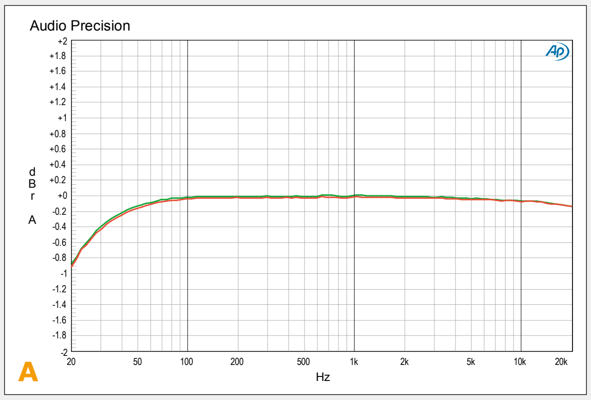
• Lineair:	> 68 dB (1)
	> 70 dB (2)
- Input shorted:	> 88 dB
• A-weighted:	> 78 dBA
- Input shorted:	> 95 dBA
• Output level with 2.5 mV input:	468 mV
• THD:	< 0.001 % (BW 20 kHz)
• Deviation from RIAA curve:	<0.1 dB (100 Hz to 10 kHz)
	< 1 dB (20 Hz)
	< 0.15 dB (20 kHz)

We also recorded several measurement curves. The most interesting of these is the deviation from the theoretical RIAA curve.

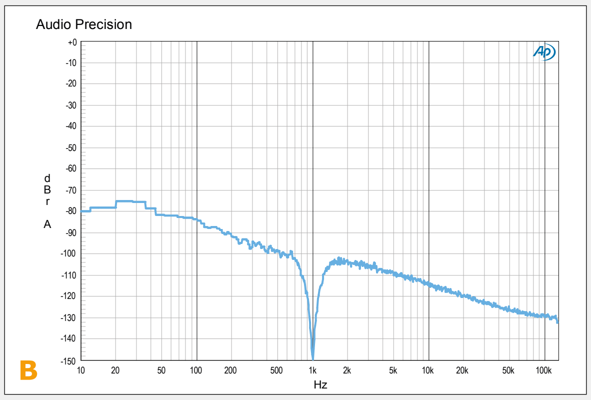
Chart A shows the output amplitude with the input driven by a sinusoidal signal corrected according to the RIAA standard. The deviation at 20 Hz is approximately ±0.9 dB. At 20 kHz it is ±0.1 dB. To give you an idea of the harmonic distortion, we performed an FFT analysis at a frequency of 1 kHz.

Chart B shows the spectrum from 10 Hz to 130 kHz with the fundamental frequency at 1 kHz suppressed. There are no harmonics at all visible in this figure, which was obtained by averaging 16 measurements for better clarity. From this we can conclude that the THD is less than 0.001%. We also performed an FFT analysis with a 10 kHz signal to see how the preamplifier behaves in the treble range.

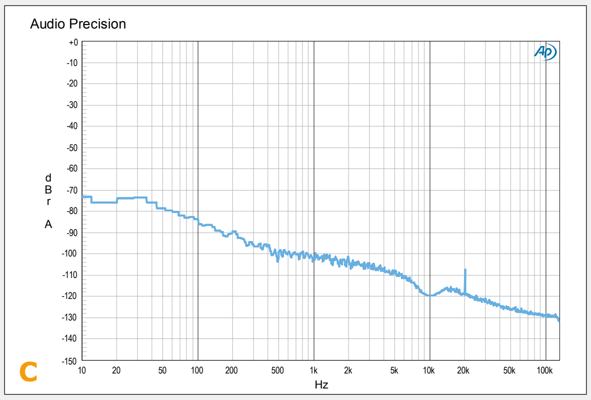
The only thing visible in **Chart C** is the second harmonic at a level of −107 dB, corresponding to 0.00045%.



A



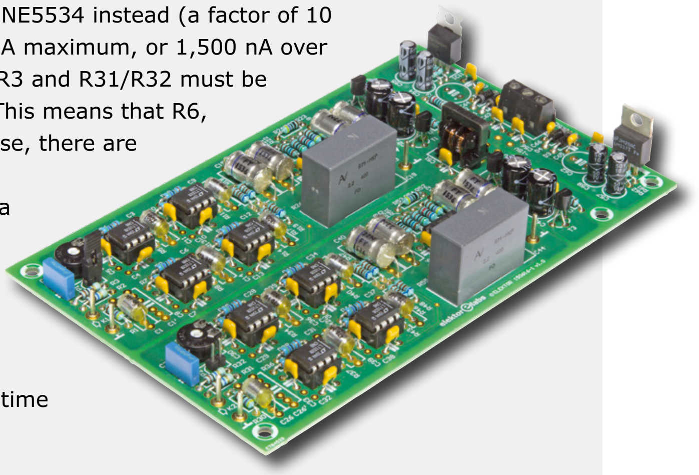
B



C

Opamp options

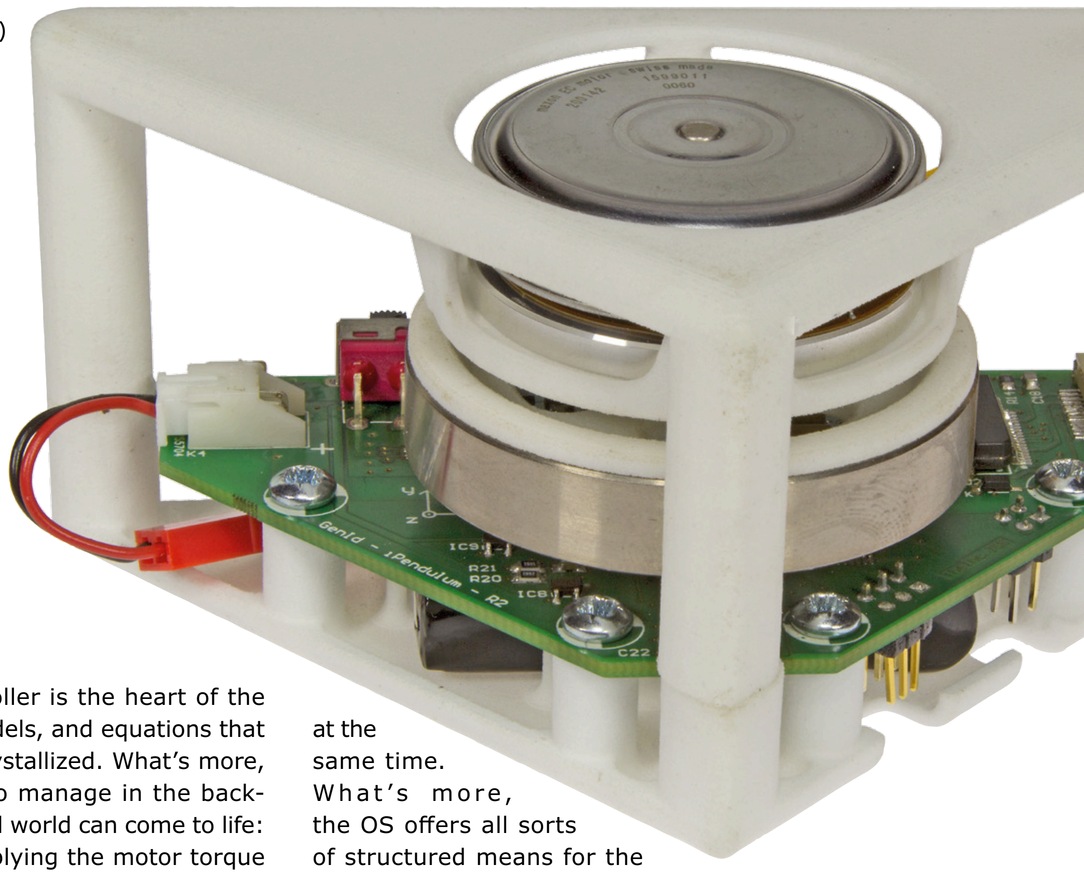
The LT1028 is an expensive opamp, and there are eight of them on the preamp board. If you find them too expensive and want to use lower-cost opamps, you can use the old faithful NE5534 instead (a factor of 10 cheaper). However, the bias current of the NE5534 is much higher — 800 nA maximum, or 1,500 nA over the full specified temperature range. If you go this route, the values of R2/R3 and R31/R32 must be reduced to 2.2 MΩ. The NE5534 is internally compensated for a gain of 3. This means that R6, R9, R12, R15, R35, R38, R41 and R44 must be increased to 100 Ω. Of course, there are many other pin-compatible opamps that you could try out.
We also looked for alternatives to the LT1028, since it has been around for a long time and there are probably even better devices available now. One interesting candidate is the LME49990. However, it is only available in the SO8 package. To allow this device to be used with a DIP socket, we designed an SO8 to DIP8 adapter board measuring 10 x 10mm, which is suitable for single and dual opamps. There is also room for a decoupling capacitor on the adapter board. We plan to publish the adapter board sometime soon, along with a full description.



i-Pendulum

Part 2 – software, assembly, control

By **Jean-Sébastien Gonsette** (Belgium)



After the theory, here comes the practical! This second part will guide you in building your inverted pendulum. Once the electronics have been wired up and the mechanical parts assembled, you'll be able to listen to the heart of the pendulum using the Windows software designed for this purpose.

Software

The firmware in the PIC microcontroller is the heart of the project — this is where the rules, models, and equations that enable the pendulum to move are crystallized. What's more, there are all sorts of mechanisms to manage in the background so that this entire little internal world can come to life: communicating with the sensors, applying the motor torque drives, communicating with the PC via a UART for sending the telemetry data, etc. The control loop itself imposes certain constraints. The digitizing time step is of the order of around 10 ms. Even though the calculation load is not very great, the software needs to react quickly. Also, all of the software that is not within this control loop, like the UART communication, must be able to operate in the face of regular interruptions. Writing a program like this for a bare processor is not easy. The elementary approach – to have one main super-loop that calls all these modules in turn – is only possible if there are not too many of them. What's more, this soon becomes a real headache when they all have periods that differ by orders of magnitude. Typically, if a fast processing loop has to co-exist with a slower calculation loop in the background, we need to think about how to reconcile them both: co-operative time sharing, use of interrupts, etc. If on top of this you want to take advantage of the processor's power-saving mechanisms (going into stand-by when it has nothing to do), it's a real nightmare. In all cases, it's only viable on a small scale, as it amounts to doing the work of an operating system (OS).

A Real-Time Operating System (RTOS), however, offers the advantage of being able to organize its code in a more modular way. Each part of the application can run independently of the others in its own task; the OS takes care of the time sharing in the background to give the illusion that all the tasks are running at the same time. What's more, the OS offers all sorts of structured means for the tasks to communicate with each other (while avoiding horrible global variables). Compared to a conventional OS, an RTOS is light and responsive. And as it doesn't require much memory, it can even be used on small microcontrollers. For all these reasons, the pendulum firmware runs on a custom RTOS named *NanoScale* (more details from [1]).

Architecture

Figure 1 summarizes the architecture of the pendulum's onboard application, organized as three subsystems:

1. Managing ***communication*** with the outside world via a serial interface. One task receives commands, while a second collects all the transmission requests from the application and sends the corresponding messages one by one. The send task performs time-division multiplexing on the messages to be sent. These are either replies to incoming commands or telemetry data sent spontaneously.
2. The ***control*** is headed up by a high priority task, activated every 10 ms. Each time this is activated, it has to acquire the sensor values, estimate the corresponding states of the pendulum, then derive the motor torque to be applied in accordance with the current algorithm. There is an algorithm for each of the pendulum's elementary functions: jump, static balance, balancing around the balance point, etc.

3. And lastly, the ***management*** subsystem takes care of the general tasks: activating/deactivating the *control*, checking the state of charge of the battery, or making the LEDs flash.

Microcontroller iinitializing and abstraction
The Board Support Package (**BSP.c**) module initializes the controller. First of all, it configures the internal clock frequency, along with all the I/Os necessary for connecting to the IC's various peripherals. Then four peripheral drivers are initialized: SPI bus, UART, ADXL345 accelerometer driver, and ISZ2510 gyroscope driver. The latter two use the SPI bus to communicate with their electronic counterpart on the PCB. And last of all, the RTOS is initialized.
The BSP module also offers an abstraction layer that makes it possible to isolate the microcontroller from the rest of the application. Thus it contains all the routines for handling interrupts, in particular INT2 connected to the motor controller tachometer output pin. This gives one pulse every eighth of a revolution of the motor, making it possible to deduce its speed. And lastly, the abstraction is based on a set of specific APIs. The module offers low-level functions to drive the LEDs, simulate an EEPROM in the internal flash memory, read the battery voltage, and drive the motor controller.

External interfaces
The **battery.c** module is simple: reading the battery voltage via one of the microcontroller's analog inputs and correcting this raw value using a calibration factor. In addition, it estimates the remaining battery charge, depending on the voltage and the current being drawn.
The **motor.c** module offers interfaces for sending the drive to the motor via the L6235 (IC3). In concrete terms, IC3 regulates the current flowing through the motor windings depending on a setpoint value set by an analog signal Vref. Since the torque of a brushless motor is proportional to the current through it, in theory the regulation sets a torque setpoint and IC3 is tasked with applying it. However, there are two snags: this IC does not work in four quadrant mode. It only regulates the torque when this is in the same sense as the rotation of the wheel.

What's more, it is not able to work with low torque setpoints. An original technique for controlling the L6235 IC makes it possible to solve the first problem: sending a PWM signal to the FWD pin (which is used to select the motor rotation direction). If a PWM signal is applied, the H bridge that drives the motor switches over very fast, thereby modulating the voltage applied to the motor in proportion to the chosen duty cycle. Refer to document [2] for more details of this technique. In this way, I was able to empirically find a combination of signals that would make it possible to regulate a negative torque, i.e. torque to brake the wheel.

The second problem is trickier and is still not satisfactorily resolved. For the moment, I am contenting myself with bypassing the L6235 regulation when low torque values are required. At that point, I use the voltage modulation technique explained

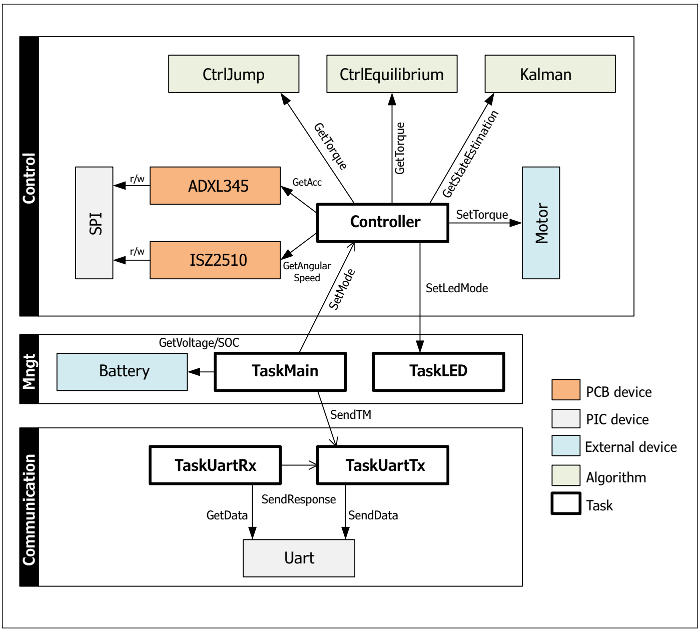
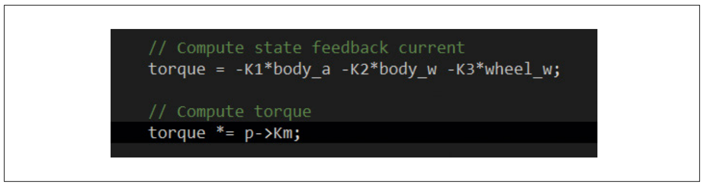


Figure 1. Software architecture.

above, basing myself on the motor's theoretical operating curve. In this way I obtain an average motor torque for a regulation period *dt*. The result isn't quite right, because there's a discontinuity when the rotation direction of the motor is reversed. For the moment, that prevents regulating the pendulum around zero wheel speed.

Sensor driver
The ADXL345 sensor driver (IC4) is found in the eponymous module **ADXL345.c**. This is an API that reflects the functions of the sensor: selecting the data transmission speed, "power saving" mode, etc. The driver communicates with the sensor by transmitting appropriate SPI frames. The primary function is to recover the most recent acceleration values (in m*g*) read in the three axes: `ADXL345_GetAccelerations`.
The ISZ2510 sensor driver (IC6) functions in a similar way and is found in the **ISZ2510.c** module. Its primary function is likewise to recover the most recent angular speed value (in thousandths of a °/s): `ISZ2510_GetAngularSpeed`.

Task distribution
The microcontroller's work is divided up into various tasks with differing degrees of criticality. The OS governs these tasks, constantly switching execution to the active task with the highest priority. The lowest priority task is obviously the one that takes care of driving the three LEDs: **TaskLED.c**. Each LED is associated with a little finite-state automaton that makes them flash separately or in a synchronized way. I'll come back to this later. Then in order of priority comes the management task **Task-Main.c**. At start-up, this ensures that all the pendulum's internal functions are in working order: motor control and correct response from the sensors. If everything is OK, it activates the control task and then goes into wait for each time its cycle ends. As soon as the end of a cycle is signaled, every 10 ms, it collects all the data of interest and sends them in telemetry frames via the UART. This data is made up of the values



```
// Compute state feedback current
torque = -K1*body_a -K2*body_w -K3*wheel_w;

// Compute torque
torque *= p->Km;
```

Figure 2. The pendulum's balance is based on these simple lines of code.

returned by the sensors, the state estimated by the Kalman filter, and lastly the setpoints sent to the motor for regulation. In addition to this, the task also verifies every minute that the battery voltage has not dropped below an acceptable level.

The **TaskUartTx.c** task sends all the data (messages) from the pendulum to the UART: responses to commands or telemetry data. A module with a message to send adds it into the FIFO queue; this task unstacks the messages one by one so they can be sent over the UART.

The **TaskUartRx.c** task handles all the incoming commands in the UART that make it possible to obtain information about the state of the system and are not sent automatically in the telemetry data: state of the OS, battery state, etc.
The most critical task is the one that takes care of regulating the pendulum every 10 ms: **Controller.c**. A regulation cycle comprises the following steps:

1. acquisition of sensor values (accelerometer and gyroscope)
2. estimating the state of the pendulum depending on these values via the Kalman filter (implemented in KalmanFree.c)
3. calculating the torque drive to be applied to the motor in accordance with the movement being performed (CtrlEquilibrium.c or CtrlJump.c)
4. and lastly, applying the motor drive.

The heart of the regulation (**Figure 2**) comes down to two lines of code that calculate the torque to be delivered to the motor at each moment in order to maintain the pendulum's balance. The rest is only detail – but it's important not to lose sight of the fact that "the devil is in the details".

Assembling the pendulum

The restricted dimensions of the pendulum mean that the size of the PCB is a major constraint. Hence it has SMD components

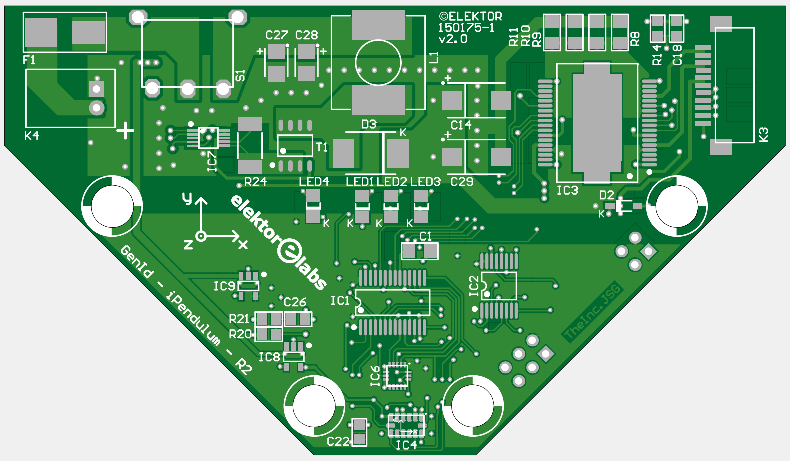
Component List

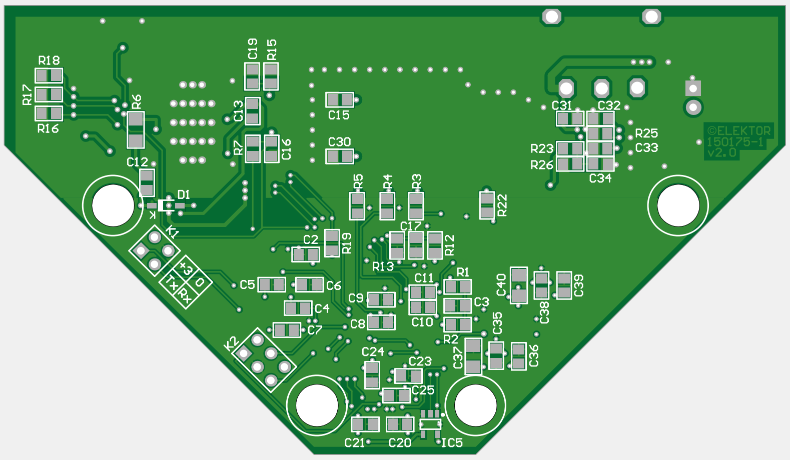
Resistors
Default: 0805, 0.125W, 1%
R1,R21 = 10kΩ
R2 = 330Ω
R3,R4,R5,R22 = 130Ω
R6 = 100Ω, 1206, 0.25W, 1%
R7 = 100kΩ
R8,R9,R10,R11 = 1 Ω, 1206, 0.25W, 1%
R12 = 56kΩ
R13 = 11kΩ
R14,R20 = 20kΩ
R15,R26 = 24.9kΩ
R16,R17,R18 = 4.7kΩ
R19 = 1kΩ
R23 = 215kΩ
R24 = 15mΩ, 2512, 1W, 1%
R25 = 160kΩ

Capacitors
Default: 0805, 50V, 10%
C1,C37,C40 = 10µF 16V, 1206
C2,C3,C4,C5,C6,C7,C8,C10,C21,C22,C23,C25,C26,C30,C31 = 100nF
C9,C11,C12,C17,C24,C36,C39 = 10nF
C13,C15,C33 = 220nF
C14,C29 = 68µF 16V, 20%, 0.05Ω, SMD-D
C16 = 5.6nF
C18 = 1.2nF
C19 = 33nF
C20,C32,C35,C38 = 1µF
C27,C28 = 47µF 10V, 20%, 0.035Ω, SMD-D
C34 = 100pF

Inductor
L1 = 5.0µH shielded high power inductor

Semiconductors
D1, D2 = 1N4148W
D3 = MBRS340T3G
LED1 = SMD, 1206, red
LED2 = SMD, 1206, yellow
LED3, LED4 = SMD, 1206, green





IC1 = dsPIC33EP128MC202-I/SS, programmed (Microchip sampling service)
IC2 = SN65C3221E
IC3 = L6235PD
IC4 = ADXL345BCCZ
IC5 = SN74AHC1G32DBVT

fitted on both sides. The second concern comes from the two sensors: the accelerometer and the gyroscope. These types of devices only exist in LGA (Land Grid Array) packages, so they don't have legs you can solder using a soldering iron: a reflow oven is the only way to do it!

Connector K1 makes it possible to load the firmware (available here [3]) into the microcontroller. The source code is also available and can be recompiled using the MPLAB X environment from Microchip.
Figure 3 shows the various "layers" of the assembled pendulum.

Testing the PCB
Test the loaded PCB before moving on to the mechanics. Power it with a voltage of 6–9 V on K4, or connect up the LiPo battery. LED4 must light to indicate power is present. The program then checks that it is able to communicate with the accelerometer and gyroscope.

Then connect the motor up to K3. Hold the motor's stator and apply power. The motor will turn briefly and, if everything is alright, LED1 will not flash.

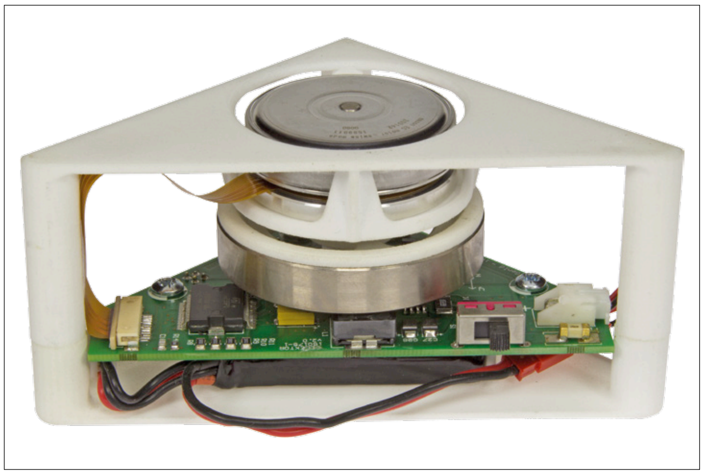
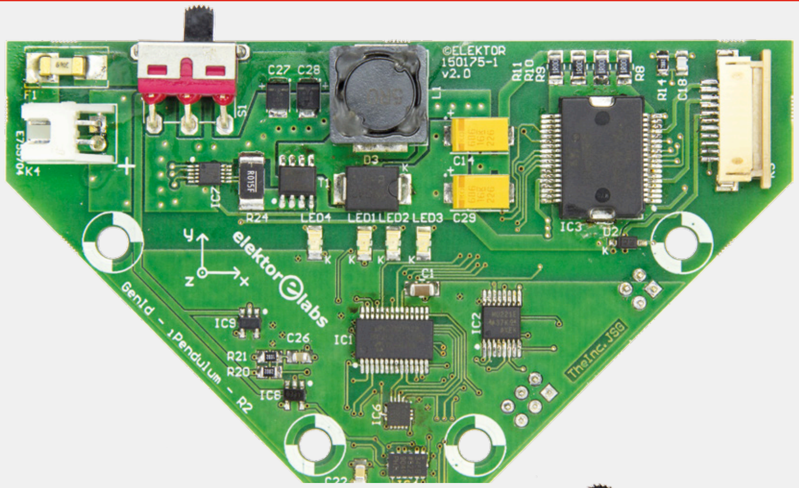
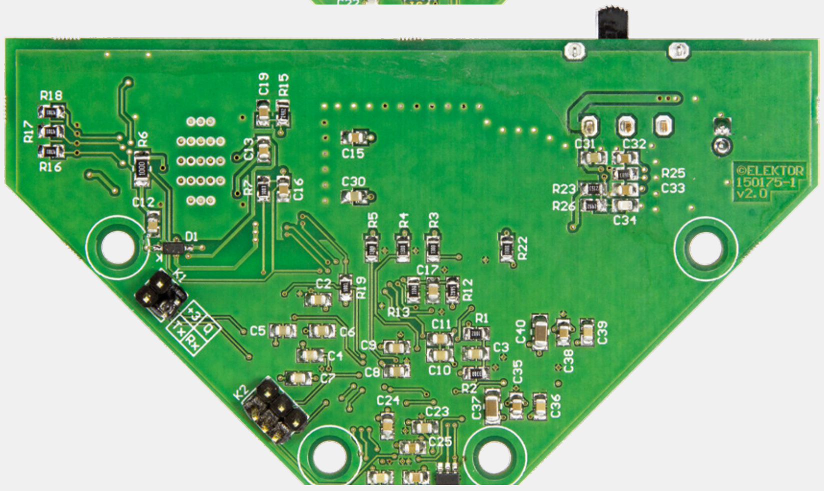


Figure 3. Assembled pendulum, bottom to top: battery, PCB, inertia wheel and motor, all sandwiched between the two half-shells.

The PCB has three LEDs (red, yellow, green) that give a summary indication of what the pendulum is doing. The most important, red LED1, indicates critical errors that prevent the pendulum working. The way it flashes then depends on the

Miscellaneous
F1 = 8A, 125V, SMD
K1 = 4-pin (2x2) pinheader, 0.1''pitch
K2 = 6-pin (2x3) pinheader, 0.1''pitch
K3 = 11-way FFC/FPC ZIF connector, 1mm pitch, right-angle
K4 = 2 way XH disconnectable crimp style connectors
S1 = slide switch, SPDT, right-angle
PCB

Not on PCB
M1 = Motor, EC 45 flat Ø42.9mm, brushless, 12V, 30W, with Hall sensors
BATT1 = LiPo battery, 7.4V, 2S, 30C, 450mAh (Internet)

Mechanical
Shell PCB, 3D printer
Shell Motor, 3D printer
Wheel Axle, 3D printer
Fly wheel, Ø outside: 55mm, Ø inside: 48mm, H: 10mm (Misumi)
Screws for motor, M3, 6mm
Screws for enclosure + PCB, NO6X1/2
Battery adapter cable

Battery adapter cable
RCY connector - socket pin contact
RCY connector - socket housing
XH connector - plug socket contact
XH connector - plug housing
AWG 20 single wire, red, 40 mm
AWG 20 single wire, black, 40 mm

IC6 = ISZ-2510
IC7 = MAX668
IC8 = LP2985-33DBVTE4
IC9 = LP2992IM5-3.3/NOPB
T1 = FDS6680AS

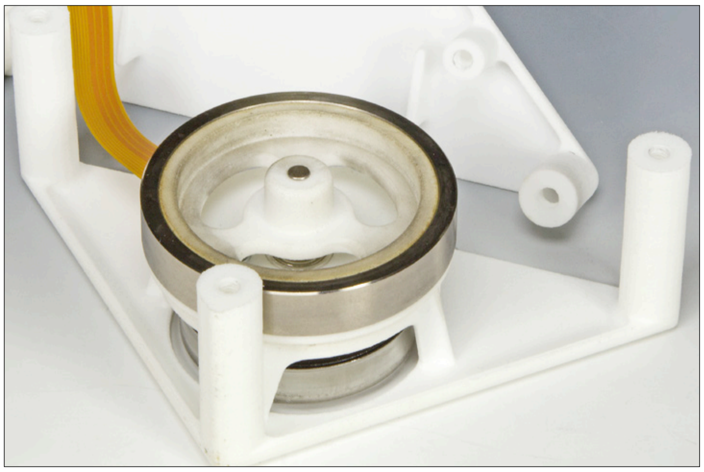
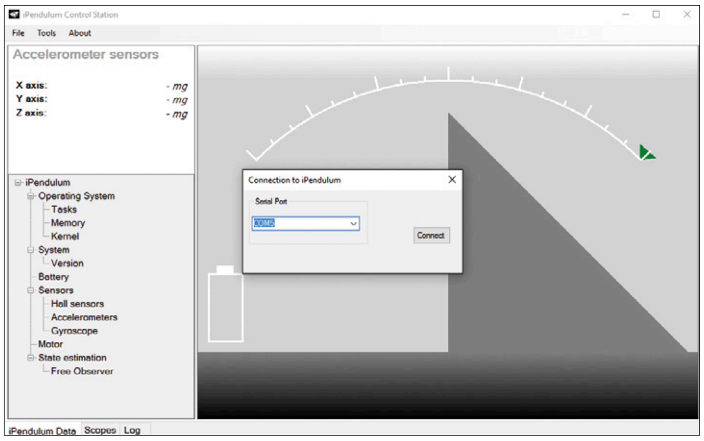


Figure 4. Inertia wheel added to the motor.

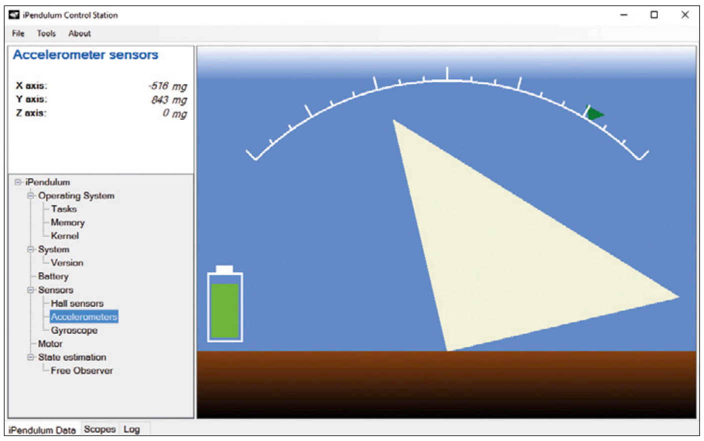
source of the error. Straightforward flashing means that the battery voltage is too low; flashing in bursts of two flashes if the motor is not detected; flashing in bursts of three or four flashes if the processor is unable to communicate with the sensors. When the pendulum is working normally, several states are possible. The first is waiting for a stable position before making



X axis: - mg
Y axis: - mg
Z axis: - mg

Serial Port: COM3

Figure 5. Connecting the pendulum to the *iPendulum Control Station* troubleshooting software.



X axis: -516 mg
Y axis: 843 mg
Z axis: 0 mg

Figure 6. View of the pendulum with the orientation it has in the real world.

a jump. In this situation, none of the LEDs is lit. When the pendulum is ready to make a jump and is accelerating its inertia wheel, the LEDs chase in the direction of the jump. And lastly, green LED3 flashes alone when the pendulum is balancing.

List of parts

Assembling the pendulum is child's play if you have all the required parts. Here's a list of them:

1. PCB that is working, but at this stage still has to be adjusted.
2. LiPo 2S battery 5 cm × 3 cm × 1 cm. I used the 7.4V Lipo 2S 30C 450mAh from TeamOrion. The supply cable for this battery terminates in a JST-RCY connector (known by modelers as a BEC connector), which cannot be soldered to a PCB. So you need to make an adaptor from the battery's JST-RCY connector to the JST-XH connector on the PCB. You can get expensive tools for assembling the wire, plastic connector, and metal terminations all in one go. They also have an irritating habit of being compatible with one type of connector only. My advice: assemble the connectors by hand using a soldering iron. This takes a bit longer, but this is not a problem for just one cable.
3. Three plastic components: one half-shell = motor support, one half-shell = PCB support, the inertia wheel hub (3D printing files from [3]). I used an on-demand printing service ([4]), and I chose basic white plastic as the material. Don't choose any other material, as this will affect the weight of the pendulum and the tolerances for the dimensions.
4. Brushless motor, EC 45 Flat series, no. 200142, from Maxon. This very low profile motor is perfect for slipping into the pendulum.
5. Metal ring to give the inertia wheel its ground; ext. dia. = 55 mm, int. dia. = 48 mm, thickness = 10 mm; part no. AWSM-D-D55-V48-T10 from Misumi ([5]). In designing this project, I purposely avoided the need to machine complicated metal parts. So the complex part of the wheel is printed in plastic, while the outer surround is a simple metal ring.
6. Some screws for assembling all the parts together: three 6 mm M3 screws for fixing the motor, seven other screws for plastic for the PCB and the half-shells.

Mechanical

The inertia wheel is the component that requires the greatest attention. First of all, the metal ring has to be fitted onto the plastic hub. In principle, the dimensions ought to correspond and pressing in the vise is all it will take to assemble them together for good. If the ring has trouble fitting, file the hub down very slightly. If the ring is "too big", add a few drops of superglue. Then drill out the plastic hub's central hole (4 mm Ø) so as to insert the motor shaft there: do this as straight as possible, e.g. by using a pillar drill. Test your bit first on another piece of material. Any imbalance will result in unbalanced running when the wheel rotates.

Fix the motor into its plastic half-shell using its three screws. Pass the ribbon cable through the hole in the shell. Fit the inertia wheel onto the shaft (**Figure 4**). When doing this, press on the motor's central shaft, not on the outer surround, so as not to distort the motor. If it doesn't fit, enlarge the hole slightly.

If it isn't tight enough, fix everything with a little adhesive. *Attention:* the inertia wheel sits just above inductor L1, there is really very little space between them. To remedy this, it is preferable to add a thin washer between the motor and the 3D-printed support when fixing. This makes it possible to have around 1 mm between the flywheel and L1. Next I modified the 3D files to avoid the need to add washers.

The tricky steps are over. Fix the PCB into its half-shell using four of the screws for plastic. Add the battery connecting cable. The final step consists in assembling the two halves of the pendulum together. Connect the motor ribbon cable to K3. Assemble the two half-shells using the remaining three screws for plastic. Congratulations! The pendulum is almost ready to use. All that remains is for you to slip the charged LiPo battery into its space behind the PCB.

Set-up

The pendulum only tries to balance or jump when it is correctly oriented, i.e. when the axis around which it pivots is in contact with the ground. If the motor is lying face down or up, the pendulum goes into standby. So lay it on its left or right side and operate the switch. After remaining still for five seconds in this position, the pendulum will attempt to perform a jump. It spins its inertia wheel up to speed then blocks it so as to jump up to its balance position. You can also put it straight into balance by raising it manually from its rest position. As soon as the pendulum reaches a sufficient angle, the motor provides the torque needed to reach its balance point all on its own. If you move the pendulum out of balance, it will attempt to right itself, insofar as it is able to. However, beyond a certain point, it will have to give up and fall onto its side. After the required five seconds without moving, it will then attempt to jump back up again.

Connection to the control tower

The *iPendulum Control Station* software (available from [3]) is a little program written in C# that runs under Windows. It makes it possible to connect to the pendulum's serial connection in order to collect information about the status of the firmware and to calibrate the internal organs. This program is only used for troubleshooting purposes or while developing the pendulum, so as to observe what is going in inside it in real time.

Run the program, go into the *Tools→Connect* menu. Select the number of the COM port to which the pendulum is connected and establish the connection (**Figure 5**). If everything is alright, the connection dialogue box disappears and the main screen is displayed in color. This contains a stylized representation of the pendulum and the angle is makes with the ground (**Figure 6**). Various tabs are available at the bottom of the page. By default, the main screen is selected and displays the pendulum information in real time.

Description of the application

Move the pendulum by hand, you'll see it move on the main screen. The left-hand pane contains a scrollable tree diagram so you can observe the complete state of the pendulum in real time:

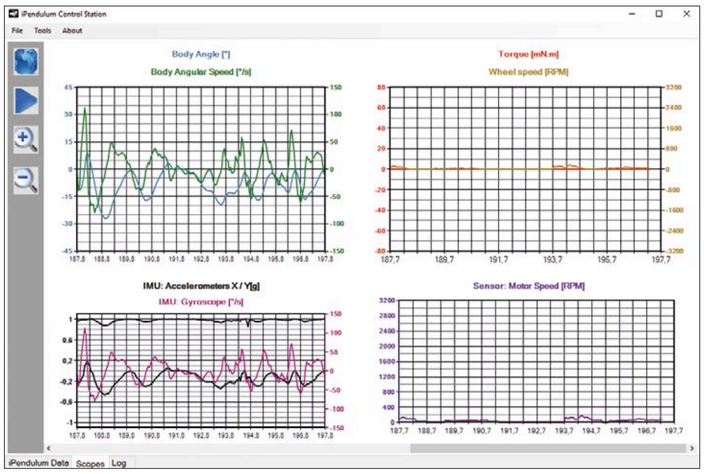


Figure 7. Log of telemetry data transmitted by the pendulum, brought together in four graphs.

- *Operating System*: information specific to the RTOS (version, memory usage, tasks, percentage use of CPU)
- *System*: general information about the firmware
- *Battery*: battery voltage and state of charge
- *Sensors*: raw data returned by the sensors
- *Motor*: torque delivered by the motor
- *State*: pendulum state as estimated by the Kalman filter.

The ***Scopes*** tab (2nd screen) contains a log of the telemetry data transmitted by the pendulum, in the form of four synchronized graphs that always display the same time window (**Figure 7**):

1. top left – estimated state of pendulum
2. top right – information about the motor: torque delivered and speed
3. bottom left – information returned by the accelerometer and gyroscope
4. bottom right – raw speed as measured by the motor controller. This speed is always positive, irrespective of which direction the motor is rotating in, as the controller is physically unable to distinguish between them.

All the graphs are dynamically refreshed as fresh data arrives. You can zoom using the mouse wheel or the dedicated buttons on the left of the screen. You can also scroll the graphs, either using the horizontal scroll bar, or by clicking on a graph and dragging with the mouse.

Under the ***Log*** tab (3rd screen), you can specify the file where you want to store the complete contents of the dialog between the application and the pendulum. Select a file, then click on *Start Logging* to start recording.

Battery calibration

The pendulum sends a measurement of the voltage supplied by the battery. This gives an idea of the state of charge, which avoids reaching the critical voltage below which the LiPo battery will be damaged. This threshold is around 3.7 V/cell, i.e. 7.4 V here. The *Tools→Calibration→Battery* menu makes it possible to calibrate the measuring chain for this voltage in a dialogue box that has two fields for displaying the raw measurement

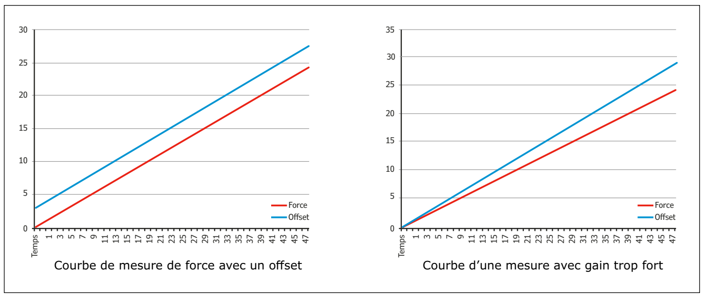


Figure 8. Calibrating the inertia sensors. Left: in red the accurate curve, in blue the curve affected by an offset. Right: in red the accurate curve, in blue the curve affected by a scaling error (gain)

and entering the calibrated measurement. To find this out, measure the battery voltage with a voltmeter.

Calibrating the inertia sensors

The pendulum continuously estimates its rotational speed and its angle to the ground, with the help of the gyroscope and accelerometer. These sensors have numerous specific measurement errors. However, since precision is not critical here, only offsets and gain errors are worth correcting (**Figure 8**). An offset is a constant systematic error that affects all measurements in the same way. To find this out, all you have to do is to put the sensor in a position where it ought to be indicating zero, then read its output value.

This is easy for the gyroscope: to find out the offset, we just read the sensor output when the pendulum is at rest. This easily-corrected error can have a major impact. A gyroscope measures a rotational speed, which when integrated can give an idea of the position. Integrating a constant error then amounts to obtaining a position error that will continue to increase. Note too that the offset in a gyroscopic sensor has a tremendous tendency to drift with temperature or to vary slowly over time.

The situation is less simple for accelerometers, as the sensor must be positioned in an axis perpendicular to a horizontal plane in order to obtain a zero value. Easier said than done! It's hard to align a sensor perfectly like this. The solution is to take the average of the accelerometer values in two mirror-image situations, in such a way that the deviations with respect to the horizontal cancel each other out. Accelerometer offsets have a lesser effect than on gyroscopes. The accelerometers give an image of the pendulum position directly, without requiring an integration step.
A scaling error affects the gain of the system. Thus the value supplied presents non-unity gain compared to the exact value. The gyroscope gain is not corrected in the pendulum, as measuring this error requires special equipment to turn the pendulum at a precise speed. However, it is possible to correct the accelerometer gains, as we have a ready-made reference value: gravity. If the pendulum is lying flat or on its side, the accelerometer will measure either zero acceleration or a precise fraction of *g*.

It's complicated, but the calibration procedure (*Tools→Calibration→IMU*) is there to help you. All you have to do is to put the pendulum in a well-defined position each time you are prompted to do so. Follow the instructions right to the end to complete the procedure.

Conclusion

Now you know all about the inverted pendulum. I hope this project has roused your curiosity and lifted just a corner of the veil on modem regulation methods. This pendulum has no inherent use other than to amuse, intrigue, or simply encourage you to discover and experiment with new techniques. However, we shouldn't lose sight of the fact that these techniques are omnipresent in our daily lives, whether in the objects we use or in Nature. Control loops are in use in cars and communication systems, and in planes and rockets for following a stable trajectory. The living world too exhibits such abilities, for example in the interactions between proteins in cells, and for any other biological parameter that needs to be regulated: temperature, blood pressure, etc.

The pendulum codes (firmware and control tower) being completely open, now it's your turn! ◀

(150695)

Web Links

[1] www.genid.eu/dp/?q=Products\NanoScale\WhatIsNanoScale

[2] www.st.com/st-web-ui/static/active/en/resource/technical/document/design_tip/DM00052458.pdf

[3] PCB files, firmware, 3D printer files (half-shells, inertia wheel hub), Windows software: www.elektormagazine.com\150695

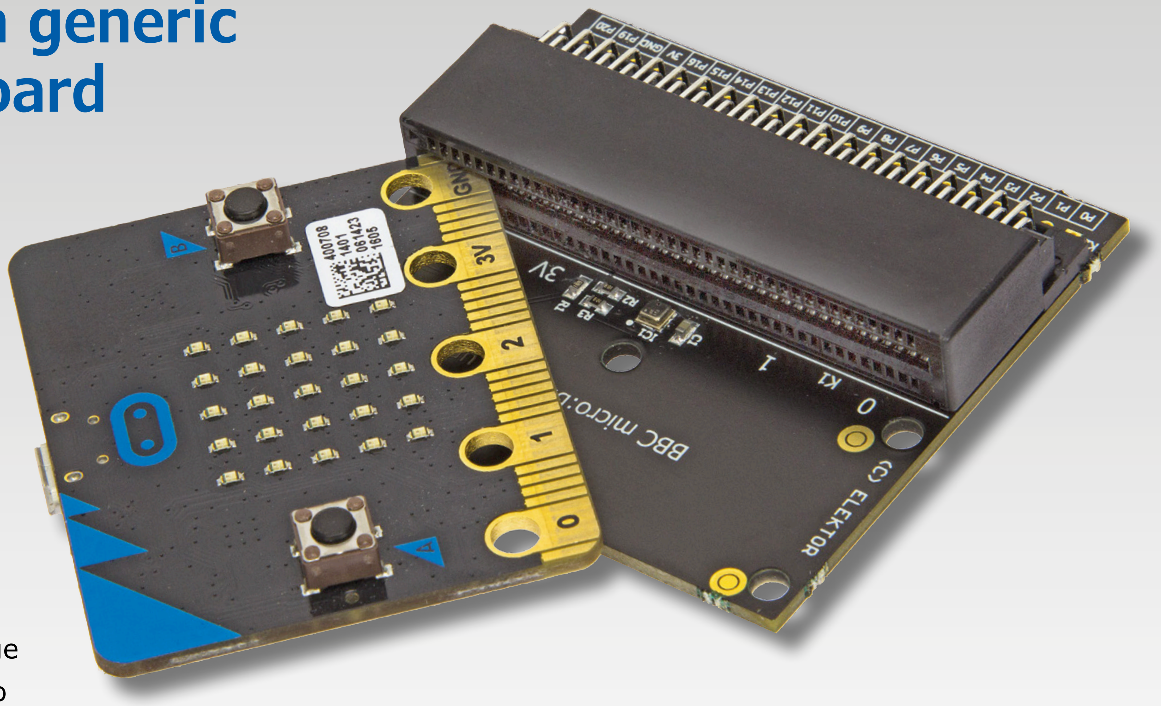
[4] On-demand 3D printing: www.sculpteo.com

[5] Misumi: www.misumi-europe.com

BBC micro:bit Weather Station

Doubles as a generic extension board

By **Clemens Valens**
(Elektor Labs)



The BBC micro:bit is a small powerhouse loaded with cool functions and supported by a suite of excellent development tools and libraries. Its edge connector allows access to goodies like GPIO ports, analog inputs, an I²C bus and an SPI port. To show how to use it we build a small weather station for it.

People love to measure. Give a child a tape ruler and it will immediately start to measure everything in sight. The BBC micro:bit, initially intended for children, is equipped with two sensors, an MMA8653 3D accelerometer that measures acceleration on three axes, and a MAG3110 3D magnetometer that measure magnetic field strengths on three axes. The latter also measures its die temperature and outputs it, giving yet another measurement result. These sensors are intended for orientation, movement and gesture detection. To complete this set of sensors, we added a BME280 combined humidity, pressure and temperature sensor from Bosch Sensortec. Typical applications of this sensor are context awareness, skin detection and vertical navigation, but it also happens to be an excellent weather sensor. It provides very precise humidity, atmospheric pressure and temperature data on an I²C or SPI bus. The joint force of the three sensors allows you to turn the micro:bit into an accurate inertial measurement unit (IMU). Or a wireless weather station if you add micro:bit's Bluetooth to the mix.

Hardware

The schematic of our add-on board is pretty straightforward (**Figure 1**): a sensor (IC1, **Figure 2**), an LED, three resistors and a capacitor. The hard part this time is the edge connector K1 because it is difficult to find. For the rest, there is not much to tell about such a simple design.

We connected the CSB signal of IC1 (pin 2) to V_{IO} (which is connected to V_{CC}). This selects the device's I²C interface. IC1 can also do SPI in three- or four-wire mode, but for that the CSB pin must be pulled Low. Once CSB is sampled Low, the device remains in SPI mode. If, for some reason, you want to select the I²C interface by using an external signal, this signal must have a level of V_{IO} before the chip is reset.
The SPI interface is compatible with SPI mode '00' (CPOL = CPHA = '0') and mode '11' (CPOL = CPHA = '1'). Mode selection is automatic, and determined by the value of SCK after the CSB falling edge. Three- or four-wire mode is controlled in software by the SPI3W_EN bit. SDI is the data line in three-wire mode, SDO is not used in this case.
Because we use the device in I²C mode, we provided the possibility of adding pull-up resistors to the bus (R2 & R3). If you only use the extension board with the micro:bit you don't need these, because they are already present on the micro:bit. When the BME280 is in I²C mode, its SDO pin (pin 5) determines its Slave address. Connecting SDO to GND sets the Slave address to 0x76; connecting it to V_{IO} like we did results in 0x77. The SDO pin cannot be left floating.
LED1 is a power indicator — it's helpful because the one on the micro:bit itself becomes invisible when the board is inserted in connector K1. LED1 remains visible at all times.
K2 is a pin header that provides easy connections of the

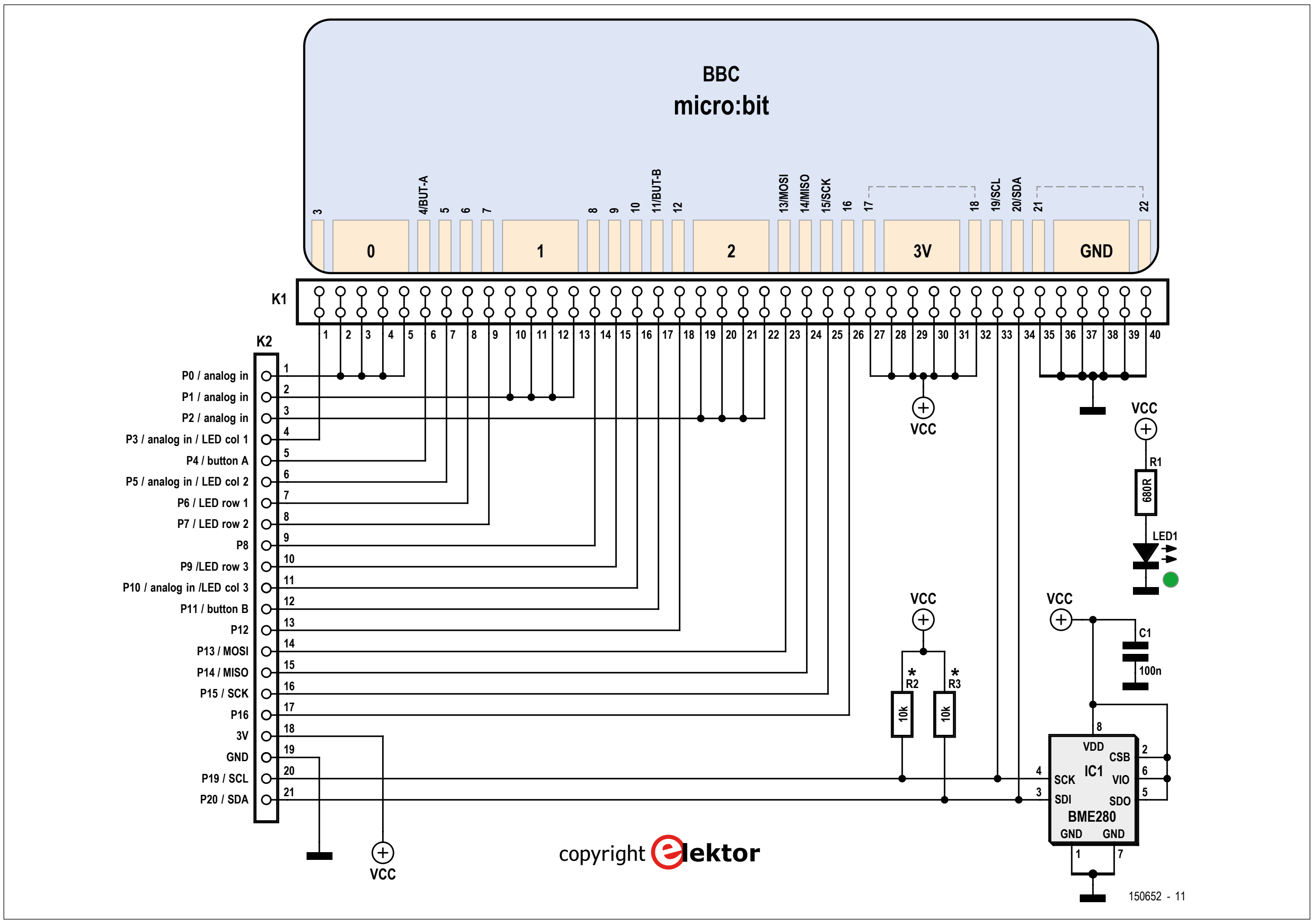


Figure 1. Schematic of the weather station extension board for the BBC micro:bit.

micro:bit edge connector to, for instance, a breadboard. The order of the pins on K2 is slightly different from the micro:bit because we chose to put them in increasing order from 0 to 20. On the micro:bit pins 0, 1 and 2 correspond to the large banana-plug-compatible holes that each take up several contacts of K1. The other contacts count from 3 to 21 from left to right (LED matrix side visible, edge connector at the bottom).

Software...

... is needed for the microcontroller to be 'at speaking terms' with the BME280 and so we wrote a driver for it. While we did this the micro:bit was still unreleased and the nice libraries for it were not available yet. However, the board was already a known platform in mbed [1] and so we wrote our driver as an mbed project. We also did an Arduino Sketch for it, you can get it at [2] or [3], so you can use the sensor with an Arduino compatible board.

Bosch Sensortec provides a driver for the BME280 on GitHub, but it is rather complicated, not only because it is supposed to work on all sorts of platforms, 64-bit, 32-bit, 16-bit, with or without floating point units (FPUs), but also because it supports every detail of the chip. For our simple weather station application we don't need all this, and since it is rather instructive to write your own driver, that is what we did. Our driver [2],[3] also lets you control every bit in every corner, but you will need the datasheet to figure out what to write to, and read from, specific locations. And since C++ permitted us to write an Arduino-style API, we wrote our driver in C++.

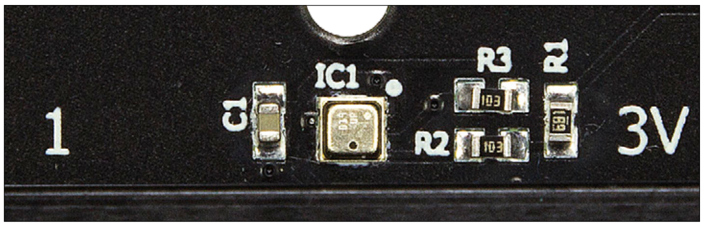


Figure 2. IC1, the tiny metal can in the middle of the board, contains three high-precision sensors.

Using our driver is easy enough. It starts by creating a BME280 object and then calling its `begin` function, as has become customary since Arduino rules the world:

```
BME280 bme280;
bme280.begin(0x77);
```

The function (or *method* if you prefer) `begin` takes the I²C address as argument, on our board it is hardwired to 0x77. If you choose not to provide an address, the driver will default to an SPI bus. The function returns 0 if the sensor was found on the bus and if its calibration data could be read. The next step is to configure the sensor:

```
bme280.writeConfigRegister(BME280_STANDBY_500_
US,BME280_FILTER_OFF,0);
bme280.writeControlRegisters(BME280_
OVERSAMPLING_1X,BME280_OVERSAMPLING_1X,
BME280_OVERSAMPLING_1X,BME280_MODE_NORMAL);
```

We used basic settings here — no fancy stuff, no high speed, no filtering. Now you're ready to collect data. You do this by first calling the function `read` to get the data from the sensor into the driver and then by calling one or more of the member functions `temperature`, `pressure` and `humidity`. Typically you would do this in a loop, but you don't have to.

```
bme280.read();
printf("T=%d degrees C, ", bme280.temperature()/100);
printf("P=%d mbar, ", bme280.pressure()/100);
printf("RH=%d%%\n", bme280.humidity()>>10);
```

Note how the values obtained are being scaled. `Temperature` must be divided by 100 to obtain degrees Celsius. `Pressure` is expressed in Pascal and to be practical must be divided by 100 to obtain milli-bars (mb). `Humidity` has to be divided by 1024 (which is the same as shifting it 10 bits to the right) to obtain a percentage. It is possible to enable floating point arithmetic in the driver by making `BME280_ALLOW_FLOAT` unequal to 0 (at the top of the file bme280.h):

```
#define BME280_ALLOW_FLOAT  (1)
```

This will probably make the executable bigger and slower, but it depends on your application if that presents a problem or not. The floating point versions of the member functions `temperature` and `humidity` do not require scaling, `pressure` is again in Pascal.

Our driver can handle both SPI and I²C busses (although we didn't test SPI) and it is the user's responsibility to provide the functions for it. The driver will call

```
i2cWrite(…) & i2cRead(…)
spiWrite(…) & spiRead(…)
```

as needed, depending on how it was set up. All four functions must be provided, but may remain empty. If you use I²C leave the SPI stubs empty, if you use SPI then leave the I²C stubs empty.
Note that in mbed I²C addresses are 8-bit because the read/write bit is included at bit 0, meaning that 0x77 becomes 0xee. In Arduino the address is 0x77.

Expanding further

Connector K2 is supposed to be a pinheader, but you are free to make anything that suits your application. With a pinheader or socket it is easy to connect the board to a breadboard, especially if you have some of those very practical breadboard wires. If you mount a pinheader, horizontal or vertical, you will be ready for our upcoming micro:bit project: the micro:bit Dock. Stay tuned…

(150652)

Web Links

[1] https://developer.mbed.org/platforms/Microbit/

[2] https://github.com/ElektorLabs/bme280-driver

[3] www.elektormagazine.com/150652

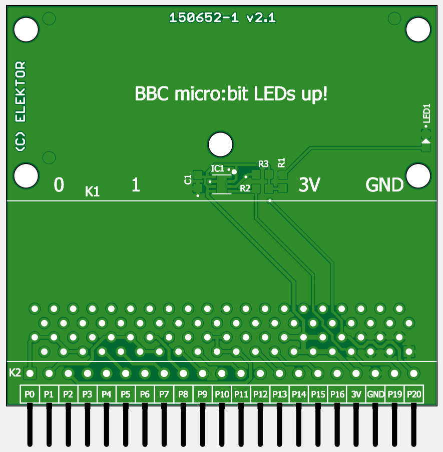
Component List

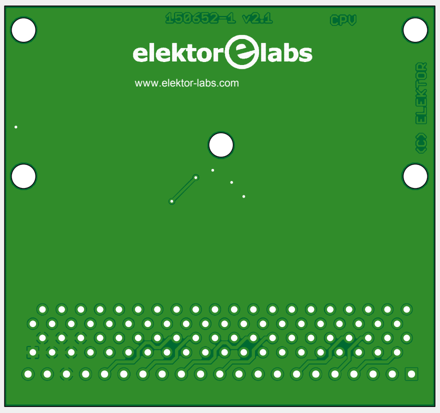
Resistors
5%, 50V, 0.1W, 0603
R1 = 680Ω
R2, R3 = 10kΩ

Capacitor
C1 = 100nF, 0603

Semiconductor
IC1 = BME280
LED1 = LED, green, 0603

Miscellaneous
K1 = socket, PCB-to-PCB, 2x40 links, 0.05'' (1.27mm) pitch
K2 = pinheader, 1x21 pins, 0.1" pitch
PCB # 150652-1 v2.1


150652-1 v2.1
(C) ELEKTOR
BBC micro:bit LEDs up!
LED1
0
K1
1
IC1
R3
R1
R2
C1
3V
GND
K2
P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 3V GND P19 P20



Bat Detector PLUS

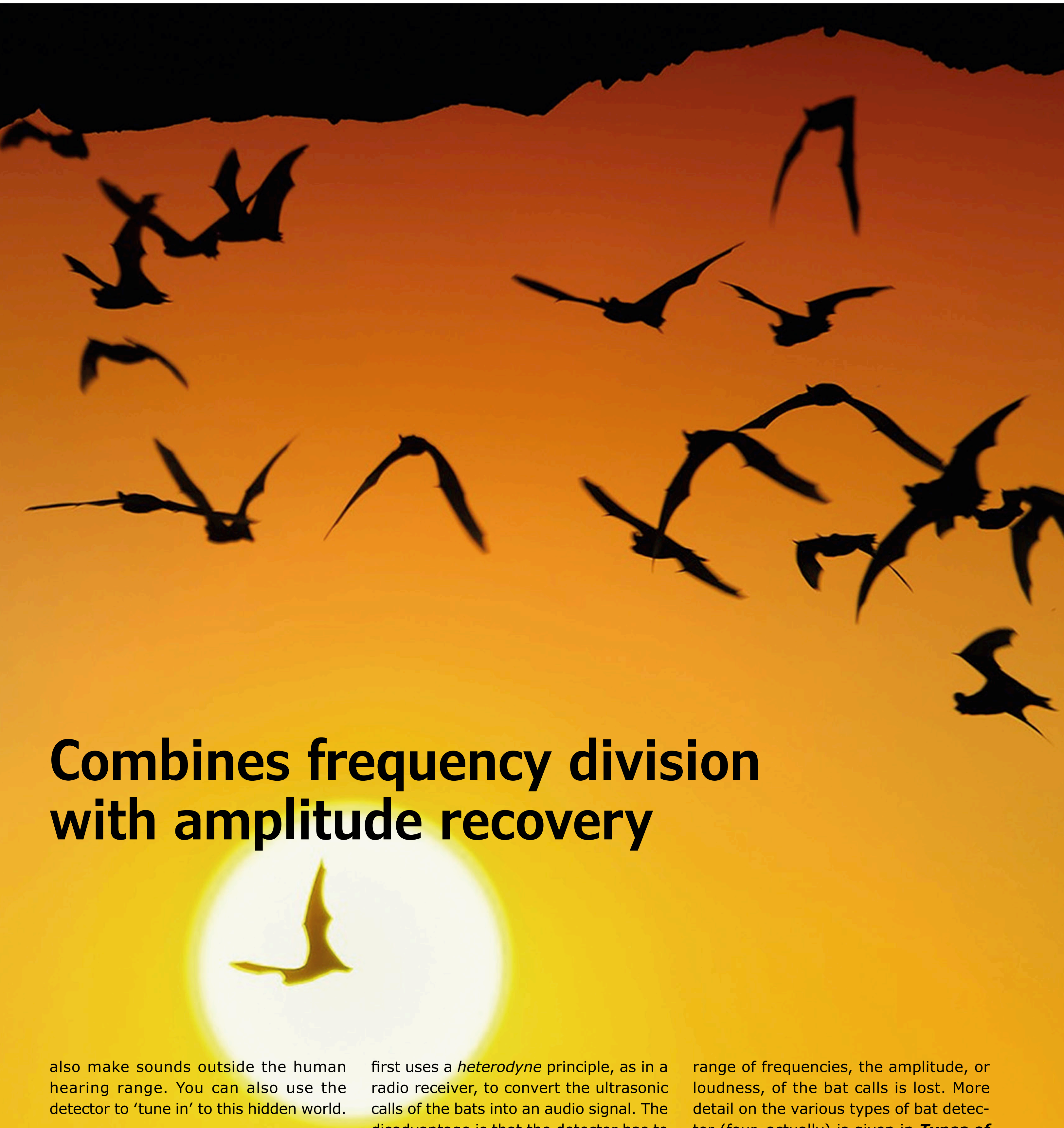
By **Kenneth Horton** (United Kingdom)

Having an apartment in Cyprus, each evening I can see bats fly past my balcony on the way to their feeding grounds. I decided that it would be a good idea to 'tune in' to the bats as they flew past. The idea culminated in a bat detector that's possibly the most refined of its kind published in Elektor.

Bats use ultrasound to both communicate and hunt. Ultrasound is just part of the audio spectrum; a pressure or sound wave that travels through air. Young humans can have a hearing range from about 20 Hz to 20 kHz, but the upper frequency limit drops off quickly as we age. Bats use higher frequencies from 20 kHz to over 100 kHz.

There are many species of bat. Each has its own distinctive call. These are a series of short clicks, and the bat listens for echoes between the clicks to locate its prey. Some bats emit a constant frequency whilst others use a range of frequencies within a call sweeping from high to low frequency. The duration and repetition rate of the call also varies between species.

Most species have calls that fall within the range 30 kHz to 70 kHz but there are exceptions. For example, the call of the Common Noctule (*Nyctalus noctula*) may be as low as 18 kHz whilst the call of the Lesser Horseshoe (*Rhinolophus hipposideros*) can be as high as 100 kHz. More detail can be found on the Durham Bat Group website [1]. Bats are not the only animals to use high frequency communication. Many small animals and insects



Combines frequency division with amplitude recovery

also make sounds outside the human hearing range. You can also use the detector to 'tune in' to this hidden world.

Heterodyning or dividing?

In order to listen to bats you need some sort of detector that will convert the ultrasound bat calls into a sound that humans can hear. A trawl of the Internet found circuits for two types of bat detector. The first uses a *heterodyne* principle, as in a radio receiver, to convert the ultrasonic calls of the bats into an audio signal. The disadvantage is that the detector has to be tuned and can only receive a narrow band of frequencies at any one time. The second type of detector *divides the frequency of the ultrasonic calls*, often by a factor of 16, so that they become audible. Although this works over a wide range of frequencies, the amplitude, or loudness, of the bat calls is lost. More detail on the various types of bat detector (four, actually) is given in ***Types of Bat Detector*** inset.

Only one circuit, designed by J L Errington and S Frenehard, was found that attempted to overcome the disadvantages of both types of detector. This used the frequency division method, but also

Main Features & Specifications

- Hybrid analog and digital circuitry
- Low current consumption: less than 5 mA in standby
- Selectable division ratio: 16, 17, 23 and 31
- Selectable low frequency cut-off: 8 kHz, 15 kHz, 18 kHz 25 kHz
- Selectable sensitivity verses battery saving
- Improved amplitude recovery
- Audio sinewave output
- Audio amplifier muted and in standby mode when there is no input signal
- 'Muted' and 'signal detected' LEDs
- Hardware and software test mode
- MEMS or electret microphone
- 1 (one) SMD device

detected the amplitude of the bat call and fed this through to the audio stage. Unfortunately, the designers had proved each stage of the design individually, but they had not produced a working prototype. Undeterred, the author redesigned parts of the original circuit and produced a very successful working detector. Details can be found on J L Errington's website [2]. Although successful, the detector still had a number of shortcomings:

- Although not unacceptable, the standby current consumption was about 15 mA;
- The division ratio was 16 so the input and output were harmonically related, increasing the chance of unwanted feedback;
- The output was a rather harsh square wave;
- The detector was susceptible to unwanted input such as high-pitched voices.

So I redesigned the detector to overcome these disadvantages, and produced an updated version, whose performance and salient qualities are listed in the ***Main Features and Specifications*** box.

Circuit description

The circuit consists of six basic modules as summarized in **Figure 1**.
A power supply provides the reference voltage for the operational amplifiers and also 5 volts for the PIC microcontroller.
The microphone picks up the sounds made by the bat. The input from the microphone is firstly amplified by a 3-stage preamplifier. This boosts the weak signals from the microphone. The signal is then split into its amplitude and frequency components. The amplitude is extracted with a 'precision rectifier' whilst the frequency is shaped for digital processing by a Schmitt trigger.
The separate amplitude and frequency signals are then fed to a PIC microcontroller. This uses the input frequency to synthesize a sinewave at a frequency within the human hearing range. The amplitude signal modulates the sinewave so that the loudness of the bat call is reproduced.
Finally, the sinewave is amplified and fed to a loudspeaker or headphones. To save power, the amplifier is muted when there is no input.

At this point we're ready to discuss the actual design whose schematic appears in **Figure 2**. Let's take it section by section.

Power supply

The circuit is designed to operate from a single 9-volt PP3 (6LR22; 6F22) battery. In order to provide a reference voltage for the operational amplifiers, the battery voltage is split by R1 and R2 and fed to an operational amplifier buffer IC1. This provides a zero-volts reference, with the battery providing plus and minus 4.5 volts (±4.5 V) relative to the reference. C1 through C10 provide smoothing of the supply voltage rails. Although slightly unconventional, C9 and C10 were found essential to prevent unwanted self-oscillation, which was a major problem with the original design concept on the Internet. There is also a power supply for the PIC microcontroller which requires a 5-volt

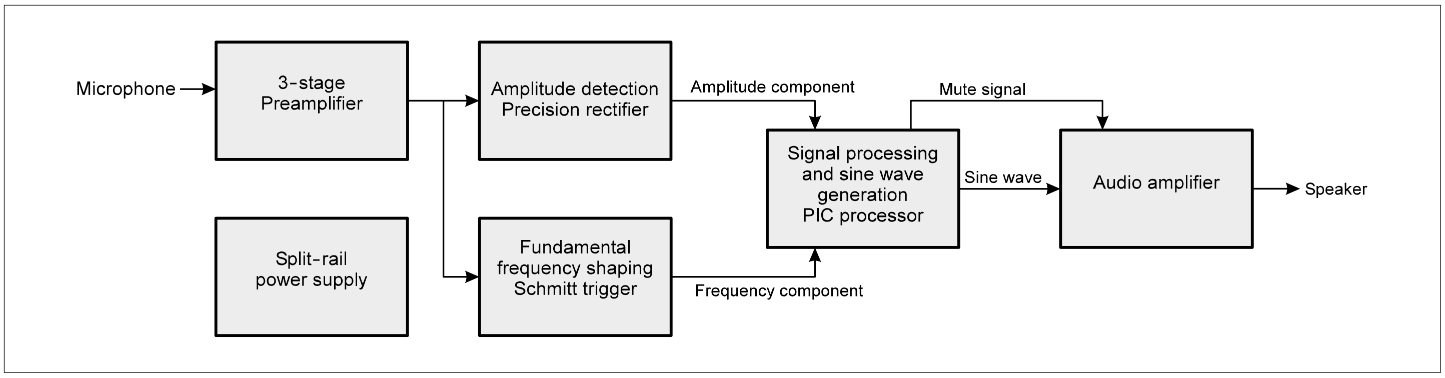

Microphone
3-stage Preamplifier
Amplitude detection Precision rectifier
Amplitude component
Mute signal
Signal processing and sine wave generation PIC processor
Sine wave
Audio amplifier
Speaker
Split-rail power supply
Fundamental frequency shaping Schmitt trigger
Frequency component

Figure 1. Functional analysis of the circuit in the form of a block diagram.

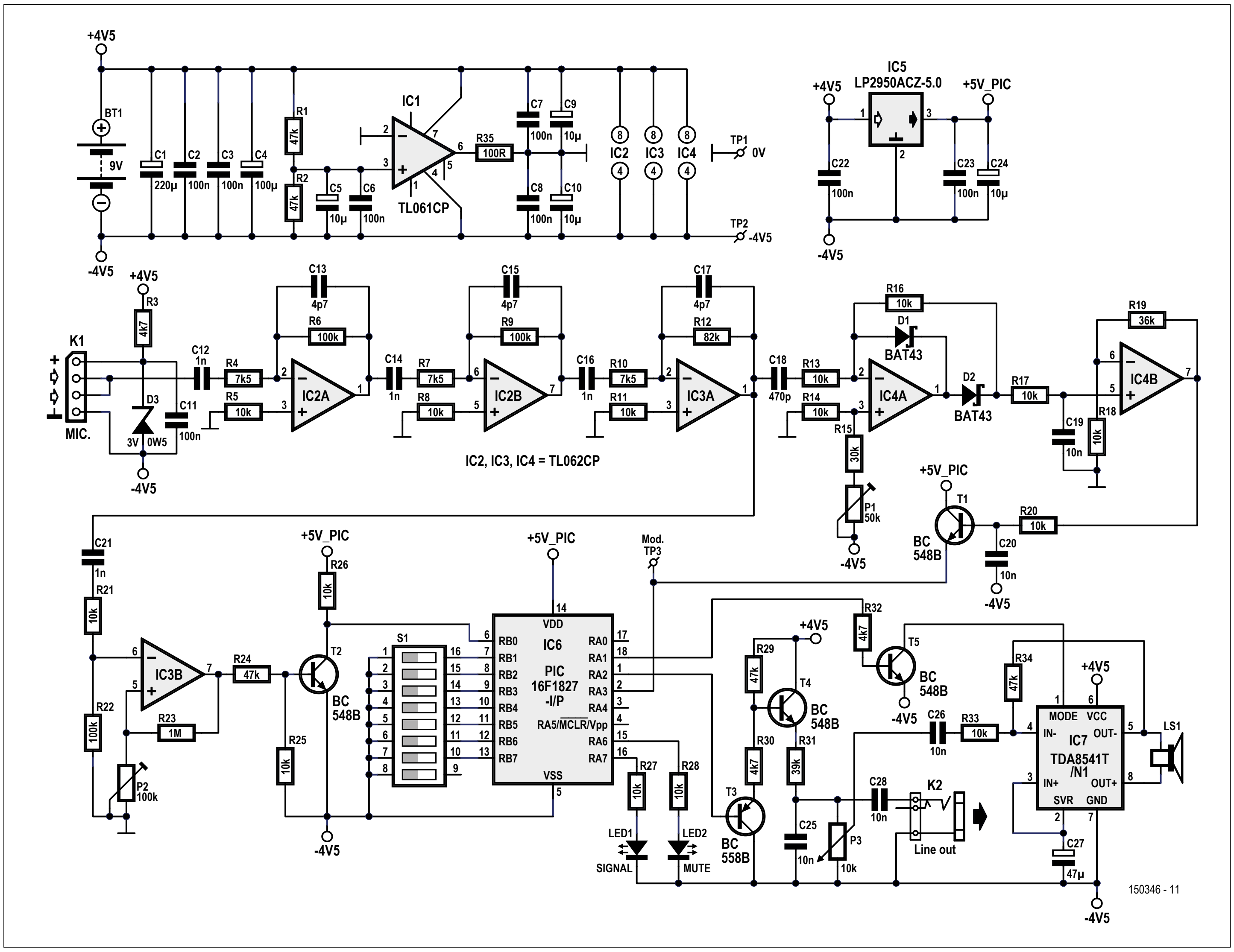


Figure 2. Schematic of the bat detector based on **both** frequency division and loudness detection.

supply. This is provided by a 100-mA 5-volt regulator, IC5 and smoothing capacitors C22 to C24.

Microphone preamplifier

The microphone is connected via connector K1. Components R3, C11 and D3 allow for a range of microphone options. The choice of microphone is discussed later. The output from the microphone is passed through C12 and R4 that form a simple high-pass filter and into an operational amplifier, IC2A. This provides a gain of approximately 13. C13 provides modest attenuation to unwanted higher frequencies. A TL062 was chosen for IC2 for its ultra-low power consumption. The TL062CP has a gain bandwidth of 1MHz (although some manufactures claim 2 MHz). With a gain of 13, this gives a theoretical upper frequency cut-off of about 77 kHz. In practice, this was found to be considerably higher.

The output of IC2A is passed through two similarly configured amplifiers IC2B and IC3A resulting in an overall gain of about 1,860.

At this stage, the bat call is split into its fundamental high-frequency and amplitude components.

Amplitude detection

In order to isolate the amplitude component, C18 and R13 form a simple high-pass filter that feeds a precision rectifier, or 'super diode' based around IC4A, R16, D1 and D2. The constellation acts as a conventional diode, but without the associated forward voltage drop. With an ordinary diode, small input voltages (less than the diode's 'forward voltage') are lost. With a precision rectifier, even small input voltages are transferred to the output, and so the output tracks the input more faithfully.

In the absence of an input signal, the output of a conventional precision rectifier circuit is at reference voltage and it moves towards the positive supply rail when a signal is input. The latter digital stages of the bat detector need the amplitude to be referenced to the negative battery voltage. In order to achieve this, a variable bias voltage is added to the positive input of IC4A via R15 and trimpot P1. This biases the output of IC4A toward the negative supply rail in the absence of an input signal.

The high-frequency component of the output from IC4A is removed by R17 and C19, leaving just the amplitude component of the bat call. This is amplified by approximately 4.6 times by IC4B. The output of IC4B is passed to transistor T1 which acts as an emitter follower buffering the output from IC4B. The emitter of

Listing 1. PIC source code (extract only; full file available at [3])

```
; We have a valid signal
Mainloop_20
  btfss Switch_7    ; Switch 7 - Amplifier inhibit
  goto  Mainloop_30
  Amp_on        ; Switch on Amp

Mainloop_30
  Set_timer_0    ; Timer 0 is used for low frequency cut-off timing
  Signal_LED_on

; NOTE the timing of the remainder of this code is critical
; Do not add to the overall size of the loop.
; The loop will handle input frequencies up to about 150 KHz.

Mainloop_40
  movfw Division_ratio   ; Number of steps in the sinewave
  movwf Count    ; Used as sinewave table offset

; Wait for interrupt - PORTB 0 to go High
Mainloop_50
  btfsc INTCON,TMR0IF  ; Test for Timer 0 timeout
  goto  Mainloop_10  ; Frequency too low - reject

  btfss INTCON,INTF  ; Wait for next input INT pin (B0) (Schmitt trigger in)
  goto  Mainloop_50  ; Go back and wait

  movfw Low_cutoff   ; Timer 0 is used for low frequency cut-off timing
  movwf TMR0     ; Reset timer 0

  bcf INTCON,INTF  ; clear interrupt on INT pin (B0) (Schmitt trigger in)

  movfw Count    ; Use count to select entry in sinewave table
  addwf Table_pointer,w  ; Add in the base of the table
  callw      ; Get the sign wave amplitude value

  movwf INDF0    ; Write to D to A.
                 ; Use indirect addressing so don't need banksel!

  decfsz  Count,f    ; The sinewave is read end to beginning to save time
  goto  Mainloop_50  ; by having end of loop = 0
  goto  Mainloop_40  ; Count expired, need to reload
```

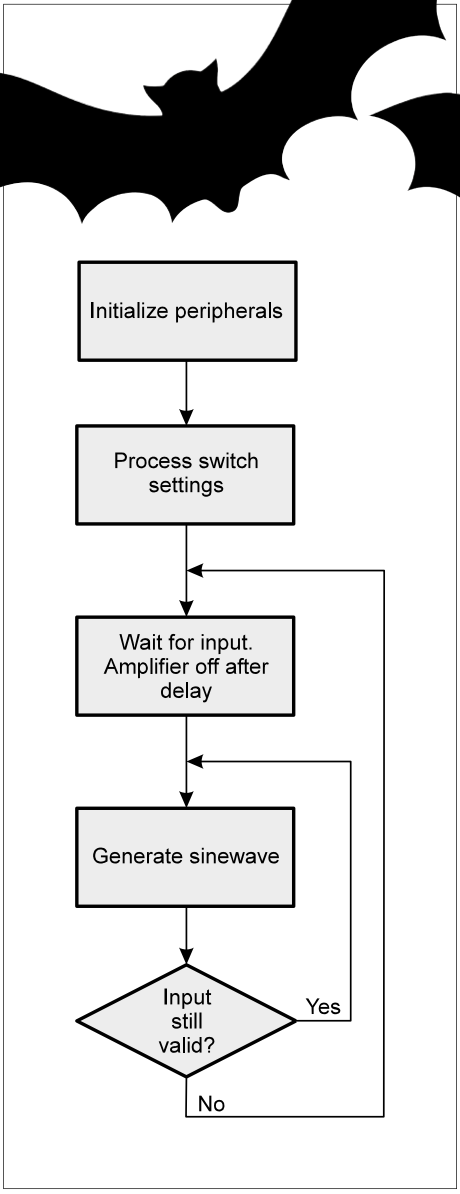


Figure 3. Extremely simplified program flowchart.

T1 is approximately 0.3 volts with respect to battery minus in the absence of an input signal, and reaches approximately 4.4 volts in the presence of a very strong signal. This amplitude signal is input to the PIC microcontroller. Capacitor C20 provides further filtering of any remaining high-frequency components.

Schmitt trigger

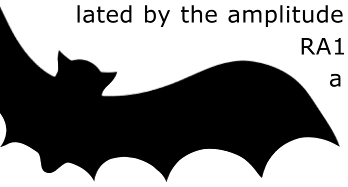
Returning to the output of IC3A, the high-frequency element of the bat call is processed by IC3B. Firstly, the signal is passed through a simple high-pass filter, C21 and R21. IC3B is configured as a Schmitt trigger which produces a squarewave output from the input signal. The Schmitt trigger adds a variable amount of hysteresis to the input signal via R23 and trimpot P2, and this prevents small fluctuations of the input signal — around the reference voltage — causing false triggering. The output of IC3B is buffered by T2 in order to reference it to the battery negative (minus) rail, before being passed to the PIC microcontroller. The remainder of the circuit is referenced to battery minus.

Microcontroller

The microcontroller, IC6 is the heart of the detector. It uses the built-in 16-MHz internal oscillator for its clock. The main functions of the microcontroller are to:

- reject input frequencies below the lower threshold;
- synthesize an audio sinewave from the frequency and amplitude components;
- mute the audio amplifier when it is not in use;
- indicate the status on two LEDS.

In more detail, the microcontroller processes two input signals. The analog input RA3 receives a voltage relative to the amplitude of the bat call whist RB0 receives a square wave equal to the frequency of the bat call.
Seven DIP switches are connected to RB1-7 respectively, and control how the microcontroller processes the input information.



There are two outputs from the microcontroller. RA2 provides an audio sinewave (approximation) which is a divided-down version of the input frequency, modulated by the amplitude input. Port line RA1 then, provides a mute signal for the audio amplifier when there is no input.

The two status LEDs, LED1 (SIGNAL) and LED2 (MUTE), are driven by RA6 and RA7.

Audio amplifier

The sinewave audio output from the microcontroller has low drive capacity and needs to be buffered before it can be used. The lowest part of the sinewave is always close to battery minus whist the peak will be a few hundred millivolts for quiet signals, rising to about 4.4 V for very loud signals. T3, R29 and R30 form an emitter follower which tracks the output signal. The emitter of T3 will always be about 0.6 volts higher than the microcontroller output. R29 and R30 are a potential divider that feeds a second emitter follower, T4. This re-references the signal to battery-minus via R31 and the volume control potentiometer P3. Capacitor C25 provides some smoothing of the synthesized sinewave. The audio is then fed to audio amplifier IC7 via volume control P3, C26 and R33. The amplifier gain is set by R34. The TDA8541 amplifier was chosen as it is specifically designed for battery operation and can be switched into standby mode to save power when not in use. This is achieved by R32 and T5.

Software

The firmware running in the PIC micro is linear from beginning to end of the program with jumps forward or backward as required. In the source code, a number of macros are used, purely to aid clarity within the code. The source code is a free download for all readers [3]. The much requested **PIC fuse settings** are in the same zip file. Downloading it is recommended for an understanding of the quick description below. A snippet of the PIC source code is given in **Listing 1**. If you have a fear of PIC source code, you may have to content yourself with the flowchart in **Figure 3.**

The first section of the program initializes all the microcontroller peripherals. This is followed by a section that pre-processes the switch settings and stores appropriate values in RAM to speed up subsequent processing.

The program then waits for an input signal on RB0. This is processed and compared with the selected low frequency cut-off threshold. If successful, the amplifier is switched on and control is passed to the audio sinewave generation section. Whilst a valid signal is being received, greater than the lower frequency threshold, the software continues to generate a sinewave, which is based in the input frequency. Once the signal stops, control is passed back to the routine that waits for a valid input signal and if, after a short delay, no new signal has been received, the amplifier is switched off.

The sinewave generation routine will successfully process input signals to over 150 kHz.

Built-in timer 0 is used for low frequency threshold measurement. Timer 1 is used to control the amplifier timeout period.

There is a certain amount of 'jitter' around the low frequency cut-off threshold. This can lead to pulses being missed when the input frequency is close to the cut-off frequency. This is because the input pulses arrive asynchronously to the timing loop within the program. This

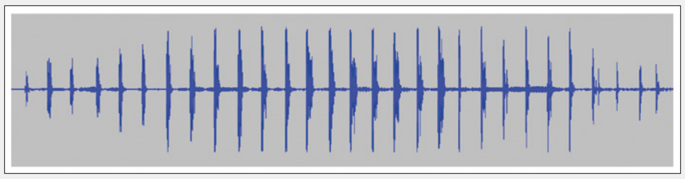
Types of bat detector

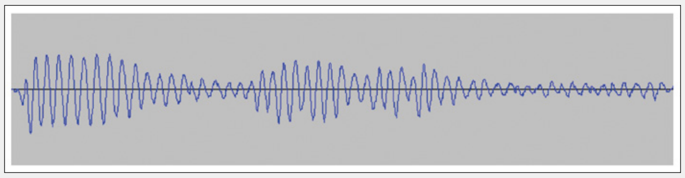
There are four types of commonly used bat detector.

Heterodyne detectors mix an oscillator frequency close to the bat's call with the bat call itself. This produces two new frequencies, one the sum of and the other the difference of the two frequencies. If the difference frequency is within the human audio spectrum it can be fed to an amplifier. The principle is very similar to that of a radio receiver. The advantage of a heterodyne detector is that it retains the amplitude and other characteristics of the bat call. The disadvantage is that it can only receive a very limited range of frequencies at any one time and has to be tuned like a radio receiver. If you are not tuned in, you don't hear the bat!

Frequency division detectors divide the frequency of the bat's call using a digital counter; often by a factor of 16. This converts the call in to the range of human hearing. The advantage is that even the highest and lowest pitched bats can be heard without any adjustment of the detector. The disadvantage is that the amplitude and characteristics of the bat's call are lost. Unless the output is processed, the detector also generates a rather harsh square wave output.

Time expansion detectors work by recording the bat's call, for example to a computer memory chip, and then playing the call back at a much slower speed, such as by a factor of ten. The advantage of a time expansion detector is that it retains the amplitude and other characteristics of the bat call. The disadvantage is that whilst it is playing back it is not recording, so the detector is only listening for a tenth of the time.





The final approach is different in that it is not real time. Modern digital recorders, such as laptops and tablets can often capture very high frequency sounds when used with a suitable microphone. You do not know what you have recorded, but signals can be slowed down and played back in the comfort of your own home long after the bats have gone!

means there can be a slightly different delay before recognizing each pulse. This is not a problem in practice, because the cut-off frequency can be set well below the bat calls being detected.
As mentioned previously, the seven DIP switches control various aspects of the software. Their function is summarized in the ***DIP Switching!*** inset. The default (open) settings are all that will be required for the majority of uses, and the DIP switch could be omitted, or wire links used to permanently select various functions.
It should be noted that the switch settings are only read when the detector is first switched on. If the switch settings are altered, the detector needs to be switched off and on again.

Construction
The PCB designed by Elektor Labs is shown in **Figure 4** along with the **Component List**. The board designed by Elektor Labs fits in a Camdenboss BIM box. The corners of the PCB are shaped so that it fits snugly into the box.
The microphone, volume control (P3), two LEDs, battery and 8-ohm speaker are all fitted within the body of the box and are attached to the PCB with short lengths of wire. **Figure 5** shows the labs prototype which was found to work exquisitely. Thin screened cable should be used for the microphone.
If required, a 3.5-mm jack socket can be wired across the ends of the volume control P3 via a 10-nF DC blocking capacitor (C28) to provide a 'Line-Out' signal for connecting to a recorder (like a laptop or smartphone). This is detailed on the schematic. A 'switching' 3.5-mm jack socket can also be fitted in the speaker lead to provide a headphones output. Both jack sockets are optional extensions.
Start populating the board with the small components such as resistors, capacitors and diodes. It is best to use IC sockets for the DIP-case ICs. IC7 is a surface mount SO-8 IC, and is the only surface mount component.
Care should be taken when fitting the electrolytic capacitors, transistors, ICs and diodes to ensure that they are the

DIP Switching! (summary)

The Bat Detector^{PLUS} can be configured in a number of ways to suit the user's requirements and preferences. Configuring is done through DIP switch block S1.
Default configuration: all switches open.
Settings effective: only when the detector is first switched on.

Switch 1
- Open = Amplitude recovery on (default)
- Closed = Amplitude recovery off

Switch 2 & 3
- 2 Open, 3 Open = Divide by 23 (default)
- 2 Open, 3 Closed = Divide by 17
- 2 Closed, 3 Open = Divide by 16
- 2 Closed, 3 Closed = Divide by 31

Switch 4 & 5
- 4 Open, 5 Open = Low frequency cut off 25kHz (default)
- 4 Open, 5 Closed = Low frequency cut off 18kHz
- 4 Closed, 5 Open = Low frequency cut off 15kHz
- 4 Closed, 5 Closed = Low frequency cut off 8kHz

Switch 6 & 7
- 6 Open, 7 Open = Amplifier off after 1 second and a long signal validity check (default)
- 6 Closed, 7 Open = Amplifier off after 5 seconds and a short signal validity check
- 6 Open, 7 Closed = Amplifier off permanently (for 'Line-out' recording)
- 6 Closed, 7 Closed = Hardware and software test

Switch 1 disables the amplitude recovery and all bat calls will be reproduced at full amplitude.
Switches 2 and 3 set the division ratio and also the number of steps in the generated sinewave. The higher the division ratio, the lower the frequency of the resulting audio. Also, the higher the division ratio, the higher the accuracy of the synthesized sinewave. This is because higher division ratios allow more discrete points to be defined within each sinewave cycle.
A division ratio of 16 is provided for compatibility with other frequency division detectors. The other division ratios are prime numbers as this minimizes the chance of unwanted feedback.
Switches 4 and 5 set the low frequency cut off threshold.
Switches 6 and 7: when switch 7 is open, switch 6 allows a choice of sensitivity settings.
When switch 6 is open, the amplifier switches off after 1 second of inactivity and 5 cycles of a bat's call have to be received before the amplifier is switched back on. The TDA8541 amplifier introduces a short delay between the amplifier switching on and audio being produced to prevent a switch-on click, so the beginning of some calls may be lost.
When switch 6 is closed, the amplifier remains on for 5 seconds and only 3 cycles of the bat's call are required to activate the amplifier. The detector is thus in the 'ready to go' state for much longer. This mode increases battery drain and is more susceptible to false triggering.
When switch 7 is closed, switch 6 takes on different functions. If switch 6 is open, the amplifier is permanently switched off. This is used if the optional Line-out jack socket is fitted and the bat calls are being recorded and the internal speaker is not required.
When both switches 6 and 7 are closed, the software performs a test routine. The LEDs are flashed and a continuous tone is generated which is optionally modulated by the microphone input depending upon the setting of switch 1.

Component List Main Board

Resistors
R1,R2,R24,R29,R34 = 47kΩ 5%, 0.25W, 250V
R3,R30,R32 = 4.7 kΩ 5%, 0.25W, 250V
R4,R7,R10 = 7.5kΩ 1%, 0.6W,350V
R5,R8,R11,R13,R14,R16-R18,R20,R21,R25-R28,R33 = 10kΩ 5%, 0.25W, 250V
RR6,R9,R22 = 100kΩ, 5%, 0.25W, 250V
R12 = 82kΩ 5%, 0.25W, 250V
R15 = 30kΩ 5 %, 0.33W, 250V
R19 = 36kΩ 5 %, 0.33W, 250V
R23 = 1MΩ 5%, 0.25W, 250V
R31 = 39kΩ 5%, 0.25W, 250V
R35 = 100Ω 5%, 0.25W, 250V
P1 = 50kΩ 10%, 0.5W, trimmer, 23-turn
P2 = 100kΩ 10%, 0.5W, trimmer, 23-turn
P3 = 10kΩ 20%, 0.2W, rotary potentiometer, logarithmic law, with switch, single

Capacitors
C1 = 220µF 20%, 50V, 5mm pitch, 10x16 mm
C2,C3,C6,C7,C8,C11,C22,C23 = 100nF 10%, 50V, X7R, 0.2'' pitch
C4 = 100µF, 20%, 50V, 3.5mm pitch, 8x11 mm
C5,C9,C10,C24 = 10µF 20%, 50V, 2mm pitch, 5x11 mm
C12,C14,C16,C21 = 1nF 5%, 100V, C0G/NP0, 0.2'' pitch
C13,C15,C17 = 4.7pF ±2.5pF, 500V, NP0, 5.08mm pitch
C18 = 470pF 5%, 100V, C0G/NP0, 0.2'' pitch
C19,C20,C25,C26,C28 = 10nF, 10%, 100V, X7R, 0.2'' pitch
C27 = 47µF, 50V, 2.5mm pitch, 6.3x11 mm

Semiconductors
D1,D2 = BAT43, DO-35
D3 = BZX79-C3V0, 3V zener diode, 0.5W, DO-35
LED1 = blue, 3mm, high intensity
LED2 = red, 3mm, high intensity
T1,T2,T4,T5 = BC548B
T3 = BC558B
IC1 = TL061CP, DIP-8
IC2,IC3,IC4 = TL062CP, DIP-8
IC5 = LP2950ACZ-5.0, LDO, 5V, 0.1A, TO-92
IC6 = PIC16F1827-I/P, DIP-18, programmed, Elektor Store # 150346-41
IC7 = TDA8541T/N1, SMD SOIC-8, Newark/Farnell # 1854043

Miscellaneous
K1 = 4-pin pinheader, vertical, 0.1'' pitch
K2,LED1,LED2,LS1 = Pin header, 1x2, vertical, 0.1'' pitch
BT1 = 2-way PCB screw terminal block 0.2'' mm, 630V
TP1,TP2,TP3 = 1 pin from pinheader
S1 = 8-way DIP switch block
BT1 = battery retainer clip (for battery PP3, 9V) with wires
Enclosure, ABS, 40x65x120mm, Camdenboss BIM2004/14-BLK/BLK, Newark/Farnell # 2445837
LS1 = miniature loudspeaker, 8Ω, >0.5W
MIC. = phone audio connector, socket, 3.5mm, 3 contacts, panel mount
PCB, Elektor Store # 150346-1
Kit of parts incl. PCB, case, and assembled MEMS microphone board 150346-91; Elektor Store # 150346-71

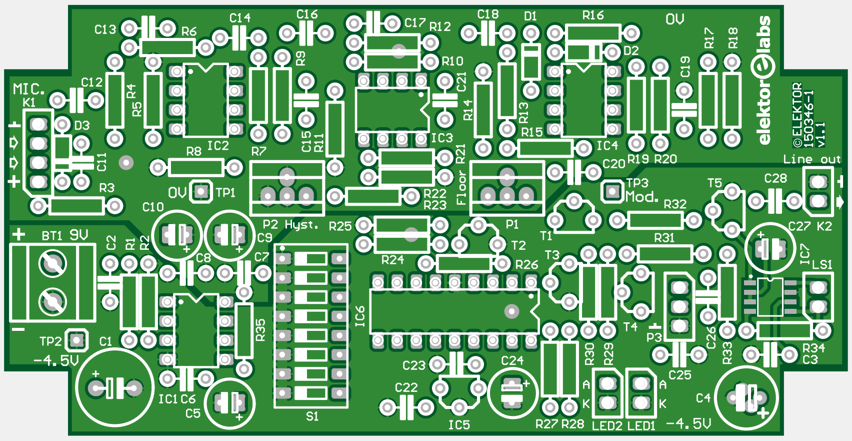
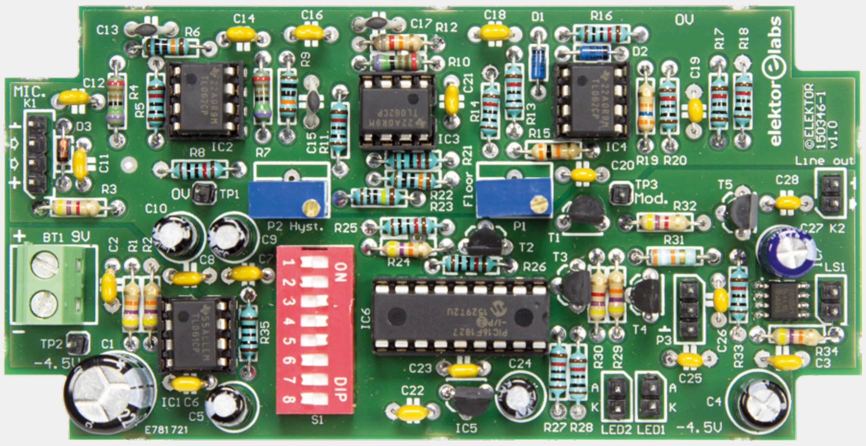


Figure 4. Printed circuit board designed by Elektor Labs for the Bat Detector with Amplitude Recovery.



correct way round. The trimpots P1 and P2 are vertical types, i.e. with all three pins in-line.

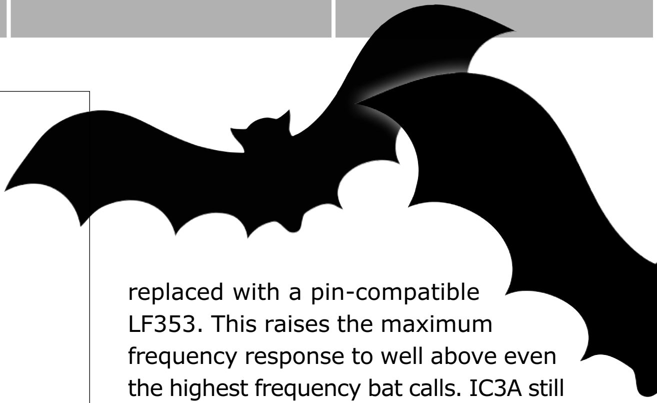
Microphones and gain considerations

The selection of a microphone is probably the most difficult part of the project. A MEMS microphone requires a separate power supply whilst a '2-pin' electret has a common power supply and audio output. The various options are shown in **Figure 6**.

The microphone of choice is a MEMS microphone (**Figure 6a**), although they are extremely small and thus difficult to mount. MEMS stands for Micro Electro-Mechanical Systems. A Knowles type SPU0410HR5H–PB has been found to work well. Many other types of MEMS microphones should also work well as they tend to be extremely responsive to ultrasound. Knowles have recently released a MEMS microphone specifically for ultrasound, the SPH0641LU4H-1. Unfortunately, at the time of writing, it has been impossible to obtain one (we got one, *Labs*).

The other choice of microphone is an electret (ECM) insert (**Figures 6b, 6c**). Both MCE-4000 and EK3132 electret microphones have been found to work reasonably well and an electret from a discarded cordless 'phone also gave good results. However, many other electret microphones were found to be very unresponsive at ultrasonic frequencies.

The tested electret microphones were found to have lower output at ultrasonic frequencies than MEMS microphone and thus feedback resistors R6, R9 and R12 may need to be increased to 120 kΩ or even 150 kΩ to provide additional gain. The latter value reduces the theoretical upper frequency limit of the TL062CP operational amplifiers to around 50 kHz. This may be too low for some species of bat. If this is a problem, IC2 can be



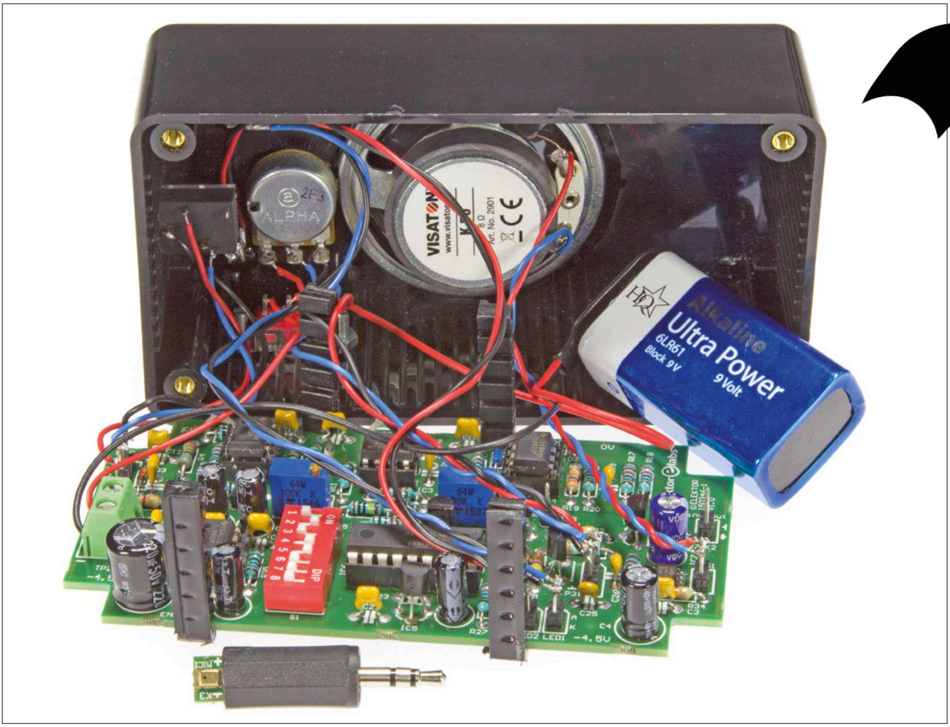


Figure 5. The case, opened to show wiring connected.

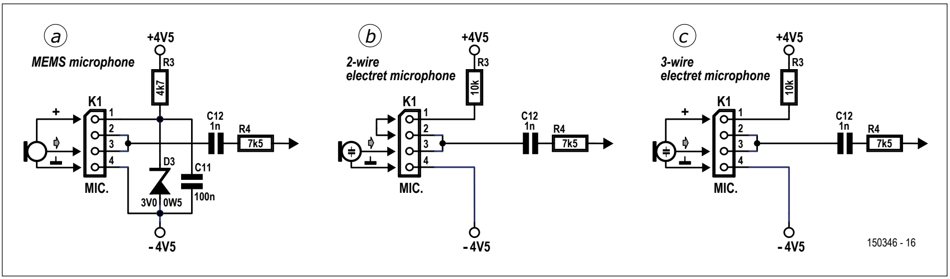


Figure 6. You have three options for the microphone:
MEMS (a; preferred…), 2-wire electret (b); or 3-wire electret (c).

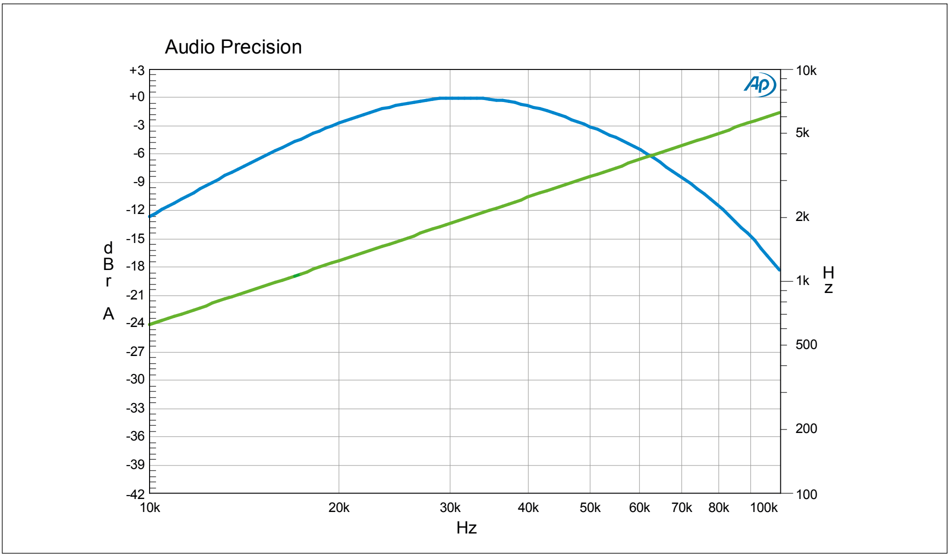


Figure 7. Frequency response graphs obtained with TL062CP fitted (blue) or LF353 fitted (green). The 100-plus kHz response is only needed for certain species of "high-band" bats.

replaced with a pin-compatible LF353. This raises the maximum frequency response to well above even the highest frequency bat calls. IC3A still attenuates the higher frequencies, but not to a significant amount. The downside of an LF353 is that it increases the current consumption of the circuit.

The graph in **Figure 7** shows the frequency response of the circuit recorded on Elektor labs' Audio Precision analyzer, with a TL062CP (blue) *versus* an LF353 (green) fitted in position IC2. It should be noted that the graph was produced by injecting a signal directly into IC2A and thus does not take into account the characteristics of the microphone which will have a significant effect on frequency response.

The higher gain needed for electret microphones can lead to unwanted feedback modulating the generated sinewave at higher output volumes. This is especially noticeable if R6, R9 and R12 are replaced with 150-kΩ resistors. If using an electret microphone, it is recommended that the volume be kept fairly low or that headphones are used.

The PCB has a 4-pin connector to ease the connection of the various microphone options.

MEMS microphone board

A mini board was designed especially for the MEMS microphone (if you can get one), see the schematic in **Figure 8** and the board layout in **Figure 9**. The board saves you the job of keeping the stray capacitances around the device in check. The little board is available ready-assembled and tested from the Elektor Store (#150346-91) and can be fitted into a 3.5-mm stereo jack plug as shown in **Figure 10**.

Adjustment

Carefully inspect the PCB for any solder bridges. Set all switches to the Open position and then attach a 9-V battery and switch on. Connect a multimeter between TP (test point) 1 and TP2. TP1 should be at approximately half battery voltage. One or both of the LEDs should light. Jangling metal objects, such as a set of

keys, in front of the microphone should produce a sound in the speaker.
Trimpots P1 and P2 now need to be adjusted. This needs to be done in a quiet environment, well away from any ultrasonic signal sources. Be aware that many electrical and electronic items may be generating ultrasound.
The first task is to adjust P1 — which sets the audio level — for the quietest bat calls. Attach a multimeter between TP2 (battery minus) and TP3 (Mod.). Adjust P1 until you have a reading of 0.3 volts. This is best done with a used, but good, battery as the voltage will vary slightly as the battery voltage drops with use. This is an initial setting, and P1 can be slightly adjusted to suit individual requirements later.
The final adjustment is P2 which sets the hysteresis for the Schmitt trigger. This is best done with DIP switch 6 in the Open position. Adjust P2 to the point where the blue SIGNAL LED just goes out. If the LED flashes very occasionally, this is not a problem. Jangle a set of keys in front of the microphone and the red MUTE LED should go out and the blue SIGNAL LED come on. When you stop jangling the keys, the SIGNAL LED should go out and the MUTE LED should come back on after a second's delay as the amplifier is switched into standby. Adjust P2 until this happens cleanly every time the input signal stops.
The setup is now complete.

Test mode

If DIP switches 6 and 7 are both closed, the software enters the test mode when the detector is switched on. Both LEDs are flashed and with a division ratio of 23, a 1-kHz tone is generated. Different division ratios generate correspondingly different frequencies. If switch 1 'Amplitude Recovery' is open, the tone will be modulated by the microphone input. In the absence of a signal, the tone will be quite low. This can be used adjust P1 to suit operating conditions so that distant bats can just be heard. If the microphone picks up a signal, the volume of the tone will increase.
If switch 1 'Amplitude Recovery' is closed, the tone will be output at full volume.

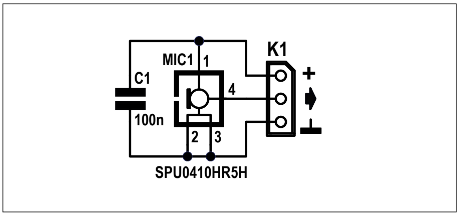


Figure 8. SPU0410HR5H MEMS microphone circuit.

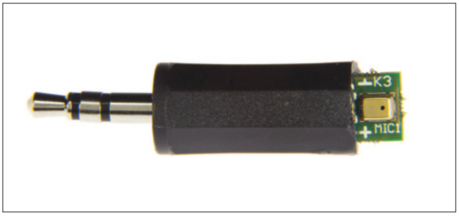


Figure 10. The MEMS microphone can be fitted into the sleeve of a 3.5-mm jack plug.

Component List MEMS Microphone Board

Capacitor
C1 = 100nF 5%, 25V, C0G/NP0, SMD 1206, optional

Miscellaneous
MIC1 = MEMS Microphone SPU0410HR5H-PB (Knowles), Mouser # 721-SPU0410HR5H-PB
K1 = connections for wires or pinheader
Phone Audio Connector, Plug, 3.5mm, 3 contacts, Cable Mount

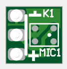


Figure 9. Miniature PCB especially for the MEMS microphone option.

PCB, Elektor Store # 150346-2
Ready assembled MEMS microphone board, Elektor Store # 150346-91

Operation

Operation is very straightforward. Firstly set the DIP switches to the desired settings (**inset**). The detector is then turned On by the volume control, and the volume set.
Feedback from the speaker to the microphone is always a potential problem with bat detectors. This should not be a major problem with the recommended MEMS microphone, but as mentioned previously, the higher gain needed for electret microphones can lead to unwanted feedback which modulates the generated sinewave. The volume should be kept to modest levels or headphones should be used with electret microphones.
One or both LEDs will be on at all times when the detector is switched on, so there is always a visual indication that the detector is on. The LEDs have the following meaning:

Mute LED On, Signal LED Off

The detector is in standby mode waiting for a signal. The amplifier is off to conserve the battery.

Mute LED On, Signal LED On or flashing

A signal is being detected but the software has not recognized it as a bat. The amplifier is off to conserve the battery.

Mute LED Off, Signal LED On

A bat has been detected. Its call should be audible through the speaker. The signal LED will remain lit after the call has finished and will be switched off as the detector goes back into standby.

If the optional Line-out jack socket has been fitted and the detector is connected to a recorder, the internal amplifier can be permanently switched off be setting switch 6 Open and switch 7 Closed.
It should be noted that the switch settings are only read when the detector is first switched on. If you change the switch settings, you will need to switch the detector off and on again to obtain the new settings.
The following traces show a typical bat call; a succession of short clicks. The second graph expands one of the clicks to show the detail.

Making modifications

Depending upon the microphone used, it may be necessary to adjust the overall gain of the operational amplifiers IC2 and IC3A. For a MEMS microphone, the gain can be increased by replacing R12 with 100 kΩ or reduced by making one, or both, of R6 and R9 82 kΩ.
As previously discussed, R6, R9 and R12 will all need to be adjusted for an electret microphone. Try to keep them all of similar value if possible. Increasing the value above 150 kΩ is not recommended and will reduce the bandwidth of the amplifiers. As previously mentioned, IC2 can be replaced with an LF353 to improve bandwidth.
The LEDs have 10-kΩ current limiting resistors to save power. Should brighter

LEDs be required, their value can be reduced, but remember you are likely to be using the detector in the dark.
R19 sets the amount of gain for the amplitude and sets the audio output level for strong signals. It is not recommended that this be changed, but higher values will increase the gain whilst lower values will reduce it.

A number of parameters can be changed in the software [3] such as the low frequency cut-off, division ratios, amplifier timeout and the threshold for validating that the input signal is a bat. These parameters are all grouped together as #define statements for convenience. The software also includes additional sinewave tables for divide by 13, 19, 29, and 37 should the user wish to experiment.
The timing loop within the program has been carefully crafted, and should not be modified (**Listing 2**).
Figure 11 finally shows my suggestion for 'artworking' the Bat Detector case.

My thanks go to John Errington for his original design concept and the encouragement and prompting he has given me to keep improving the design.

(150346)

Web Links

[1] Durham Bat Group: www.durhambats.org.uk/bat_calls.htm

[2] Types of bat detector: www.skillbank.co.uk/bat_detectors/index.html

[3] Project support page: www.elektormagazine.com/150346

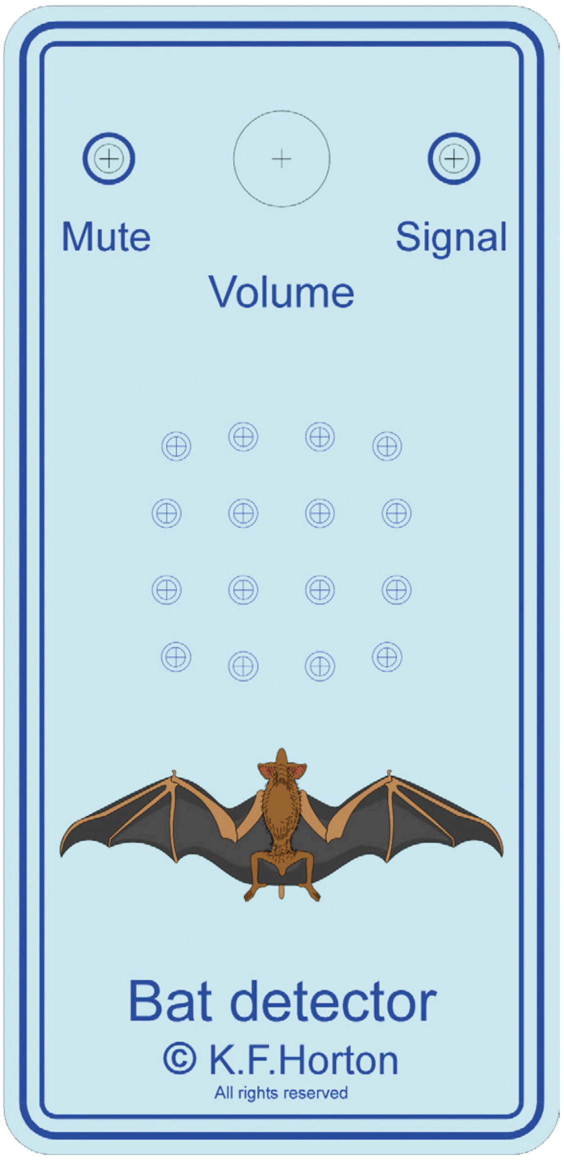


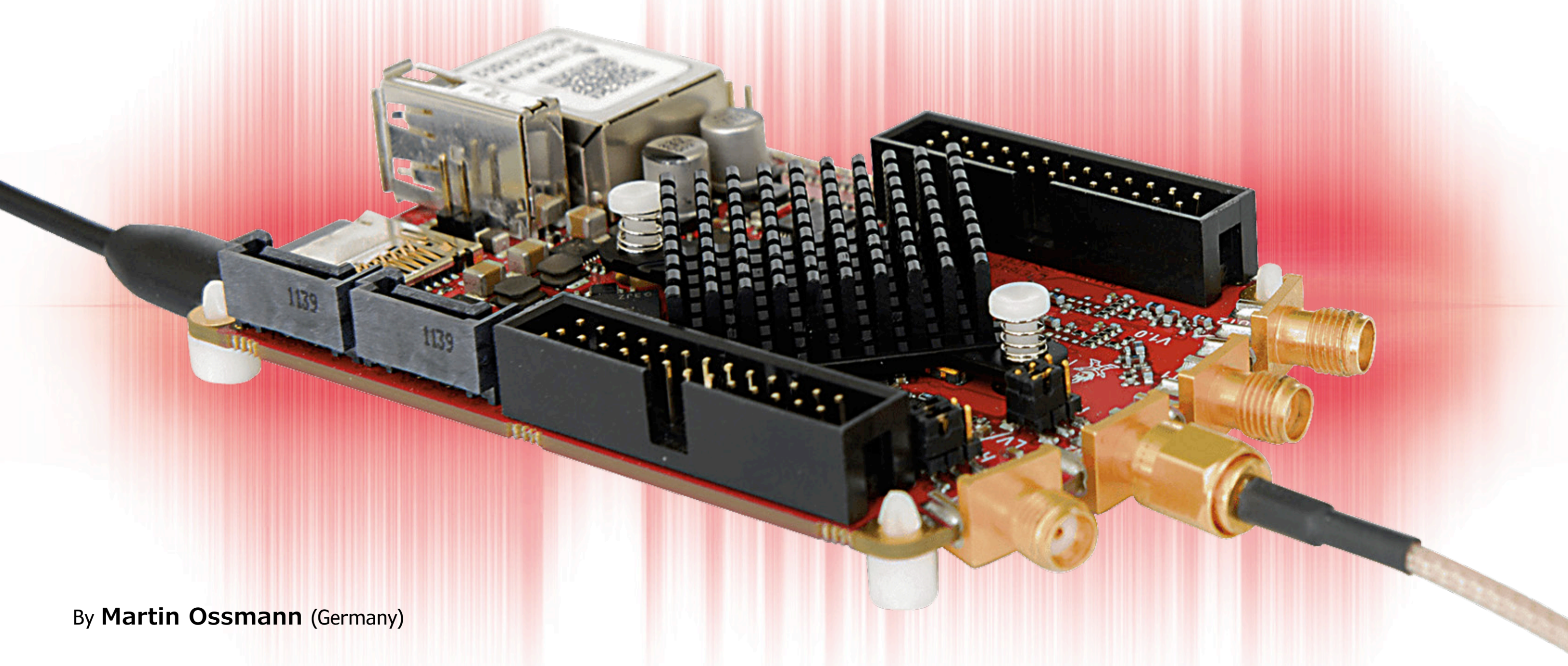
Figure 11. Suggested artwork for affixing to the instrument cover. Feel free to add your own inspiration. Actual size: 152 x 72 mm.

Listing 2. #defines for the timing loop.

```
#define Quick_amp_off .10   ; Delay before switching off amp in 1/10 seconds - switch 6 high
#define Slow_amp_off  .50   ; Delay before switching off amp in 1/10 seconds - switch 6 low

#define   Long_Valid_count   .5    ; Number of valid input triggers before taking action - switch 6 high
#define   Short_Valid_count   .3   ; Number of valid input triggers before taking action - switch 6 low

#define   ratio_a .23   ; Division ratio - switch 2 high & 3 high (13,16,17,19,23,29,31,37)
#define   ratio_b .17   ; Division ratio - switch 2 high & 3 low (13,16,17,19,23,29,31,37)
#define   ratio_c .16   ; Division ratio - switch 2 low & 3 high (13,16,17,19,23,29,31,37)
#define   ratio_d .31   ; Division ratio - switch 2 low & 3 low (13,16,17,19,23,29,31,37)

#define   LF_cutoff_a   .25000    ; Low frequency cut off - switch 4 high & 5 high
#define LF_cutoff_b   .18000    ; Low frequency cut off - switch 4 high & 5 low
#define   LF_cutoff_c   .15000    ; Low frequency cut off - switch 4 low & 5 high
#define   LF_cutoff_d   null    ; Preset at approximately 8 kHz - switch 4 low & 5 low
```

Filtering on the Red Pitaya

Part 1: FIR filters



By **Martin Ossmann** (Germany)

When FPGAs are used in signal processing applications, digital filtering is one of the most important standard operations. Previously in Elektor we have looked at the relatively simple CIC (cascaded integrator-comb) filters for decimation and interpolation [1]. Now we look at filters whose behavior can be specified more closely.

In this first installment we will look at FIR (finite impulse response) filters; the second part will deal with IIR (infinite impulse response) filters. For both types of filter we have the same two problems to solve:

- How do we implement the filter in an FPGA?
- How do we determine the parameters of the filter so that we realize the type of response (say lowpass or bandpass) that we want?

The second problem can be solved with the help of the ARM CPU on the Red Pitaya board, which is easily capable of running simple filter design software. The Red Pitaya can therefore work autonomously both generating filters on the ARM CPU and implementing them at a high sample rate (125 megasample/s) on the FPGA.

FIR filters

The structure of an FIR filter is shown in **Figure 1**. There are three ingredients. The blocks labeled 'z^{-1}' store a signal and delay it by one sample period T_S ($T_S = 1/F_S$ where F_S is the sample rate). In the FPGA such a block is usually implemented as a register whose width is equal to the word size of the signal to be delayed: in this case this is the input signal to the filter. Then we come to the adders (circles marked with plus signs), which add two signals together. These are implemented as binary adders in the FPGA: often special blocks are available to help build fast adder units.

And finally we have the multipliers, which here are used to multiply a signal by a specified constant. This constant will also be stored in a register so that we can modify the filter coefficients at run-time with the help of the ARM CPU. Again, FPGAs often offer special-purpose blocks to perform multiplications quickly. In the case of the Red Pitaya board the DSP48E1 blocks [2] come in very handy for signal processing applications like this.

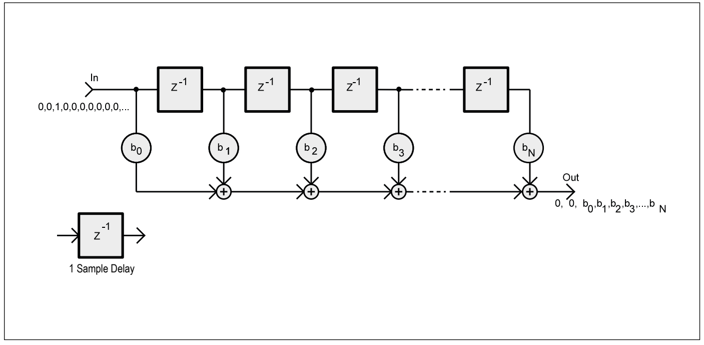


Figure 1. Structure of an FIR filter.

Listing 1. Direct-form implementation in Verilog.

```
1 parameter sumSize = 38 ;
2 parameter sigSize = 14 ;
3 parameter coeffSize = 18 ;
4 parameter nTaps =32 ;
5 reg signed [coeffSize-1: 0] bk[nTaps-1:0] ;
6 wire signed [sumSize-1: 0] sk[nTaps-1:0] ;
7 reg signed [sigSize-1: 0] xk[nTaps-1:0] ;

8 assign sk[0] = bk[0]*xk[0] ;

9 generate genvar m;
10  for(m=1 ; m<nTaps ; m=m+1
11    assign sk[m] = sk[m-1]+bk[m]*xk[m] ;
12 endgenerate

13 integer k ;
14 always @(posedge adc_clk_i) begin
15  if ( firStrobe ) begin
16    xk[0] <= filterIn ;
17    for(k=1 ; k<nTaps ; k=k+1 )
18      xk[k] <= xk[k-1] ;
19    filterOut <= sk[nTaps-1] ;
20  end
21 end
```

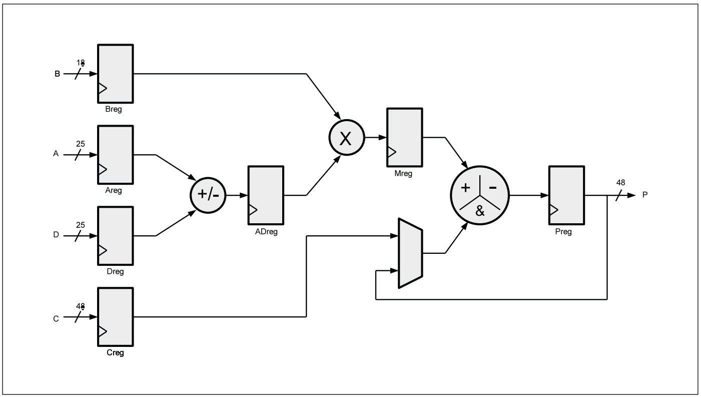
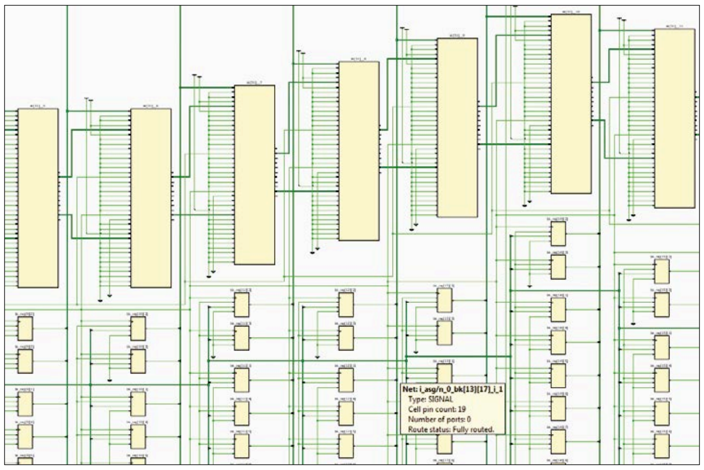


Figure 2. Structure of the DSP slices in the Red Pitaya's FPGA.



Net: i_asg/n_0_bk[13][17]_i_1
Type: SIGNAL
Cell pin count: 19
Number of ports: 0
Route status: Fully routed.

Figure 3. Filter circuit in the FPGA (excerpt).

Xilinx DSP48E1 DSP slices

The FPGA in the Red Pitaya contains DSP blocks which are specially designed for use in digital signal processing applications. Their structure is shown (in simplified form) in **Figure 2**. The Zynq device in the Red Pitaya contains 80 such blocks; larger FPGAs can have as many as 10 000.

The first thing we see in the signal path is a row of registers. These can be used, for example, when it is desired to pipeline operations for increased speed; if they are not required they can be individually disabled. Next to these registers are the computational units. For example, it is possible to add the two quantities on inputs A and D (both 25 bits wide) together, This 'pre-adder' is provided so that the implementation of symmetric FIR filters can be made more efficient.

Next comes a multiplier which can multiply an 18-bit quantity by a 25-bit quantity to produce a 43-bit result. The multiplier is of course ideal for carrying out the multiplications by the coefficients of an FIR filter. After the filter is a further adder, 48 bits wide, whose first summand is normally the product output of the multiplier. The second operand can either be the C input to the block or fed back from the output register of the block after the multiplier (via Preg, 48 bits wide). As well as adding this unit can also carry out logical operations.

Inferring a DSP slice

The DSP48 slice can therefore be used to implement a range of arithmetic functions, as shown in the following examples (which use Verilog notation).

```
P = A*B
Preg <= Preg+A*B
P = (A+D)*B+C
P = A+C
Preg <= Preg + A
Preg <= Preg+Mreg , Mreg<=Breg*Areg , Breg<=B , Areg<=A
```

The pipeline registers allow several operations to be executed simultaneously, as indicated by the commas in the code above. When compiling an FPGA design the development tool (Vivado in this case) examines the source code, in Verilog or in VHDL, for suitable operation patterns. When it spots a suitable operation it tries to realize it in a DSP slice, as this is usually the most efficient approach. This process of identifying an opportunity to use a DSP slice from patterns in the source code is called 'inference'. The designer does not have to worry about the efficient implementation of the addition and multiplication operations: this is handled automatically. Alternatively it is possible to employ the DSP48 macro explicitly: this is called 'instantiation' as an explicit instance of the DSP48 slice is created. Sometimes this can produce better end results, as the automated inference process does not always find the optimal solution for the design.

Direct-form implementation

As a first step we will implement an FIR filter like the one shown in Figure 1. The Verilog source code is given in **Listing 1**. The operation described in line 11 of the code will be converted into a DSP48 slice, and the x_k registers (see lines 7 and 18)

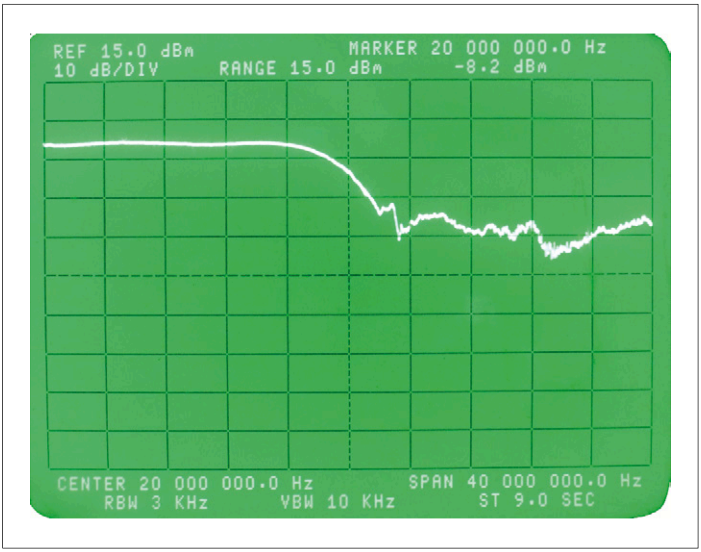


Figure 4. 'Frequency response' of the defective filter.

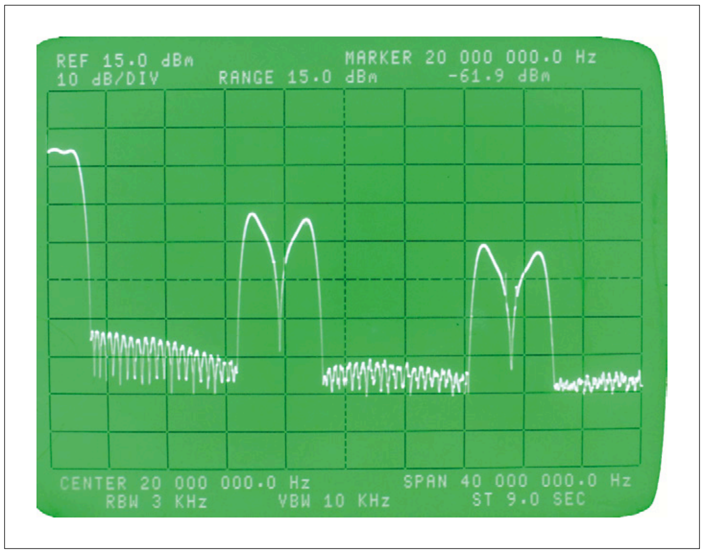


Figure 5. Frequency response at a slower clock rate.

will also be absorbed into the delay registers in the DSP slice. The coefficient registers b_k are realized as flip-flops. If we want to make an FIR filter with 32 coefficients (or 'taps') we will need a total of 406 LUTs (look-up tables), 786 flip-flops and 32 DSP slices. The 32 b_k coefficients alone, stored to 18 bits of precision, require 32*18 = 576 flip-flops. The 32 DSP slices are needed for the multipliers, adders and the x_k registers. The other flip-flops and LUTs are used for the other functions that the module has to carry out.

Figure 3 shows an excerpt of the circuit diagram of the FIR filter module in the FPGA. The large rectangles are the DSP48 slices, which are cascaded. These carry out almost all the work of the filter. Below the DSP slices are the flip-flops that store the b_k coefficients. If we run this circuit at a sample rate of 125 MHz with coefficients designed for a lowpass filter, we might get a frequency response like the one shown in **Figure 4**.

Now this does not look much like the desired frequency response, and so there must be a bug somewhere. If we had taken the time to study the information output by the Vivado IDE more carefully, we would have realized that our design contains 'timing violations': that means that the logic is not fast enough. Using the timing analysis tools we can discover the critical path, which has a delay of 56 ns. Now the maximum delay we can tolerate at a clock frequency of 125 MHz is of course 8 ns = 1/125 MHz, so our circuit is just flat out too slow for our clock rate. The problem is that the adders are all connected in a chain, and so a signal has to pass through 31 adders before reaching a pipeline register, and it is this chain that is too long. We can calculate that each individual adder must have a delay of around 56 ns/32 = 1.75 ns, which is in fact pretty quick. However, our implementation has the same problem as the classical ripple-carry adder: too many stages one after the other.

To verify that we have identified the problem, we can try running the filter at 125 MHz/8 by activating the signal firStrobe (see Listing 1) only once in every eight clock cycles. (When running at 125 MHz this signal was continuously activated.) The resulting frequency response is shown in **Figure 5**.
We must take care when interpreting this graph: the sample rate of the filter is now 125 MHz/8 = 15.625 MHz, and so the frequency response repeats every 15.625 MHz. In the lower range of frequencies (below 10 MHz) we see a clear lowpass response with a cutoff frequency of around 2 MHz and an attenuation of 50 dB in the stopband. Around 15.625 MHz and 31.25 MHz we again see the passband, but modified by the transfer function of the DAC registers which are acting as a hold circuit updated every eight clock cycles.
Now we can show how to implement a faster filter. The first technique is to use pipelining, as used in modern processors. We can break up the critical path by inserting additional registers between the stages, as shown in **Figure 6**.

Now the signal only has to pass through one multiplier and one adder between consecutive clock pulses. This implementation is certainly possible using the DSP48 slices, but we will need an extra 2x32 = 64 registers to hold the intermediate results.

The output signal comes out of the filter 32 clock cycles later; in many applications this will not be a problem, as it is only the frequency response that matters. This filter structure is

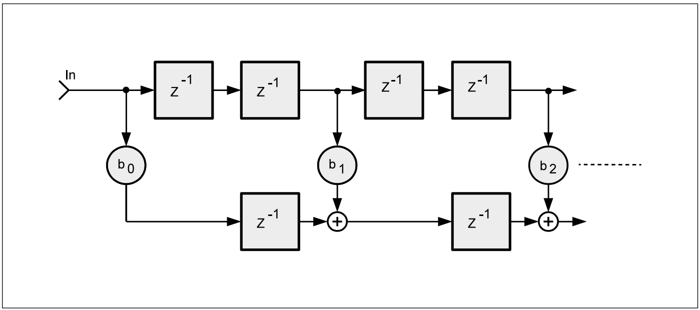


Figure 6. Pipelining a design by inserting intermediate registers.

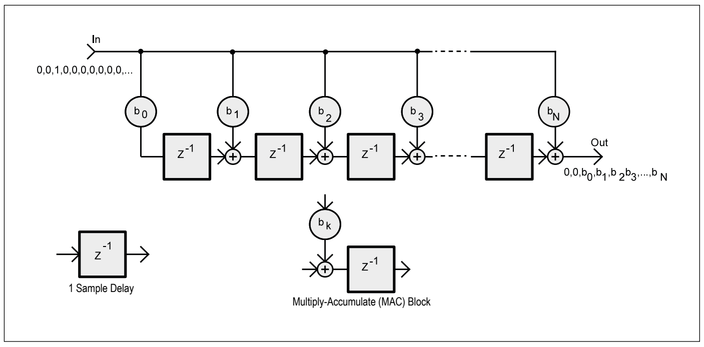


Figure 7. Transposed form of an FIR filter.

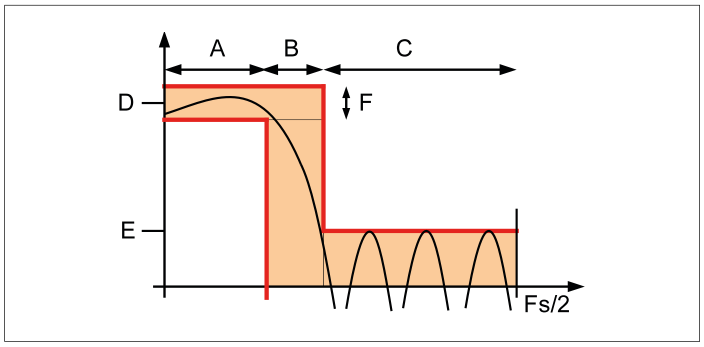


Figure 8. Specification of a lowpass filter.

sometimes called a 'systolic array' or just simply pipelining. But there is a better way.

Transposed-form implementation

With a bit of 'intelligent hindsight' (or even mathematics) it is possible to see that the filter structure shown in **Figure 7** is exactly equivalent to the one shown in Figure 1. This is called the 'transposed form' of the FIR filter.

The critical path now runs from the input through one multiplier and one adder, and so is very short. The structure is particularly suitable for realization in an FPGA. The corresponding Verilog code is along the lines of that shown in **Listing 2**.

The Xilinx Vivado development software is able to see that line 9 can be implemented entirely within a DSP48 slice. Such an operation (of the form $a_i=a+b*c$) is called a 'MAC' (for 'multiply-accumulate'). All modern FPGAs have efficient blocks dedicated to these operations, as they form part of the efficient implementation of a wide range of DSP functions, including filters, FFTs and solving systems of simultaneous equations. For the following we will use this form of the filter with 64 taps. It is of course possible to realize filters with more taps. If we run out of DSP48 slices, multipliers and adders will have to be implemented using (an enormous number of) ordinary logic blocks.

In our filters the coefficients are stored in registers and can be loaded from the ARM CPU in the Red Pitaya. We shall now look at how the coefficient values can be calculated.

Listing 2. Implementation of the transposed form in Verilog.

```
1 parameter accuSize = 38 ;
2 parameter coeffSize = 18 ;
3 parameter nTaps =64 ;
4 reg signed [coeffSize-1: 0] bk[nTaps-1:0] ;
5 reg signed [accuSize-1: 0] sumk[nTaps-1:0] ;
6 integer k ;
7 always @(posedge adc_clk_i) begin
8   for(k=0 ; k<nTaps-1 ; k=k+1 )
9    sumk[k] <= sumk[k+1] + filterIn * bk[k] ;
10   sumk[nTaps-1] <= filterIn * bk[nTaps-1] ;
11   filterOut <= sumk[0] ;
12   end
```

Filter design

Normally a filter design begins with a set of requirements that specify the allowable tolerances on the frequency response. **Figure 8** shows a plot of an example specification for a lowpass filter. The frequency axis is divided into three parts, A, B and C. The passband is region A, the stopband is region C and between them is B, called the transition band.

The desired gain of the filter in the passband is given by D and the minimum attenuation in the stopband by E. Within the passband there is a maximum allowed ripple, denoted by F. The problem now is to design a filter whose response (shown by the black curve) falls entirely within the allowable areas in the plot.

It is usual to seek a filter with the smallest possible order (number of taps) that meets the given specification. Sometimes, however, the order is specified in advance and we look for the filter with the best possible stopband attenuation; and other forms of specification are also possible. Designing high-order FIR filters is not really practical by hand, and so we use special-purpose filter-design software to do the job for us and compute a suitable set of coefficients b_k given the design specification.

The Iowa Hills FIR designer on the PC

A free filter design tool is available from Iowa Hills. **Figure 9** shows its user interface when designing a lowpass FIR filter. The filter in the example shown here has a cutoff frequency of 9.5 MHz at a sample rate of 125 MHz and will be 32 taps long. As can be seen, a stopband attenuation of 40 dB can be achieved. The nulls in the filter response are arranged so that the points of worst-case attenuation in the stopband all reach the same value. This is called an 'equiripple' design; the term is often also used to refer to filters where the passband ripple has constant amplitude. Equiripple filters are often the best that can be achieved with a given filter order. If the transition band is widened it is usually possible to improve the stopband attenuation further. The Iowa Hills design program offers a range of options to help tweak the filter response exactly.

The coefficients can be written to a file which can in turn be loaded into the CPU on the Red Pitaya board. Our filter control program can read the file and load the coefficients into the FPGA. However, this roundabout approach whereby we first design the filter on the PC is not very practical. There is an alternative filter design program, which can run independently on the Red Pitaya.

Parks-McClellan filter design on the Red Pitaya

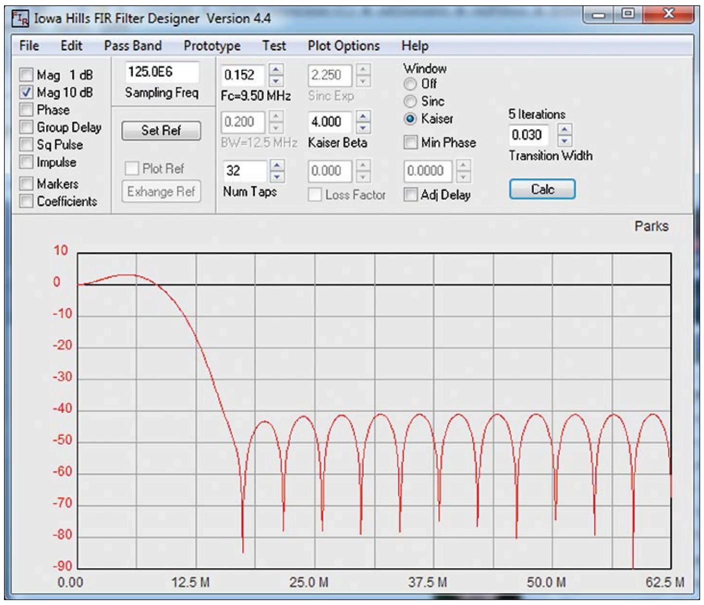
There is any number of sources for equiripple design programs. Most can trace their ancestry back to the original FORTRAN code in [3]. We have adapted the Java version from [4] to make it run on the Red Pitaya; the program lets us design high-, low- and bandpass filters as well as notch (band-stop) filters. **Figure 10** shows an example response curve for a bandpass filter: the ripple in both passband and stopband can clearly be seen.

The filter design program can take the required parameters on its command line. The program then designs the corresponding filter and loads the coefficients into the FPGA. The program also takes care of the scaling of the coefficients so that as much as possible of the full 18-bit range is used.

Figure 11 shows the frequency response of a bandstop filter. The falloff in amplitude at higher frequencies is caused by the analog output filter after the Red Pitaya's DAC.

We have shown in the article how it is possible to design and execute FIR filters on the Red Pitaya board. The filters we have used so far have all been symmetric and hence all have linear phase. It is also possible to use non-linear-phase FIR filters, and there is also the possibility of building IIR filters. How IIR filters can be implemented on the Red Pitaya will be the subject of the second installment of this series. ⏮

(150706)


Iowa Hills FIR Filter Designer Version 4.4

[ ] Mag 1 dB
[✓] Mag 10 dB
[ ] Phase
[ ] Group Delay
[ ] Sq Pulse
[ ] Impulse
[ ] Markers
[ ] Coefficients

Sampling Freq: 125.0E6
0.152 Fc=9.50 MHz
0.200 BW=12.5 MHz
Num Taps: 32
Sinc Exp: 2.250
Kaiser Beta: 4.000
Loss Factor: 0.000
Window: ( ) Off ( ) Sinc (●) Kaiser
[ ] Min Phase
[ ] Adj Delay 0.0000
5 Iterations
Transition Width: 0.030

Figure 9. User interface presented by the Iowa Hills filter design program.

Web Links and References

[1] Enhanced FM Stereo on Red Pitaya, Elektor 5/2015: www.elektormagazine.com/150326

[2] 7 Series DSP48E1 Slice User Guide, Xilinx

[3] T. W. Parks and J. H. McClellan. *Chebyshev approximation for nonrecursive digital filters with linear phase*. IEEE Trans. on Circuit Theory , 19:18994, March 1972, and http://michael-gellis.tripod.com/dsp/pgm21.html

[4] http://afni.nimh.nih.gov/pub/dist/src/FIRdesign.c

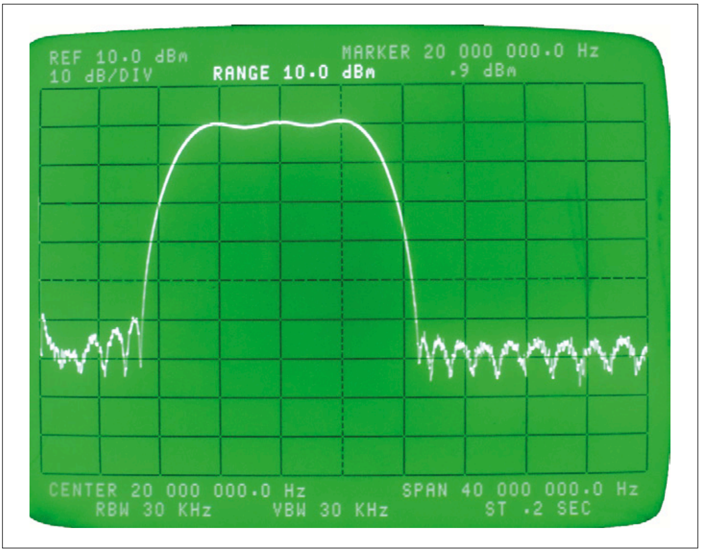


Figure 10. Frequency response of an FIR bandpass filter.

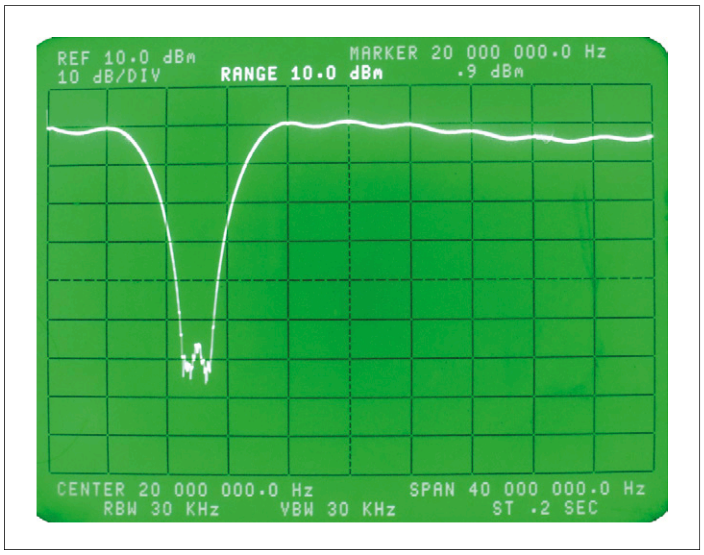
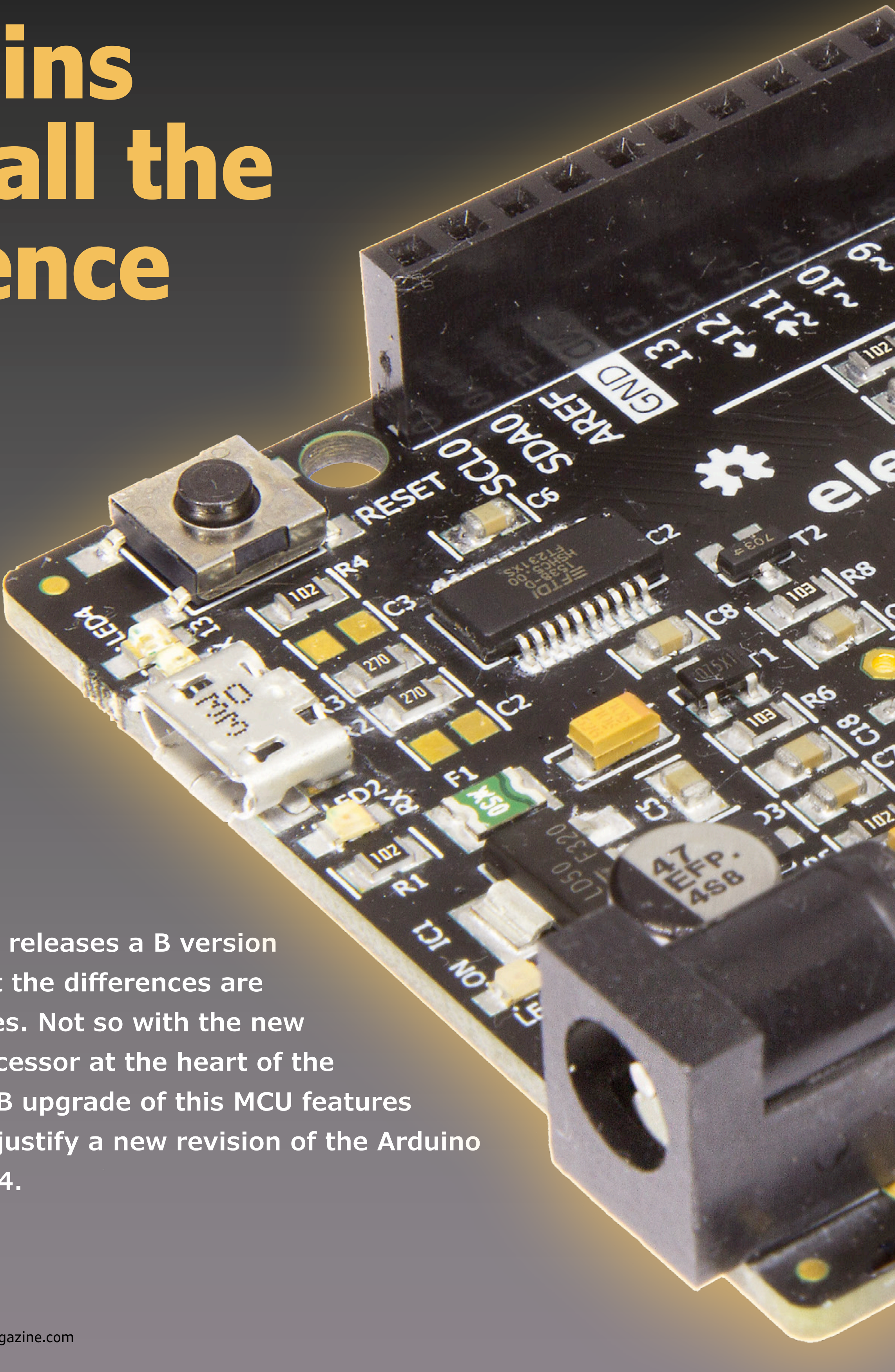


Figure 11. Frequency response of an FIR notch filter.

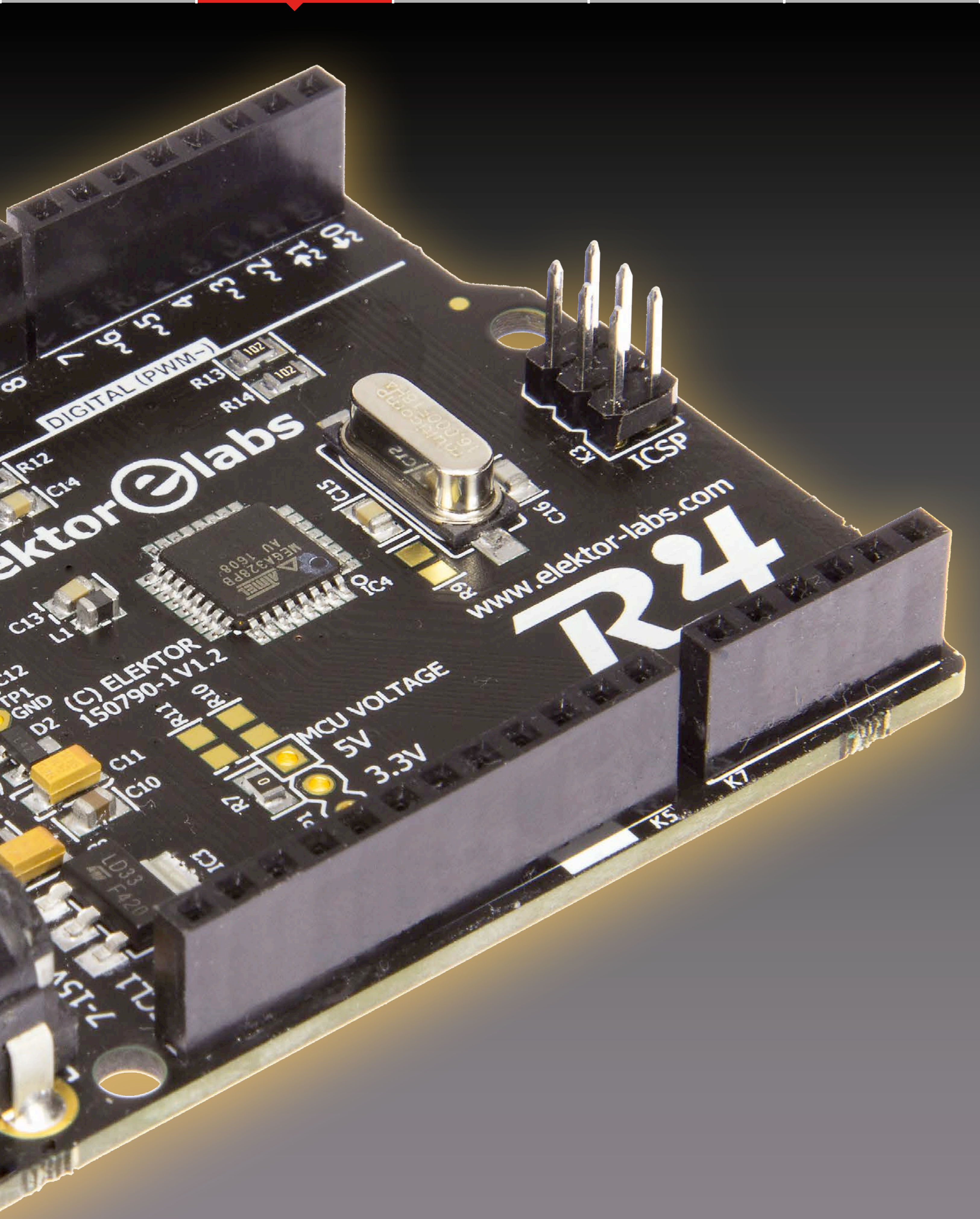
Elektor Uno R4

Four pins make all the difference



By **Clemens Valens** (Elektor.Labs)

When a manufacturer releases a B version of an existing product the differences are marginal in most cases. Not so with the new ATmega328P, the processor at the heart of the Arduino Uno R3. The B upgrade of this MCU features new peripherals that justify a new revision of the Arduino Uno R3. Here is the R4.


DIGITAL (PWM~)
elektor labs
www.elektor-labs.com
R4
ICSP
MCU VOLTAGE
5V
3.3V
(C) ELEKTOR
150790-1V1.2
MEGA328PB

Elektor Uno R4 Key Features

- ATmega328PB @ 16 MHz
- 2 x UART
- 2 x I²C
- 2 x SPI
- 9 PWM outputs
- 8 analog inputs
- 24 GPIO pins
- On-board 5 V and 3.3 V voltage regulators
- Arduino compatible Boards package
- Open source, open hardware design.

From A to B

At the end of 2014 Atmel launched ‘B’ versions of some of their more popular AVR microcontrollers: the ATmega48PB, -88PB and -168PB. Even though the B types are not drop-in replacements for the older models, the differences with the non-B versions were not spectacular. Besides the lower price, the most notice-

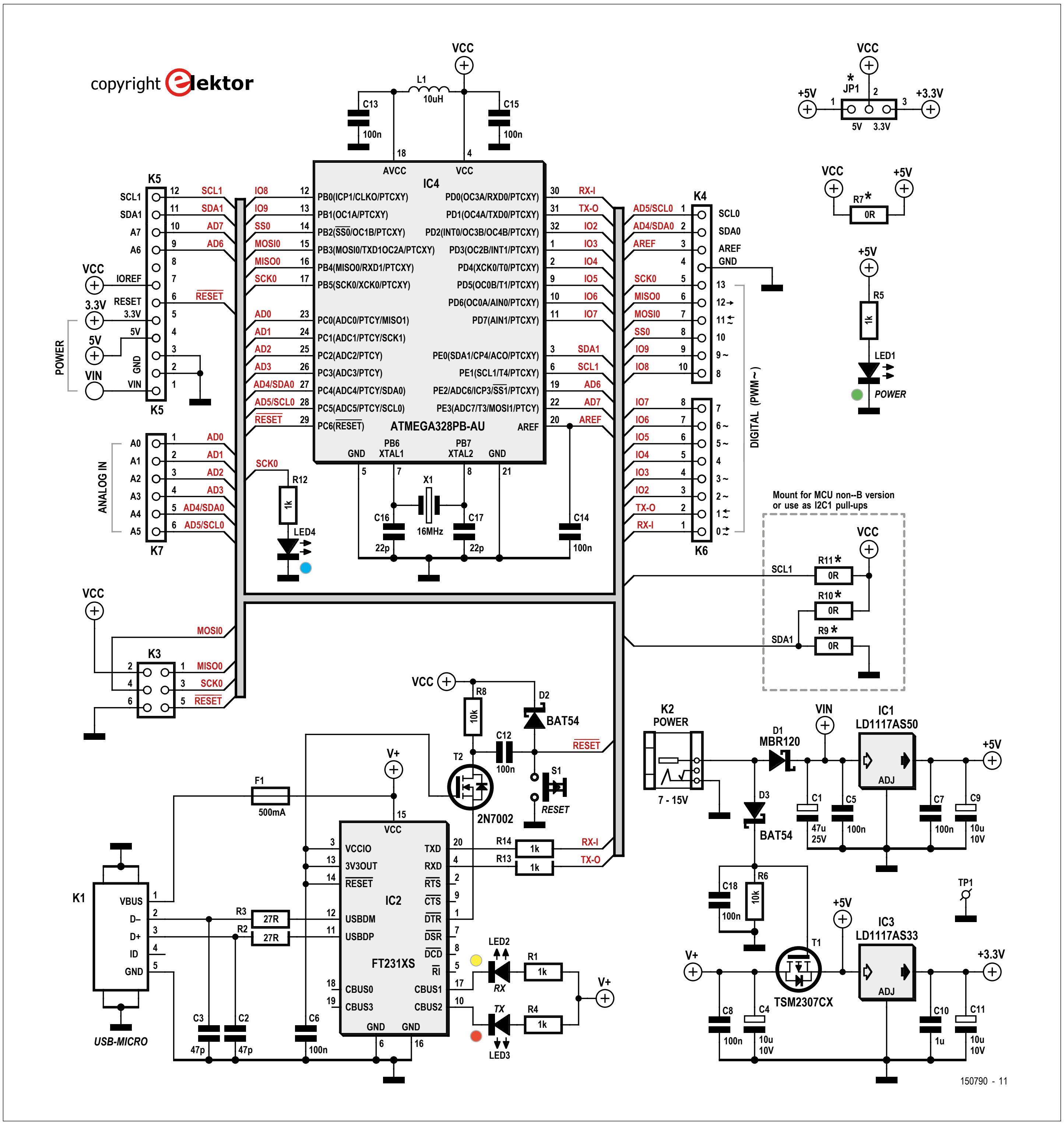


Figure 1. The Elektor Uno R4 is not much more than a breakout board for the MCU with a serial-to-USB converter, a power supply and a few LEDs.

able new feature was the addition of a 4-bit port E, implying that these chips would only be available in 32-pin packages. B-types can run programs compiled for non-B types, but the opposite is not necessarily true. Then some six months later came the ATmega328PB and things got interesting.

Unlike its little B-rothers, the B-version of the ATmega328 really does have more to offer than the non-B (**Table 1**). To start with, the B type is only available in 32-pin packages, hence has four pins extra compared to the classic 28-pin DIP ATmega328 found at the heart of the Arduino Uno. These four extra pins now expose the new GPIO port E. On previous 32-pin packages the extra pins were two extra analog inputs, VCC and GND. As before, the four extra pins also connect to the two additional analog inputs but now they also connect to a second I²C peripheral, to a part of a second SPI peripheral, to the output of the analog comparator and to the outputs of timers 3 and 4.

The changes are not limited to port E and its multiplexed functions. Did I mention the second USART? Yes, there are two of these now. Furthermore, the device has gained two 16-bit timers increasing the number of PWM channels to 10, and it has been equipped with an output compare modulator (OCM) for the easy creation of burst signals. Thanks to the Peripheral Touch Controller (PTC) every I/O pin can now be used as touch-sensitive pin without the addition of external components and a unique ID lets you identify the device uniquely (duh).

Table 1 lists the most important differences between the old and the new versions. Actually, the ATmega328PB is so much different from the non-B type that one wonders why it didn't get its own name.

Take the R4

Now that we know why we would want to use the 328PB, how can we use it? Today, the first answer that comes to mind is: make an Arduino-compatible board for it. And that is exactly what we did. Based on the many sources available on the internet we added some supporting circuitry to the processor and designed a nice PCB for it. To reflect the changes with respect to the Arduino Uno revision 3 (R3) — which our board is backwards compatible with — we decided to dub ours: *Elektor Uno R4*. That's how original we are. Let's run quickly through the schematic (**Figure 1**). The MCU (IC4) has most of its pins connected to extension connectors K4 to K7; after all, it is a kind of break-out board (BoB). K5 is the connector with the four pins of port E allocated and it is now 12 pins long. K3 is the ISP connector usually employed to burn the bootloader into the MCU.

IC2, an FT231XS, provides the USB-to-serial port interface. It is a recent 3.3-volts only chip from FTDI, similar but cheaper than the FT232R. Its two LEDs (LED2: RX, LED3: TX) are powered from the USB bus, so they will light up even if the MCU is not powered. More on that later.

Components T2, R8, C12 and D2 form a level shifter for the Reset signal coming in over the DTR line. D2 clamps high-voltage spikes to VCC. There is no additional

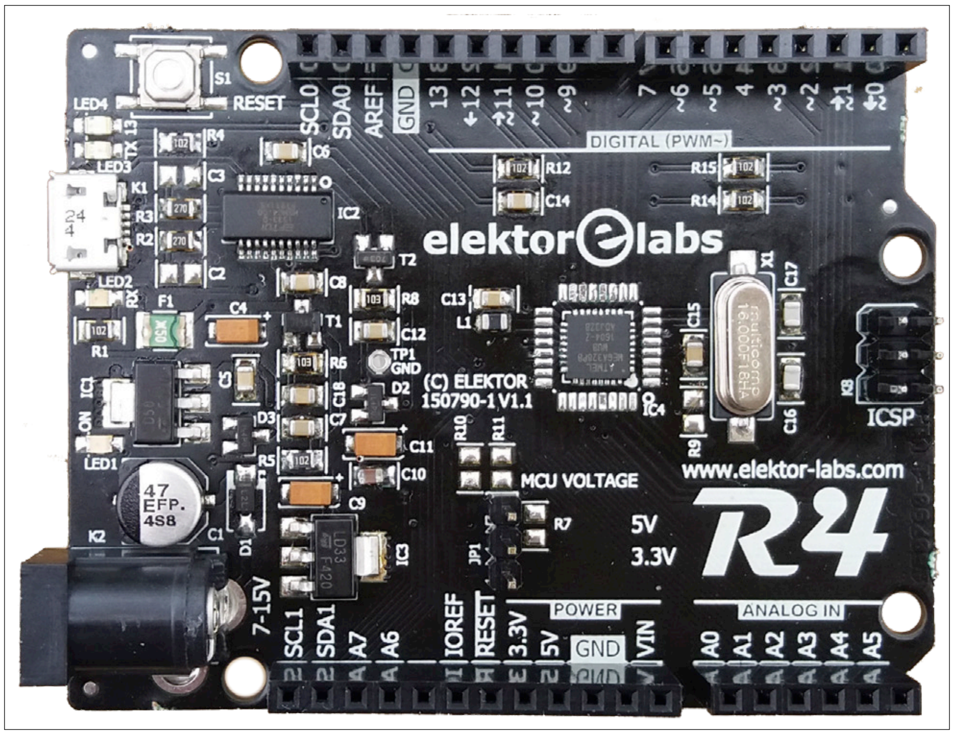


Figure 2. Because the 328PB in TQFP was not available we built one with a QFN package. Note that this is a revision 1.1 prototype.

Table 1. Key differences between the ATmega328 and its B(rother) type.

ATmega328	Normal version	B type
Pins	28 (DIP), 32	32
GPIO	23	27
USART	1	2
I²C	1	2
SPI*	2	4
Peripheral Touch Controller	-	1
Analog channels	6 (8 for 32-pin types)	8
Analog Comparator output	-	1
16-bit Timers	1	3
Output Compare/PWM channels	6	10
Output Compare Modulator	-	1
Input Compare channels	1	3
Unique Device ID	-	1

*The USART is capable of SPI also, which doubles the number of SPI ports.

Figure 3. It's a long and lonesome road to Arduino Inside.

pull-up resistor for pushbutton S1 on the Reset signal because there is one inside the MCU (always been there, BTW).
Then we arrive at the power supply. It consists of a 5-V regulator in case the board is powered by an external power source, followed by a 3.3-V regulator that is necessary for the 3.3-V pin on K5. The 3.3-V regulator is a bit beefier on our board to allow 3.3-V shields to draw some current.
A 5-V switchover circuit has been created with T1, D3, R6 and C18. The maximum gate-source voltage of T1 is specified as 20 V making the construction safe up to 24 V. Nevertheless, we recommend limiting the input voltage to 15 VDC. If, for some strange reason, the USB bus voltage is higher than specified by its standard and the output of IC1 surprisingly low, then the MCU may be powered through the body diode of T1. This may harm the transistor when a lot of current is drawn by the rest of the board, but this is a kind of freak condition that we did not try to prevent.

We added JP1 to allow two things not possible with an Arduino Uno R3:
Power the MCU from 3.3 V or from 5 V
Remove power from the MCU altogether

Shields are supposed to check the IOREF pin on K5 to find out what the supply voltage of the MCU is. On an Arduino Uno R3 it's always 5 V, but on our R4 board it can be 3.3 V too. The jumper can also be used to remove the power from the MCU without disconnecting the USB port, allowing safer rewiring of the extension ports without disconnecting the board from the computer, keeping the serial port open.

R7 fixes JP1 in the 5-V position and it was added for production reasons. On our boards [1] it is mounted, leaving JP1 as an option. Out of the box the R4 has to be R3-compatible, meaning it must run from a 16-MHz crystal. According to the datasheet the MCU cannot run at such a high frequency when its power supply is only 3.3 V. For fully specified operation at 3.3 V replace the crystal by an 8-MHz type (and replace the bootloader).
Finally, a word on resistors R9, R10 and R11. These are optional resistors that make the PCB compatible with the 32-pin ATmega328 (not B). For this processor, pin 3 is supposed to be connected to 0 V, and pin 6 to VCC. R9 and R11 can do that. On the other hand, the B-type has its second I²C port on these pins and this kind of interfaces like pull-up resistors. R10 and R11 can help in that case.

Bad road ahead

Up to here it was smooth ride without potholes and other hiccups, but now we enter the rough terrain where the road kind of stops. From here we must follow a dirt track to a place called Software Integration.

The thing with new processors that come with their own — albeit more or less compatible instruction — sets is that they require programming toolchain support. The toolchain (i.e. compiler, linker, assembler) must be enhanced so that it can use the instruction set of the new processor. The toolchain that comes with the Arduino IDE does not yet "know" about the ATmega328PB. Luckily the IDE has evolved over the past years to allow for boards with custom toolchains, a nice solution for using the IDE with, for instance, ARM-based processors. So there is a way to integrate our board as long as if we bring our own toolchain.

Atmel Studio 7 (AS7) comes with support for the 328PB (a good reason to upgrade to this version if you hadn't done so already). For us this is good news, because now we can figure out what we need and, if possible, steal it from AS7. Like Arduino, AS7 uses the GNU AVR toolchain, but a more recent version. AS7 also has the definitions for the new registers of our new processor, which comes in handy. So, after a bit of research we identified what is needed to compile programs for the 328PB:

- gcc-avr 4.9.2 (or more recent);
- crtatmega328pb.o (from AS7);
- libatmega328pb.a (from AS7);
- specs-atmega328pb (from AS7);
- iom328pb.h (from AS7).

Uploading programs to the board is done with avrdude and this program isn't aware

Arduino IDE 1.6.8 serial port handling

New versions of software are not always better than the version they are supposed to replace. A good example is version 1.6.8 of the Arduino IDE that did not introduce many exciting new features and broke serial port handling. Since the IDE uses the serial port to program the Arduino board and to communicate with it, this is rather inconvenient. When trying to program a board with avrdude this error report may pop up:

```
avrdude: ser_open(): can't open device “\\.\COMxx”: Access is denied.
```

(xx = port number)

Boards relying on Bossac or proprietary serial port uploaders will produce errors also. If you run into this problem, do not panic, your board is fine — just revert to Arduino IDE 1.6.7, the last known good version of the IDE and continue. Hopefully 1.6.9 will be better.

of the 328PB either, so we must fix that too.

More than you care to know about Arduino Boards Packages

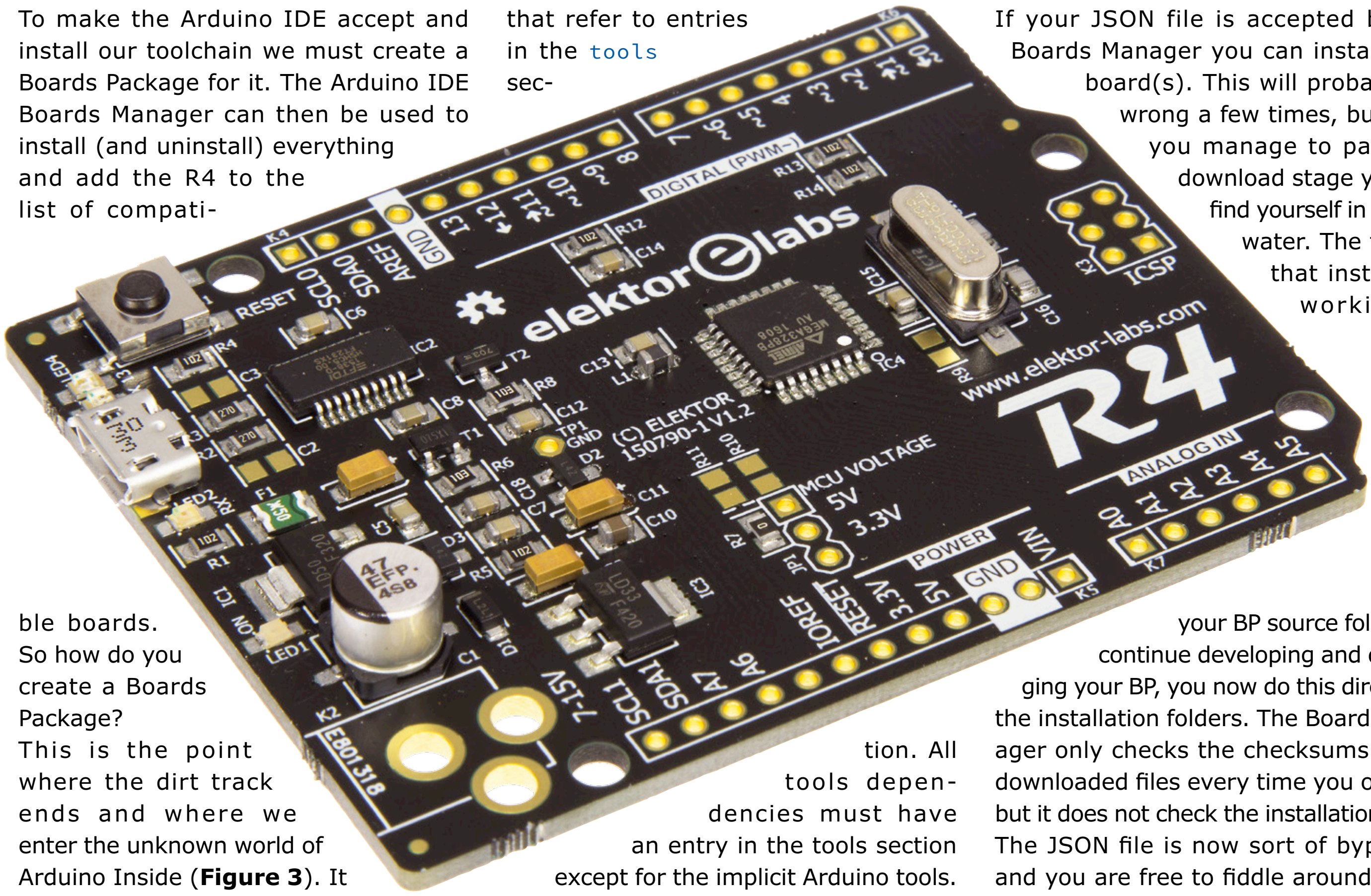
To make the Arduino IDE accept and install our toolchain we must create a Boards Package for it. The Arduino IDE Boards Manager can then be used to install (and uninstall) everything and add the R4 to the list of compatible boards.

So how do you create a Boards Package?

This is the point where the dirt track ends and where we enter the unknown world of Arduino Inside (**Figure 3**). It is not a place where no man has never boldly gone before, but it isn't very crowded either; maps tend to be drawn in white ink. I was here before to create a Boards Package for Platino [2], but this time I had to go much deeper.

A Boards Package (BP from now on) consists of a JSON file and optional archive files to hold toolchains, libraries and other important files that are needed by the board. The JSON file is a complicated beast all by itself, but it is easy-peasy compared to what lies ahead. Inside the JSON file (readable text) we find references to the boards, toolchains, etc. that the BP supports and how they all are related. It also has checksums for the files that make up the BP. The name of the JSON file must be of the form "`package_XXX_index.json`" where XXX is free form text. As an example, our JSON file is called `package_elektor_boards_index.json` [3]. This is one problem of the JSON file, its format is rather rigid and it becomes quickly unreadable due to all the [], {}, and "" and when you miss a comma (or have one too many) fixing the error can be long.

The JSON file has two main sections: `platforms` (the boards) and `tools` (the toolchains). The platforms have sections named `toolsDependencies` that refer to entries in the `tools` section. All tools dependencies must have an entry in the tools section except for the implicit Arduino tools. This illustrates another problem of building BPs: certain things are implicit and others are not, for you to find out which is which. (No wonder there is so little documentation, who would volunteer to describe such a mess and maintain the documentation?)



Inside the JSON file are also the URLs to the files that must be downloaded, their names and sizes and their checksums. Here we find a third obstacle for the creation of BPs: as soon as you change a single character somewhere in a file that is part of the BP, the file's (size and) checksum will change and you must update the JSON file accordingly. I quickly ended up writing a Python script to generate my JSON file automatically.

Once past the JSON file barrier you are expected to enter a link to it in the Arduino IDE, in the Additional Boards Manager URLs box of the Preferences dialog. For a BP end user this is fine, but when developing one you'd better make it point to a file on your computer (start the URL with `file://` instead of `https://`) because the IDE will download it every time you open the Boards Manager. Losing synchronization between what you really have and what you think you have is in that case only a matter of time.

If your JSON file is accepted by the Boards Manager you can install your board(s). This will probably go wrong a few times, but once you manage to pass the download stage you will find yourself in calmer water. The trick is that instead of working in your BP source folders to continue developing and debugging your BP, you now do this directly in the installation folders. The Boards Manager only checks the checksums of the downloaded files every time you open it, but it does not check the installation itself. The JSON file is now sort of bypassed and you are free to fiddle around in the BP installation. When the BP is exactly as you want, you zip it up, update the JSON file and publish it all on a server.

On Windows the downloaded BP files are stored in

```
<user>\AppData\Local\Arduino15\
  staging\packages\
```

whereas the installed BP is in

```
<user>\AppData\Local\Arduino15\
  packages\
```

(Replace `<user>` by the path to `AppData` on your computer.)

Note that this also offers a quick means of testing (large) BP files: make sure the JSON file is correct and then copy the new files to the staging directory instead of uploading them to a server before downloading them again through the Boards Manager.

A BP can consist of several folders. Currently I only know of the `hardware` and the `tools` folder but there may be others. The `hardware` folder contains the files that are to be used by the IDE instead of the files

Component List

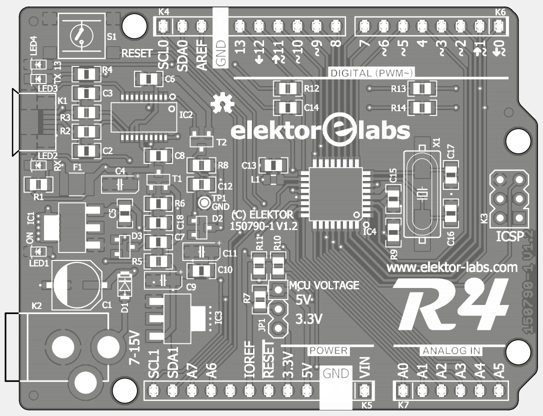
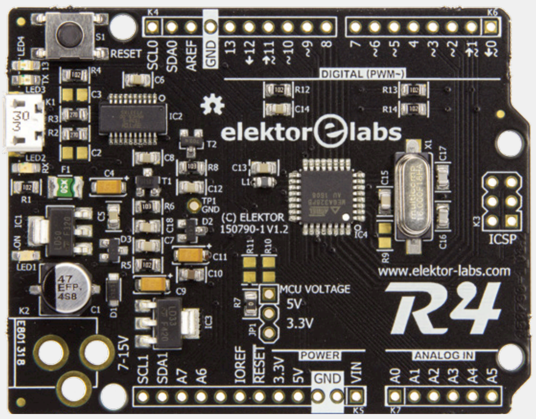
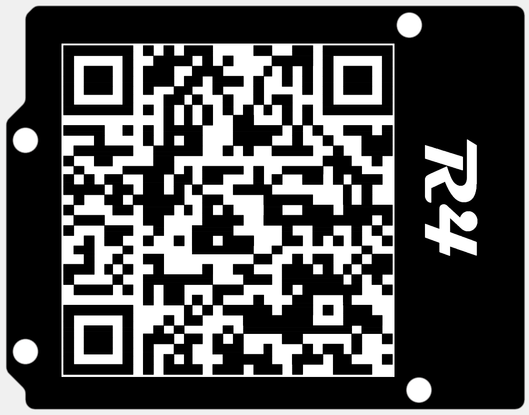
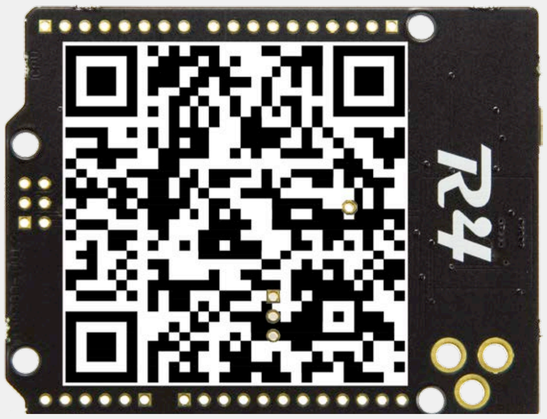
Resistors
Default: 0805
R7 = 0Ω
R2,R3 = 27Ω
R1,R4,R5,R12,R13,R14 = 1kΩ
R6,R8 = 10kΩ
R9,R10,R11 = 0Ω

Capacitors
Default: 0805
C16,C17 = 22pF
C2,C3 = 47pF
C5,C6,C7,C8,C12,C13,C14,C15,C18 = 100nF
C10 = 1µF
C4,C9,C11 = 10µF, 10V, tantalum, case A
C1 = 47µF, 25V, electrolytic, 6.3mm diam.

Inductor
L1 = 10 µH, SMD 0805

Semiconductors
D1 = MBR120
D2, D3 = BAT54
IC1 = LD1117AS50
IC2 = FT231XS
IC3 = LD1117AS33
IC4 = ATmega328PB-AU
LED1 = green
LED2 = yellow
LED3 = red
LED4 = blue
T1 = TSM2307CX
T2 = 2N7002

Miscellaneous
F1 = PTC, 500 mA
JP1 = 3-pin pinheader, 0.1'' pitch, right angled
Jumper for JP1
K1 = Micro USB type B receptacle
K2 = barrel jack, 1.95mm center pin
K5 = 12-way pinheader socket, 0.1'' pitch
K4 = 10-way pinheader socket, 0.1'' pitch
K6 = 8-way pinheader socket, 0.1'' pitch
K7 = 6-way pinheader socket, 0.1'' pitch
K3 = 6-pin (2x3) pinheader, 0.1'' pitch
S1 = tactile switch, 6x6.2 mm, SMD
X1 = 16 MHz quartz crystal
PCB 150790-1 v1.2
Firmware 150790-11

that are in the Arduino IDE's own `hardware/arduino` folder (note the `arduino` subdirectory). The paths are similar but not identical. For example, the BP path to the `bootloaders` folder is `hardware\avr\1.0.0\bootloaders\`, for the IDE it is `hardware\arduino\avr\bootloaders\`. This is valid for all other folders in the BP's hardware folder. The BP's `tools` folder contains the tools that are to be used by the IDE instead its own tools in the `hardware/tools` folder.

Now that we know what goes where we can turn our attention to the `boards.txt` and `platform.txt` files included in the BP's hardware folder. The first one must be written so that the IDE knows what our board is like. This is not too difficult if you base the description on another board that is close to it (the Uno R3 in our case, but with a different processor). There is one detail though that must be handled with care: avrdude. The boards file holds the reference to the sketch upload tool and since the default avrdude does not know the 328PB, we must provide our own tool and make the boards file point to this tool instead.

The `platform.txt` file is more complicated, a real snake pit to be honest, full of undocumented pitfalls, and the goal is to modify it in such a way that the IDE will start using the toolchain we provided instead of its own. This must be done for both the compiler toolchain and the sketch uploader. To cut a long story (too) short: we have succeeded, but it was a hard battle. Please refer to our file to see what we did [3].

Patching avrdude

I like avrdude because of its flexibility. It is the configuration file (usually `avrdude.conf`) that makes this flexibility possible and because of this it is easy to teach avrdude new tricks. For avrdude the 328PB is almost identical to the 328 and 328P, the only thing that changes is the 3-byte device signature and the name. It is therefore enough to copy the section for the 328P in the configuration file and replace 328P by 328PB to obtain this (do not forget the final semicolon):

```
part parent “m328”
id  = “m328pb”;
desc    = “ATmega328PB”;
```

```
signature  = 0x1e 0x95 0x16;
ocdrev     = 1;
;
```

Library and core mods

When you think that you are almost there, you encounter even more software issues. Like the need to provide libraries that support the 328PB's new peripherals and adapt the Arduino core. For the second USART this is easy, as it is already handled by the Arduino core. For the second I²C and SPI peripherals, however, we were forced to change the Arduino Wire and SPI libraries in order to make them peripheral independent. The core we have prepared "knows" about SPI0 & SPI1 and I2C0 & I2C1. These new libraries are of course included in the BP.

The extra timers and PWM outputs are also already supported by the Arduino core, except for the PWM output on PD2. If you look carefully at the schematic, you will see that PD2 is multiplexed with both OC3B and OC4B that are the PWM outputs. This means that although technically the chip disposes of ten PWM channels, on the outside there are only nine. What's more, these two outputs make up the Output Compare Modulator (OCM) and if you activate them both at the same time you will get their signals either AND-ed or OR-ed depending on how you set the level of PD2. If you activate only one and use AND mode, you will not get anything… In order to make PWM work on PD2 we have selected OCM OR mode with OC3B only.

To ensure Arduino Uno R3 compatibility and at the same time create continuity with the two extra analog inputs on PE2 and PE3 they have been mapped to Arduino pins 20 (or A6) and 21 (or A7). The result is that the two remaining pins PE0 and PE1 are on 22 and 23, which is a bit illogical. Since these pins are also connected to the second I²C peripheral, we have preferred to name them respectively SDA1 and SCL1 instead.

Oh, I almost forgot: of course we had to prepare an Arduino bootloader for the 328PB. It is in the BP too.

Conclusion

If you managed to read all the way through this story, you will agree that designing the hardware for this project was the easy part; all the difficulties were lurking in software and file juggling. As this will be more and more the case for future projects, we have decided to spend on ink & paper to describe it. Please accept our sincere apologies for any drowsiness caused.

If you are no more than an end user of the Elektor Uno R4 and its Boards Package please paste this URL:

https://raw.githubusercontent.com/ElektorLabs/arduino/master/package_elektor_boards_index.json

in the Additional Boards Manager URLs box of the Preferences dialog of the Arduino IDE (version 1.6.6 or newer from arduino.cc, do not use 1.7.9 from arduino.org, see **Figure 4**). Open the Boards Manager, select Contributed, install the board and start playing. The future is now yours. ⏮

(150790)

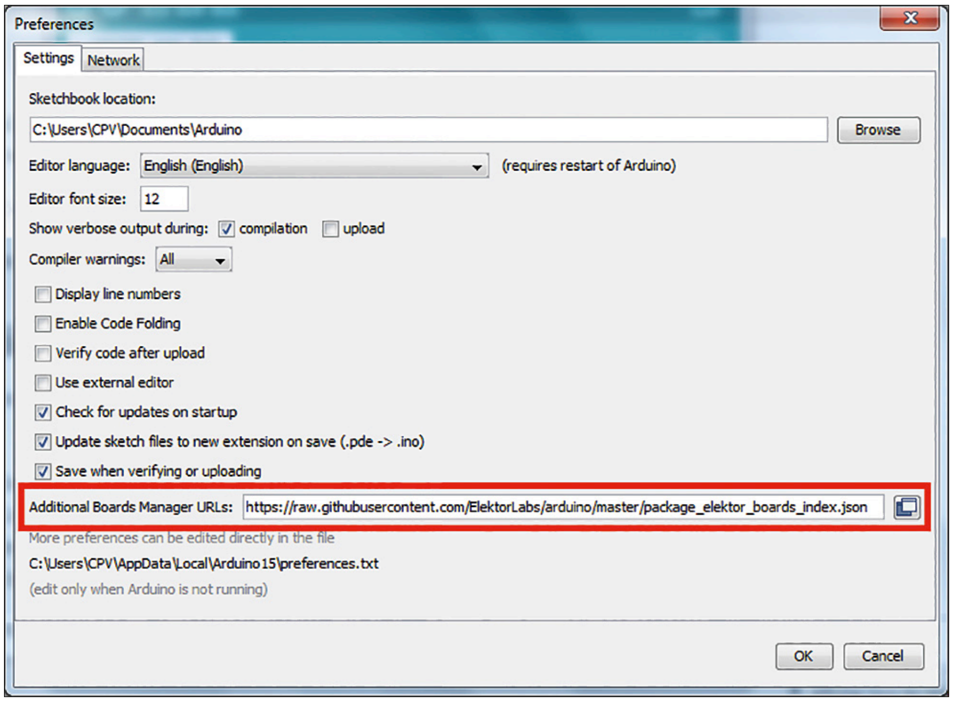

Preferences
Settings | Network
Sketchbook location:
C:\Users\CPV\Documents\Arduino
Browse
Editor language: English (English) (requires restart of Arduino)
Editor font size: 12
Show verbose output during: [✓] compilation [ ] upload
Compiler warnings: All
[ ] Display line numbers
[ ] Enable Code Folding
[ ] Verify code after upload
[ ] Use external editor
[✓] Check for updates on startup
[✓] Update sketch files to new extension on save (.pde -> .ino)
[✓] Save when verifying or uploading
Additional Boards Manager URLs: https://raw.githubusercontent.com/ElektorLabs/arduino/master/package_elektor_boards_index.json
More preferences can be edited directly in the file
C:\Users\CPV\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)
OK Cancel

Figure 4. This is where you enter the URL directing to the Elektor Uno R4 Boards Package.

Web Links

[1] www.elektor.com/150790

[2] www.elektor.com/150555

[3] https://github.com/ElektorLabs/Arduino

Brick-by-Brick Power Supply

modularity = total versatility

By **Ton Giesberts** (Elektor Labs)

Originally designed as a more compact power supply for our 10-MHz DDS Function Generator the board described in this article can double for other applications when fitted out with different modules. Selecting, combining and optimizing, that's what this project is all about.

That 10-MHz DDS Generator, Elektor project no. 150210 [1], uses an analog power supply and a separate powerline filter to create the +3.3 V and ±15 V supply voltages needed. And yes that 10-VA power transformer and AC line filter take up a lot of space. Another way to produce these voltages is to use DC/DC converter modules, powered in el-cheapo style from a notebook adapter with 19 volts DC out or so. 't will keep the whole power supply affair compact, and efficiency will be considerably higher, so no heatsink is necessary.

A DC/DC-converter module with isolated outputs provides the symmetrical supply output voltages. This way the single supply is galvanically isolated from the symmetrical power supply. Although the modules used are said to have an industry standard footprint, there are a few niggling differences in dimensions between manufacturers like Recom and Traco Power. A 6-watt version will also fit.

Schematic and the modules

Looking at the schematic in **Figure 1**, a cheap notebook adapter can be used as

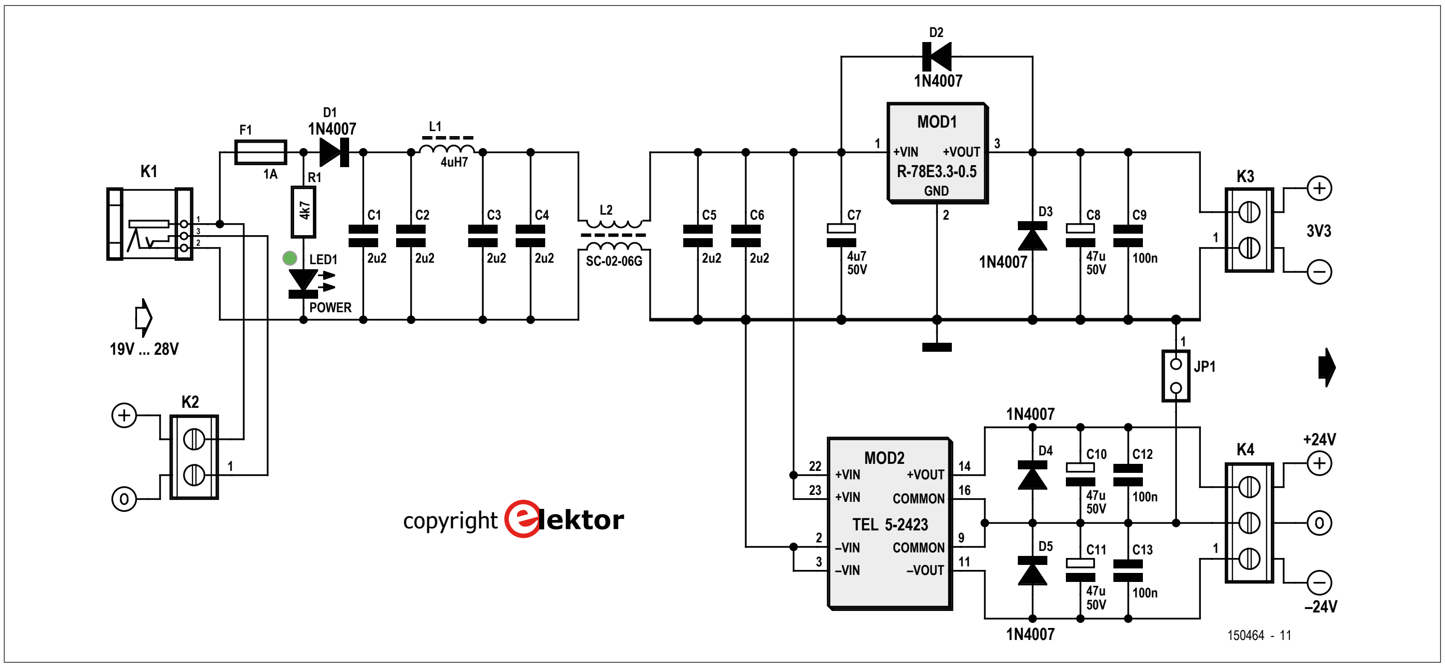
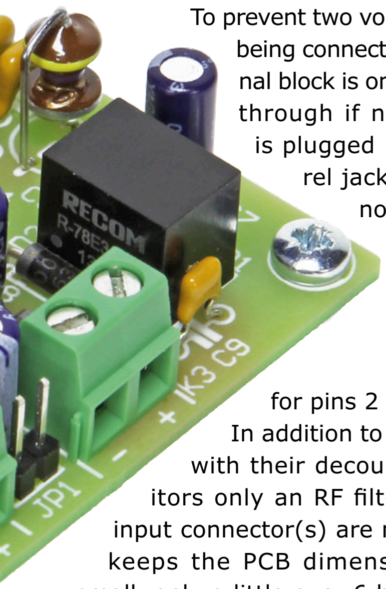


Figure 1. From DC to DC, the proper way and with many options for you to choose from, depending on the modules selected from (among others) Recom and Traco Power.

an input source hooked up on K1. With the present modules used the raw input voltage should be between 19 and 28 VDC. In case the board is used inside another device the input voltage may also be applied to screw terminal block K2. To prevent two voltage sources being connected the terminal block is only connected through if no connecter is plugged into DC barrel jack K1. If K1 is not mounted a wire should be soldered between the holes provided for pins 2 and 3.



In addition to the modules with their decoupling capacitors only an RF filter, fuse and input connector(s) are needed. This keeps the PCB dimensions rather small, only a little over 6 by 6 cm. The modules used can be replaced by others from the same series (other voltages) or types from other manufacturers, that's your call.
LED1 and diode D1 are the supply on/off indicator and reverse polarity protection device respectively.
We should not trust that $2.50 notebook power adapter from the thrift shop (or found in the attic) for purity of the output voltage OR susceptibility to RF noise so a hefty, partly asymmetrical, partly symmetrical, filter is used between it and the DC/DC converter module inputs. The filter also serves to keep SMPSU noise emissions low. It consists of inductor L1, common-mode choke L2 and a bunch of paralleled solid capacitors C1–C6 and one electrolytic, C7, for suppression of wideband noise, pulses, spikes, glitches and spurious products.

The single supply module, MOD1, is a RECOM type R-78E3.3-0.5 (got that?) and is essentially a non-isolated DC/DC step-down converter. Of the boxful of modules Recom dropped off in our lab for "scientific evaluation" (**Figure 2**) we chose the cheaper R-78Exx-0.5 series which comprises modules for 3.3 V and 5 V. For either one the maximum output current is 0.5 A. Another series that will fit (they are even a little smaller) is the more expensive R-78xx-0.5 which can also output 0.5 A but with more output voltages to choose from: 1.5 V, 1.8 V, 2.5 V, 3.3 V, 5.0 V, 6.5 V, 9 V, 12 V and 15 V. Other series have more current and/or other input voltage ranges — check out [2].

MOD2 then, is a type TEL 5-2423 from Traco Power [3]. The footprint for this widely available symmetrical ±15 V DC/DC module is industry-standard and many other manufacturers make pin compatible 5- and 6-watt DC/DC converters which may also be used. The series also comprises single-supply versions. Often the pin from the negative output is not connected then and the ground pin is missing. Other modules have an extra con-

Features

- Powered by any laptop power adapter with V_{out} >19 VDC
- Single-supply output options: 3.3 V or 5 V @ 0.5 A, optionally 1.5 V, 1.8 V, 2.5 V, 3.3 V, 5.0 V, 6.5 V, 9 V, 12 V, 15 V
- Symmetrical-supply output: ±15 V @ 6 W approx., optionally ±9 V, ±12 V
- Industry standard footprint DC/DC modules from Recom and Tracopower
- Zero-SMD, zero–microcontroller design

Figure 2. DC/DC converters allsorts from Recom in a royal presentation box sized 35 × 21 cm. PSU design "the modular way" starts here.

The prototype — grilled

±15 V (19 V input):					
1 kHz pulsed load with 100 Ω on positive supply only				With 1 kΩ load on negative supply	
Duty Cycle	V_{ripple}	$+V_{out}$	$-V_{out}$	$+V_{out}$	$-V_{out}$
20%	0.52 V	14.66 V	15.36 V	14.96 V	15.10 V
50%	0.8 V	14.25 V	15.76 V	14.81 V	15.24 V
80%	0.52 V	13.98 V	16.12 V	14.51 V	15.33 V

V_{ripple} (no load)	30–40 mV_{pp} (spikes, 54 kHz)
+15 V (100 Ω) / –15 V (no load)	+13.65 V / –16.4 V
+15 V (no load) / –15 V (100 Ω)	+15.8 V / –14.25 V

+3.3 V (19 V input): 1 kHz pulsed load with 6.8 Ω	
Duty cycle	V_{ripple} ('triangular/sawtooth')
20 %	140 mV_{pp}
50 %	200 mV_{pp}
80 %	140 mV_{pp}
At lower frequencies ripple is always smaller than 200 mV_{pp}	

V_{ripple} (no load)	30 mV_{pp} (mainly spikes, 100 Hz)
V_{ripple} (68 Ω)	40 mV (mainly spikes, 588 kHz)
V_{ripple} (6.8 Ω)	40 mV (mainly spikes, 588 kHz)

Output voltage
3.361 V (no load)
3.339 V (68 Ω)
3.340 V (6.8 Ω)

Total efficiency on +3.3 V and ±15 V	
Input 19 V / 0.4 A; output 3.3 V / 6.8Ω; +15 V / 100 Ω; –15 V, 100 Ω:	80 %
Input 28 V / 0.28 A; output 3.3 V / 6.8Ω; +15 V / 100 Ω; –15 V, 100 Ω:	78 %

Component List

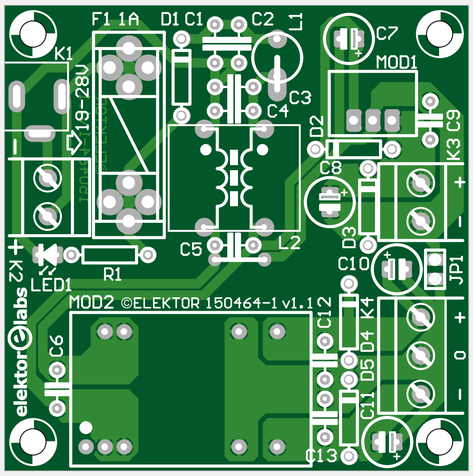
Resistor
R1 = 4.7 kΩ, carbon film, 5%, 0.25W, 250V

Capacitors
C1-C6 = 2.2µF 50V 20%, ceramic Y5V, 0.2'' pitch
C7 = 4.7µF 50V 20%, 2mm pitch, diam. 6.3mm max.
C8,C10,C11 = 47µF 50V 20%, 2.5mm pitch, diam. 6.3 mm max.
C9,C12,C13 = 100nF 50V 10%, ceramic X7R, 0.2'' pitch

Inductors
L1 = 4.7µH, 3.05A, 80mΩ, 10%, radial, 5mm pitch, e.g. Epcos B82144B2472K000
L2 = 600µH, 2A, 2 x 50mΩ, common-mode choke, 17.5x14 mm, Kemet SC-02-06G

Semiconductors
D1-D5 = 1N4007 (1000V, 1A)
LED1 = LED, green, 3mm
MOD1 = R-87E3.3-0.5, Recom (3.3V, 0.5A)
MOD2 = TEL 5-2423 Traco Power (5W, ±15 V)

Miscellaneous
K1 = DC barrel jack, 1.95mm pin, 12V, 3A
K2,K3 = PCB screw terminal block, 2-way, 0.2'' pitch
K4 = PCB screw terminal block, 3-way, 0.2'' pitch
JP1 = 2-pin pinheader, vertical, 0.1'' pitch
F1 = fuseholder, 20x5mm, 500V, 10A
F1 = fuseholder cover, 20x5 mm
F1 = fuse, cartridge, 1A slow blow, 20x5mm
PCB # 150464 v. 1.1 from the Elektor Store, www.elektor.com



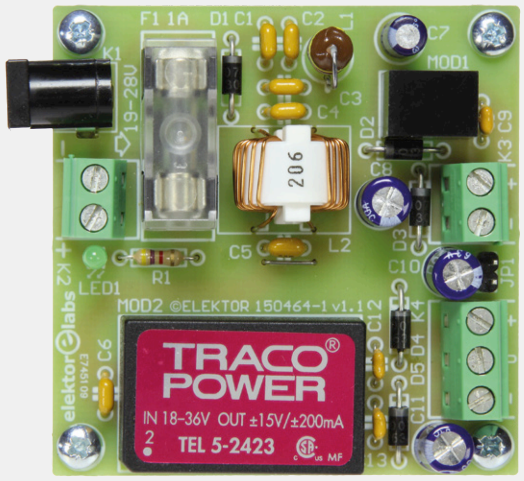


Figure 3. The extremely simple circuit board designed for the modular power supply. Position MOD2 on the board follows an industry-standard footprint hence accepts a great many different DC/DC converters out there.

trol input (pin 1, for instance the REC5 series from Recom) which can be left unconnected. Available output voltages are 3.3 V, 5 V, 12 V, ±12 V and ±15 V. For sound advice on the best device to use for your specific application, take a good look at the specs at [3]. Some other manufacturers also have 9-V, 15-V, ±5-V, and ±9-V devices on offer. The advantage in general is full galvanic isolation between input and output. On the downside, they are not as cheap in the league of standard-voltage regulators. Jumper JP1 allows interconnecting the ground lines of the two DC/DC modules. Sometimes though it's preferable to make this connection between different power supply voltages at the load end, hence the jumper. From the measurements in the **inset** we see that the dual-supply module has a slight problem with an asymmetrical load. The total voltage-out stays at 30 V but the "ground" level shifts. Hence it's advised to have roughly equal loads on the positive and negative output rails.

Caveat Emptor

When using different modules than mentioned here pay close attention to the input voltage ranges. Make sure the hi/lo input range of the two modules is the same or has a wide overlap. Here we chose two types with roughly similar input voltage ranges so the input voltage range we got from this configuration is about 19 V to 28 V. For your convenience (and our disclaimer) this is also printed beside the input connectors. For both modules, types exist with lower input voltage ranges.
Please compare datasheets for input voltage range and footprint and use common sense before selecting and fitting other modules.

Construction

Figure 3 shows the board designed for the Brick-by-Brick PSU, and close-up photos of our working and tested board. We declare all parts on the PCB easy to mount because they are through-hole only. There's one jumper wire to fit on the board — it's next to C5. We also declare that no software, microcontroller or embedded technology can be or should be identified as inherent to this project.

(150464)

Web Links

[1] DDS Function Generator, Elektor November & December 2015, www.elektormagazine.com/150210
[2] Recom R-78xx-0.5 series: www.recom-power.com/pdf/Innoline/R-78xx-0.5.pdf
[3] Traco Power TEL 5 series: www.tracopower.com/products/browse-by-category/find/tel-5/3/

reichelt elektronik

Technology connects.

PROXIMITY SWITCH
HUMIDITY
POSITION SENSOR
FLOW
PRESSURE
HALL EFFECT
ULTRASOUND
FORCE

✓ More than 45 years of experience
✓ 24-hour shipping
✓ More than 50,000 products

SUBSCRIBE NOW!

Newsletter

Receive weekly fresh information about

✓ product innovations
✓ specials
✓ Price reductions

Hall effect sensors

SIEMENS

Housing: P-SSO-3-2

Order number		Magnetic range	
TLE 4905L	£ 0,44	±17 mT	unipolar
TLE 4935L	£ 0,46	±20 mT	bipolar / latch

Radiometric sensor, linear — Honeywell

- Operating voltage: 4.5 … 10.5 VDC
- Power consumption: 8.7 mA (@ 5 VDC)
- Operating temperature: −40 … +150 °C
- Linearity: 1 % typ.
- Response time: 3 µs

Housing: TO-92,
Output: 0.2 V / 1.5 mA

Order number		Magnetic range	Sensitivity
SS 496 A1	£ 1,60	±84 mT	2,5 mV/G
SS 495 A	£ 0,91	±67 mT	3,125 mV/G
SS 495 A1	£ 2,48	±84 mT	3,125 mV/G

Radiometric sensor, linear, SMD — Honeywell

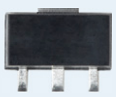
- Operating voltage: 2.7 … 6.5 VDC
- Power consumption: 10 mA
- Operating temperature: −40 … +100 °C
- Linearity: 1 % typ.
- Response time: 3 µs

Housing: SOT-89
Output: 1.0 V / 1.5 mA

Order number		Magnetic range	Sensitivity
SS 59 ET	£ 1,05	±65 mT	1,0 mV/G

Hall effect sensor, digital, SMD — Honeywell

Temperature-compensated digital hall effect sensors



Unipolar, housing: SOT-89
Output: 0.4 V / 20 mA

Order number		Magnetic range
SS 543 AT	£ 1,47	7,5 … 18,0 mT
SS 549 AT	£ 1,95	23,5 … 39,0 mT

Inductive proximity switch

CONTRINEX

Excellent inductive sensors for contactless detection of metal parts for daily use in automation solutions.
Ideal for use in demanding ambient conditions.

Excerpt from our product range.
For the entire product spectrum please visit: http://rch.lt/kJ

Order number		Ø x L	Switching distance		Installation
DWAD 509 M8 390	£ 59.75	M8 x 45 mm	0 … 4 mm	2 m cable	quasi-flush
DWAD 623 M5	£ 31.28	M5 x 25 mm	1.5 mm	2 m cable	flush
DWAD 617 M12	£ 25.27	M12 x 50 mm	4.0 mm	2 m cable	non-flush
DWAS 603 M8 129	£ 31.54	M8 x 29 mm	1.5 mm	M8 connector, 3-pin	flush
DWAS 713 M8 001	£ 49.30	M8 x 60 mm	6.0 mm	M8 connector, 3-pin	non-flush
DWAS 603 M12	£ 21.81	M12 x 60 mm	2.0 mm	M12 connector, 4-pin	flush
DWAS 503 M12	£ 31.87	M12 x 60 mm	6.0 mm	M12 connector, 4-pin	flush

Sensor tester up to 100 mA, LED, buzzer, micro USB — CONTRINEX

- Fast field tests different sensor types
- Battery

Order number	
ATE 0000 010	£ 35,23

Humidity sensor 0 … 100 % rF, TO 39 — B+B SENSORS

Digital humidity sensor with IC interface in pressure-resistant TO39 housing (up to 16 bar), suitable for dew point measurements.

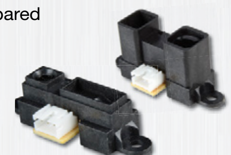
Order number		
HYT 939	£ 20,15	TO 39

Pressure sensors, UsV 4.75 – 5.25 — freescale semiconductor

Order number		PR kPa	Sens mV/kPA	Lin %
MPX 5010DP	£ 8,88	0 - 10	450	±5
MPX 5050DP	£ 10,71	0 - 50	90	±2,5
MPX 5100DP	£ 11,23	0 - 100	45	±2,5
MPX 5500DP	£ 7,05	0 - 500	9	±2,5

Distance sensors — SHARP

Very high reliability as well as increased precision compared to conventional sensors.



Types: GP2Y0A

Order number		Range (cm)	L x W x D (mm)
GP2-0215	£ 4,37	20 ... 150	29,5 x 13,0 x 21,6
GP2-0430	£ 3,33	4 ... 30	37,0 x 18,9 x 13,5
GP2-1080	£ 3,62	10 ... 80	29,5 x 13,0 x 21,6

Order number			
DMC01-SC150	£ 1,40	3-pin	Data cable

Ultrasonic sensors

Ultrasonic ceramic transmitter and receiver for 40 kHz

Ø 9,9 mm, H 7,1 mm

MUS-40E	£ 1,95	Receiver
MUS-40S	£ 1,95	Transmitter

Daily prices ! Price as of : 01.06.2016

Prices in **GBP** plus statutory VAT, plus shipping costs
reichelt elektronik, Elektronikring 1, 26452 Sande (Germany)

Payment Methods:

SOFORT BANKING · VISA · amazon payments · PayPal

Order now! **www.reichelt.co.uk**

Order Hotline: **+49 (0)4422 955-360**

welcome in your
ONLINE STORE

EDITOR'S CHOICE

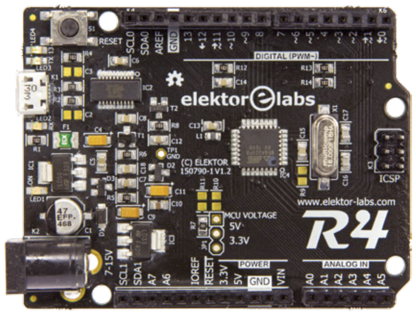
A multimeter is an indispensable tool for anyone in the electronics industry. In recent years we have also seen multimeters working wirelessly. The Mooshimeter multimeter is both part of this new generation and a full replacement for a standard multimeter. It uses a relatively small cabinet made of polycarbonate, and connects to an Android or iOS smartphone via Bluetooth 4.0. An app can then be installed to display the measured values and provide a variety of settings. The measuring part of the Mooshimeter consists of two 24-bit A/D converters allowing two channels to be measured at the same time. Furthermore, there is a built in data logger. The Mooshimeter is an excellent instrument with unique features, I would not want to miss out!

Harry Baggen
Elektor Labs

www.elektor.com/mooshimeter

Elektor Bestsellers

1. Elektor Uno R4
   www.elektor.com/elektor-uno-r4



2. Raspberry Pi 3 Model B
   www.elektor.com/rpi3
3. ARM Microcontroller Projects
   www.elektor.com/arm-projects
4. Six Digit Nixie Clock
   www.elektor.com/six-digit-nixie-clock
5. All-new Pro Tech Toolkit
   www.elektor.com/ pro-tech-toolkit
6. C Programming with Arduino
   www.elektor.com/cpwa
7. The EAGLE Companion
   www.elektor.com/eagle-companion
8. DVD Elektor 2015
   www.elektor.com/dvd-2015

Arduino Uno – 45 Projects for Beginners and Experts

This book covers a series of exciting and fun projects for the Arduino, such as a silent alarm, people sensor, light sensor, motor control, internet and wireless control (using a radio link). Contrary to many free projects on the internet all projects in this book have been extensively tested and are guaranteed to work!

member price: £24.95 • €31.46 • US $36.00

www.elektor.com/arduino-projects

3D Printing and Autodesk 123D Design

Thanks to the rapid development in 3D printing tools and services, anyone can now make things without being a member of a large organization, or without specialized facilities. This book provides you with basic knowledge and information about 3D printing technologies so that you can get started. You will learn about the latest trends in 3D printing and gain a background to product creation.

member price: £29.95 • €40.46 • US $44.00

www.elektor.com/3d-printing

Atlas DCA75 Pro Semiconductor Analyser

Featuring USB and a graphics display, this new instrument represents a major advance in handheld semiconductor analysis. Connect your component any way round to see the detailed component schematic on the LCD as well as pinout and comprehensive measurements. Supports IGBTs, Voltage Regulators, MOSFETs, JFETs (including SiC types), transistors, diodes, LEDs, Triacs (up to 10mA), thyristors (up to 10mA) and lots more.

member price: £87.95 • €112.46 • US $129.00

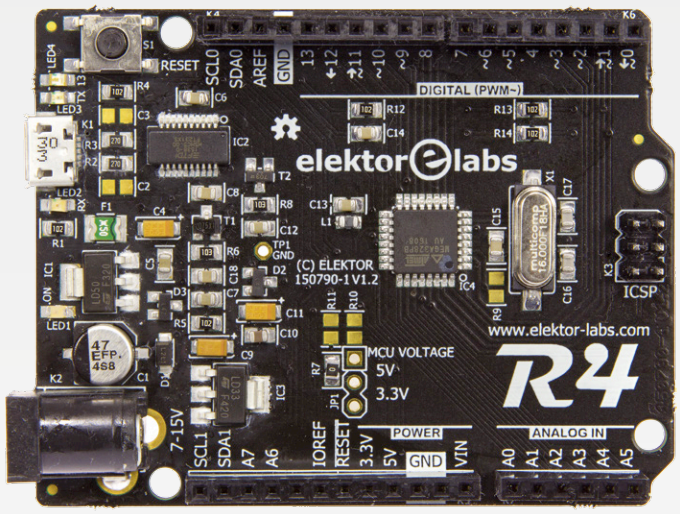
www.elektor.com/ atlas-dca75-pro

Elektor Uno R4

When a manufacturer releases a B version of an existing product the differences between the old and the new are usually small. Not so in the case of Atmel's ATmega328P, the processor at the heart of the Arduino Uno R3. The B-type of this MCU has so many extra features that a new revision of the R3 is justified: Elektor Uno R4.

At a glance:

- ATmega328PB @ 16 MHz
- 2x UART
- 2x I²C
- 2x SPI
- 9 PWM outputs
- 8 analog inputs
- 24 GPIO pins
- On-board 5 V and 3.3 V voltage regulators
- Arduino compatible Boards package
- Open source, open hardware design



MEMBER PRICE: £21.95 • €26.95 • US $31.00

www.elektor.com/elektor-uno-r4

CAN, ARM and Arduino in One Book

Limited time offer for GREEN and GOLD members:
15% Discount
plus free shipping!

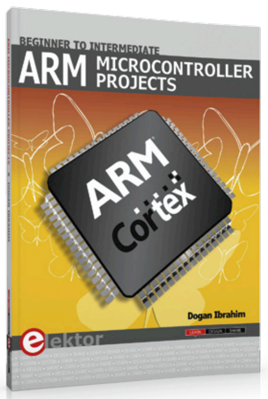
Arduino Uno R3 Successor

New microcontroller, more I/O and more features!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

ARM Microcontroller Projects

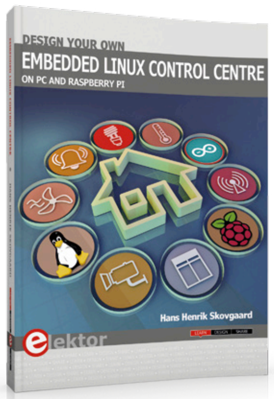


This book makes use of the ARM Cortex-M family of processors in easy-to-follow, practical projects. It gives a detailed introduction to the architecture of the Cortex-M family. The architecture of the highly popular ARM Cortex-M processor STM32F107VCT6 is described at a high level, taking into consideration its clock mechanisms, general input/output ports, interrupt sources, ADC and DAC converters, timer facilities, and more.

member price: £28.95 • €35.96 • US $41.00

www.elektor.com/arm-projects

Design Your Own Embedded Linux Control Centre 3



This book is all about building your own DIY home control system. It presents two innovative ways to assemble such a system: By recycling old PC hardware - possibly extending the life of an old PC, or by using Raspberry Pi. In both cases, the main system outlined in this book will consist of a computer platform, a wireless mains outlet, a controller and a USB webcam - All linked together by Linux.

member price: £30.95 • €38.66 • US $45.00

www.elektor.com/elcc3

Six Digit Nixie Clock

This precise Nixie clock combines modern and legacy technologies. The clock receives time signals from the GPS system and displays the time with six Nixie tubes. And this model also shows the seconds! The kit contains all the parts (bare PCBs, Nixie tubes, programmed controllers, all the other electronic components and a power supply 9V). Soldering required, not assembled. Case not included.

member price: £91.95 • €116.96 • US $134.00

www.elektor.com/six-digit-nixie-clock

Welcome to the SHARE section

By **Thijs Beckers** (Elektor Netherlands Editorial)

Airplane electronics

Have you ever paused to consider the electronics that is used on board of an airplane? Without electronics, modern airplanes would be uncontrollable and landing and take-off during poor visibility would be very dangerous, if not impossible. All kinds of safety systems and comfort elements are built in to enable passengers to be transported safely. But why then do you have to switch off your own electronics when flying on a regular commercial flight? Alright, these days switching to airplane mode is also acceptable, but nevertheless. Apart from the argument that you are in a Faraday cage, where typically 150+ mobile phones would be transmitting at their maximum power in an attempt to contact a cellphone tower, I have been unable to find any convincing research results. If you know of any then you may share it with us!

Another strange thing: why are the cabin lights dimmed during take-off and landing? For this I have been able to find a plausible explanation. The light level in the cabin is set to match the light level outside, so that in the event of an emergency the difference between inside and outside is as little as possible and your eyes don't have to adapt to the conditions outside, which can save precious seconds during an emergency situation. This is also the reason that the 'shutters' have to remain open.

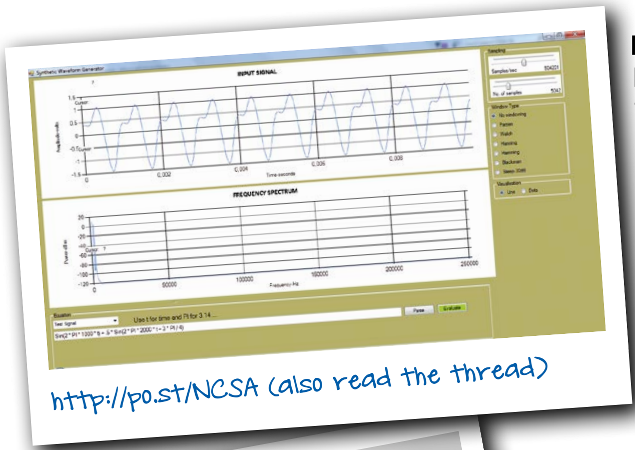
The power supply on board of an airplane is also quite interesting. Especially the choice of three-phase 115 V/400 Hz and the 28 V DC distribution network. At 400 Hz, a generator or transformer of any given power rating is much smaller and lighter then one of the same rating at 50 or 60 Hz. This is, of course, very important in an airplane. A rule-of-thumb that is used is that every kg that can be saved, results in a total reduction of 5 kg because of the additional savings of construction and fuel that are no longer required to move this one kg across the flight distance. Going to an even higher frequency would introduce several disadvantages. The main one is the voltage drop that results from the self-inductance of the cables. In an airplane the voltage drop at 400 Hz can be up to seven times greater than that at 60 Hz.

An other interesting piece of trivia to conclude: some airplanes can be controlled by an iPad, which can nearly completely replace the pilot when something goes wrong. The automatic pilot is then controlled by the iPad and the airplane can still be landed safely. ⏮

Photo credit: www.popsci.com (160022)

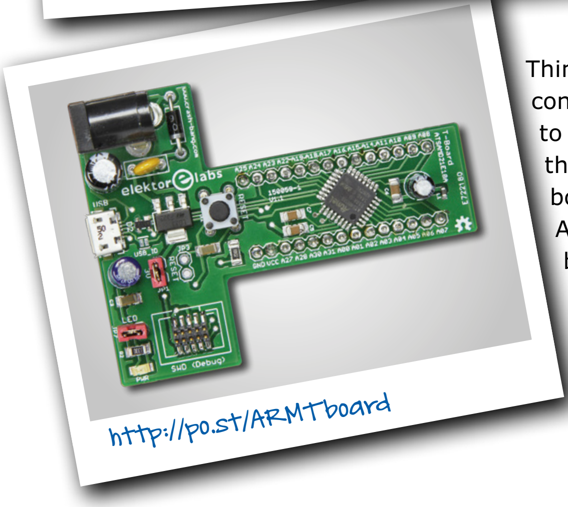
Dot Labs... Highlitez

Is there life after publication on paper?
To answer this deeply philosophical question all we have to do is browse Elektor.Labs.



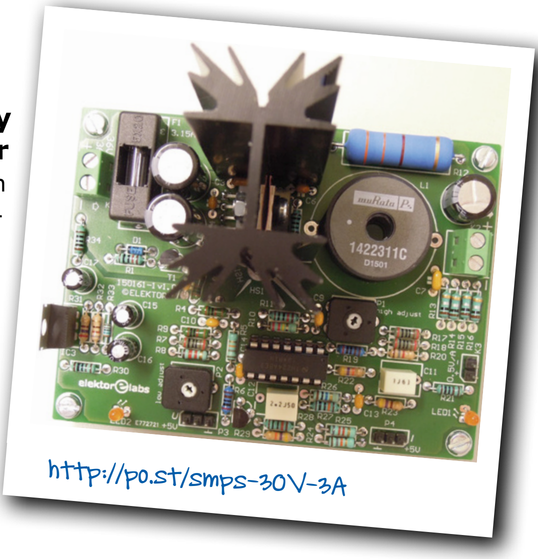
New software for the Network Connected Signal Analyzer

That things can go very fast occasionally was proven recently by the Network Connected Signal Analyzer — NCSA — project of which the first part was published in the March & April 2016 edition of Elektor Magazine. Even before we had the time to publish the second installment a major software update was announced at Elektor.Labs by user BreedJ, including a video showcasing the new features. BreedJ and NCSA designer Neal Martini joined forces and the result is a project on GitHub where you can download the improved tools. Binaries for Windows are included but with Visual Studio 2015 you can compile it yourself.



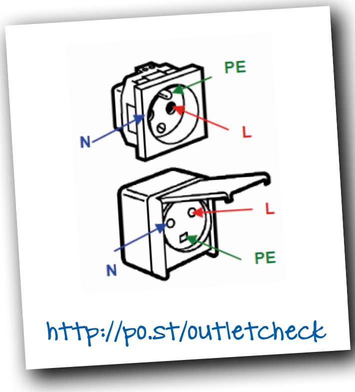
A bootloader for the SAMD ARM T-board

Things are better with bootloaders — and since this is especially true for microcontroller development boards we decided to investigate the matter for relevancy to our SAMD ARM T-Board 150059. It took some time to figure it out, but now that we know how, we have reprogrammed our stock with a bootloader. The bootloader in question is provided by Atmel and is described in application note AT07175. Our microcontroller is the SAMD21E18A and to program it through the bootloader you need to download and install Atmel's free utility SAM Boot Assistance a.k.a. 'SAM-BA'. Details on how to proceed from here can be found at www.elektormagazine.com/labs/arm-t-board-150059



The UniLab power supply gets a successor

Some subjects remain popular for DIY projects, either because a satisfactory solution wasn't found yet, or just because the result is something useful. Clocks is one example, power supplies another. At Elektor.Labs we have done many, many PSU designs, from too simple to overly complex and yet we manage to find excuses to do a new one every time. In 2010 we published a power supply called UniLab (Elektor project 090786). It was able to deliver 30 V and 3 A. This new design, based on a different step-down converter, the LM2677-ADJ, can be considered its successor. It has a higher switching frequency (260 kHz), which reduces the size of the output filter.



Are your power outlets wired correctly?

Home Improvement was never more popular. Unfortunately many DIY enthusiasts combine a vague understanding of electrical wiring with boundless creativity and since the continental European three-terminal power outlet can be wired in six different ways, of which only one is correct, dangerous situations can easily arise. So how about building a tool to identify the terminals of fishy outlets and prevent lethal accidents? This is exactly what user Frederik did when he developed a monitor to show if a power outlet is wired properly by lighting a few LEDs. It is capable of identifying phase-phase-earth (115 VAC) and phase-neutral-earth (230 VAC) connections making it useful in many countries. ◄

(150819)

Your own Media Player

For PC, Raspberry Pi, tablet and many more

Harry Baggen (Elektor Labs)

Every PC, laptop, smartphone and tablet these days comes complete with software preinstalled for playing photo, audio and video files. But this is usually not general-purpose or flexible enough; some formats are not supported, you will miss certain configuration settings or, for example, it cannot play files from across a network connection. What then, is the best alternative?

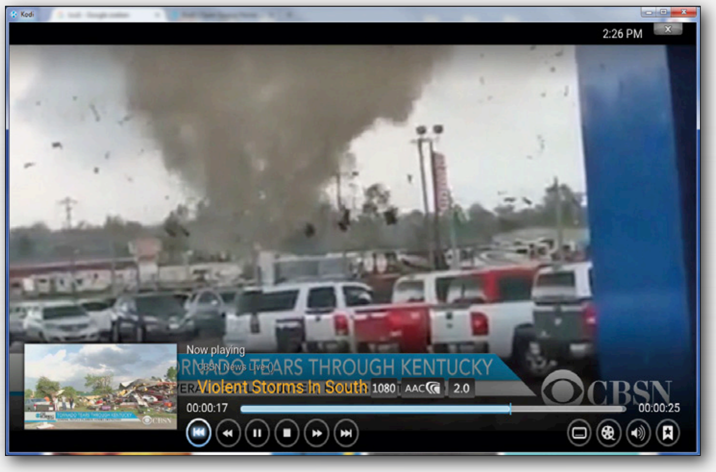
There is a whole bunch of media player software out there for all kinds of platforms, but when it comes to versatility and expansion capability one clearly stands out from all the others: Kodi, perhaps better known by its former name XBMC. This is an open-source media player which has been developed by the XBMC foundation, a group of enthusiastic programmers/users. The Kodi/XBMC was at one time (in 2004) developed as an independent media player for the first Xbox, hence the name Xbox Media Center (shortened to XBMC). Later on, versions were also developed for other operating systems, such as Windows, Linux, iOS and Android. Also various stand-alone versions have been made for smart-TVs, settop boxes and media players. Kodi now supports more than 75 languages.
One of the strong points of Kodi/XBMC is the option of adding so-called add-ons, which can be used to increase its functionality or to provide access to external media sources via the Internet. In addition to a large number of official add-ons there are also many 'gray' add-ons in circulation, which can be used to access all kinds of illegal audio and video sources. The XBMC foundation is increasingly emphatic in distancing itself from these.

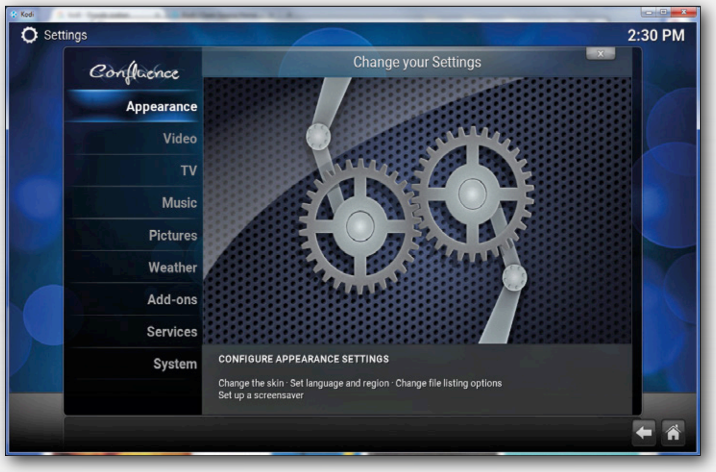
In this article I would like to say something about the many features that Kodi has to offer and on which devices you can use it. Many colleagues and acquaintances appear to be quite familiar with the name, but don't quite know how to get started with it themselves. It is actually quite easy and on YouTube there are by now hundreds of instructional videos to be found that can extend a helping hand.

Many versions

On the Download page of the official Kodi website [1] there are all the current versions available for Windows, Linux, Mac OS X, iOS, Android and a few specific hardware platforms such as Raspberry Pi and Amazon Fire TV. For most of these system the user only needs to download the installation file and run that on their computer. For Android devices Kodi is also available from the Google Play Store. These days many more Android-based media-players are appearing on the market and Kodi is then quickly installed following this route.
An enormously popular hardware platform for building yourself a cheap media-player is the Raspberry Pi. Probably a large number of all the RPis that have been sold have been used for this purpose. Already starting from the very first version of the RPi special versions have been assembled for this. One of the most popular is Open ELEC, which stands for Open Embedded Linux Entertainment Center [2]. This is a so-called *Just Enough Operating System* (JeOS); you therefore do not need to install Linux or any other OS first, all you need is already built into the software. OpenELEC is not only available for the Raspberry Pi A, B, 2 and 3, but also for various other hardware platforms such as PCs with an X86-processor (ideal for a HTPC, no Windows required) and media boxes with a Freescale IMX6 CPU (Cubox). Although the installation of the OpenELEC software on the RPi is not hard, the procedure is a little different compared to the normal Kodi version. You just have to download a disk image (a literal copy of the memory contents) and this has to be copied to a USB-stick or (micro)SD card. This cannot be done by copying the image using Windows Explorer or similar. On a separate page [3] at OpenElec it is explained how you can do this from Linux, Windows and Mac OS X. Most Elek-

A large number of all the RPis that have been sold are used for Kodi





tor readers will be using Windows and in this case you need to use the program Win32Diskimager [4] to make a 1:1 copy of the image to the USB-stick or SD-card. After that, the stick or card only needs to be plugged into the RPi and the RPi can be turned on.

Usage

At first it takes a little bit of time to get used to the Kodi interface and the many configuration options. It is a good idea to first familiarize yourself with the user interface and configuration options using the Quick Start Guide in the Kodi wiki [5]. Under System/Settings there are a large number of tabs with configuration options, where the access level can also be switched between Basic, Standard, Advanced and Expert (start with Standard, only the most important settings are then displayed). You will not need to change most of the settings, but it can be nice to select under Appearance/International the desired language for the interface and under Videos/Subtitles the desired language for the subtitling. At System/Audio output you can select the desired audio output format. With an RPi this depends in whether its HDMI-output is connected to a surround receiver or directly to the TV.

In addition to playing videos and photos from your own media collection it is, of course, also interesting to active a number of add-ons which give access to a large number of Internet media sources, varying from Missed Broadcast to cooking and news programs from various countries. When you select Videos/Add-ons and subsequently Get more… (the same is true for Pictures and Music) there appears a list of available add-ons which are available in Kodi by default. You can then install the desired add-on, after which it appears in the Videos/Add-ons overview. This list can be extended with even more add-ons. An overview of all the official Kodi add-ons can be found at [6]. Here you will have to keep an eye on the Kodi version that you are using. Each version number of Kodi has its own name which is used to group the add-ons. The current version 16 is called 'Jarvis'. The installation of additional add-ons is described in the Kodi wiki [7], but it is also recommended to search for this subject on YouTube.

Not all add-ons are approved by the Kodi developers, but there are nevertheless a large number of interesting unofficial add-ons in circulation (not to be confused with illegal). We name two sources which have collected a large number of add-ons and compiled those into handy zip files, so that the user has quite easy access to a large number of add-ons in one go. SuperRepo [8] has collected more than 2000 add-ons, while TVAddons [9] offers some 1200. There are many illegal ones among these, so take care which ones you install! ⏮

(160031)

Web Links

[1] https://kodi.tv/

[2] http://openelec.tv/

[3] http://wiki.openelec.tv/index.php/HOW-TO:Installing_OpenELEC/Creating_The_Install_Key#tab=DiskImage

[4] https://sourceforge.net/projects/win32diskimager/

[5] http://kodi.wiki/view/Quick_start_guide

[6] http://kodi.wiki/view/Category:All_add-ons

[7] http://kodi.wiki/view/Add-ons

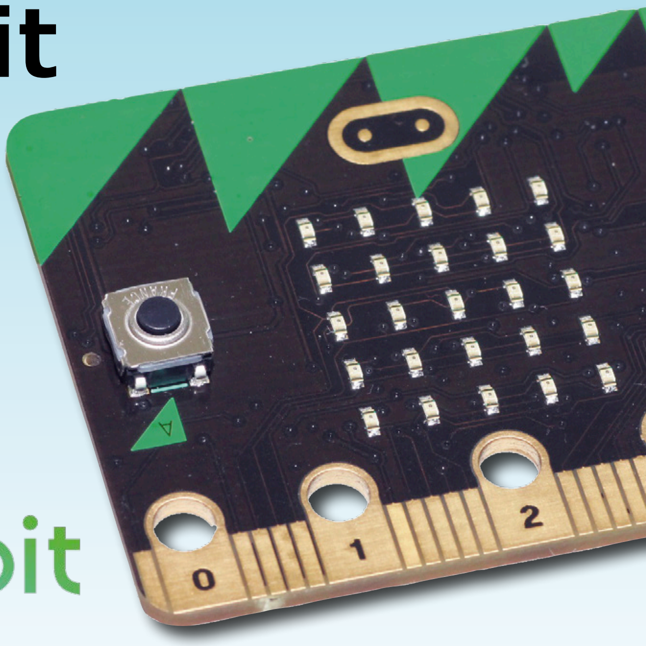
[8] https://superrepo.org/

[9] https://www.tvaddons.ag/

The BBC micro:bit

Reviewed: Auntie Beeb's microcontroller development for kids

By **Clemens Valens** (Elektor.Labs)



After Earth, Air, Water and Fire, Software has become the fifth element of life. Unlike the first four elements, Software is the result of hard human labor, produced by millions of fingers hammering away on grotty computer keyboards. In a bid to secure enough brainpower for tomorrow's Software needs, the BBC has started to mine the UK's 11- and 12-year-olds by giving every kid a micro:bit microcontroller development board for free.

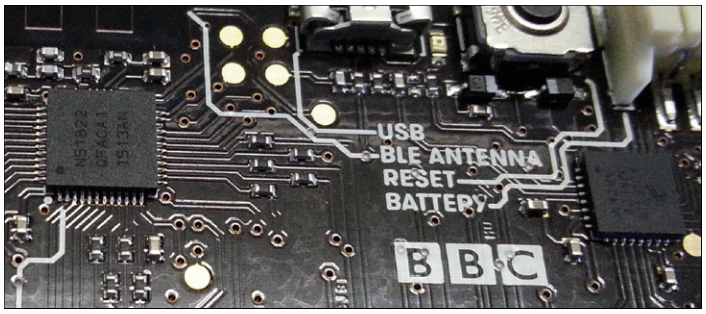


Figure 1. Close-up of the main electronics on the BBC micro:bit.

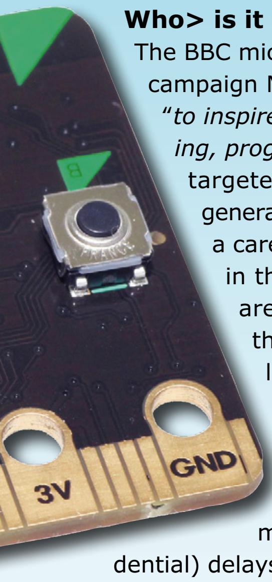
BBC micro:bit specifications

- nRF51822 32-bit ARM Cortex M0 @ 16 MHz
- 16 KB RAM
- 256 KB Flash memory
- Bluetooth Low Energy master/slave capable
- 5x5 LED matrix
- MMA8652 3-axis accelerometer
- MAG3110 3-axis magnetometer
- Two pushbuttons
- 5 banana/crocodile clip connectors
- Edge connector with SPI, I²C, GPIO and 6 analogue inputs
- USB Micro B connector
- JST power connector (3 V)
- Drag 'n' drop and Over-The-Air (OTA) programming
- Programmng tools: TouchDevelop, Blocks, Javascript, MicroPython, C/C++
- mbed enabled

What> is it?
The BBC micro:bit is a small (5 x 4 cm) microcontroller board with an ARM Cortex-M0-based microcontroller with integrated Bluetooth Low Energy (BTLE) radio from Nordic (nRF51822, **Figure 1**), a 5 x 5 LED matrix, two pushbuttons, a 3D accelerometer (NXP MMA8652), a 3D magnetometer (NXP MAG3110) and an extension connector. It can be drag-'n'-drop programmed over USB as an ARM mbed-compatible mass storage device (implemented in a second ARM Cortex-M0 MCU, an NXP Kinetis KL26), and over-the-air (OTA, *sic*) thanks to its Bluetooth radio. All the parts on the board are sponsored by their respective manufacturers who helped a lot in keeping the board's costs as low as possible. Over 25 organizations have partnered with the BBC to create the board.

The board's graphical design is intended to appeal to children and has the shape of a face (the pushbuttons are the eyes, the extension contacts its teeth) with different silkscreen hairdos of several colors. The board was also designed with safety in mind, which is why it is powered from an external battery pack (two AAA 1.5 V cells) instead of from an on-board button cell as was the case for earlier designs. Power can also be applied through the USB connector.

Since the ultimate goal of the BBC is to get children to create software [1], the micro:bit is supported by a suite of online programming tools targeted at novices. Currently four different languages are available: Block Editor and TouchDevelop, both from Microsoft, a visual Java Script editor from Code Kingdom and MicroPython. The more advanced user can also program the board directly in C/C++ from the mbed development environment [2].



Who> is it for?

The BBC micro:bit is part of the BBC's 2015 educational campaign Make It Digital [3]. The campaign that aims *"to inspire a new generation to get creative with coding, programming and digital technology"* is not just targeted at school children but at people from all generations with the hope to motivate them to start a career in digital technology. It is estimated that in the UK alone 1.4 million digital professionals are needed over the next five years and with the micro:bit the BBC hopes to fill that gap at least to an extent.

The little microcontroller board is up for grabs for all Year-7 pupils (11- and 12-year olds) in the UK, at schools but also those educated at home. This amounts to almost one million boards. Due to unforeseen (and confidential) delays distribution of the boards do not start until the end of March 2016. A few months later (exactly when remains unknown) the board will also be made available to other people through resellers. The intention is to open-source at least parts of the software and to publish the schematics. Because of the interesting hardware and many available tools the micro:bit is sure to appeal to electronics enthusiasts and other makers.

The website supporting the BBC micro:bit not only helps children to create fun applications, it also tries to educate teachers and (grand)parents so that they can help the/their kids. Lessons and courses are available for downloading to get programming novices started. In short, the micro:bit can be used by anyone willing to learn and invest a bit of time and effort.

What> can you do with it?

Besides scrolling text messages and producing other visual effects on the LED matrix the micro:bit can be used for many other applications. Because of its battery pack and Bluetooth LE connection the board is an excellent candidate for IoT, wearable and mobile applications (**Figure 2**). Its on-board sensors allow for orientation and movement detection making it suitable for games and game controllers or remote controls for other devices. The board can also be used as the brains of an application, like a robot or cart, by using the extension connector.

The edge extension connector breaks out 19 GPIO pins of the main MCU (plus power supply), giving access to the pushbuttons, six analogue inputs, a(n) SPI bus and the I²C bus that is also connected to the accelerometer and magnetometer (**Figure 3**). Furthermore, five pins have been designed as large holes to accept banana plugs and crocodile clips allowing quick and easy connections to breadboards and other hardware.

Some of the extension pins connect to the 5 x 5 LED matrix, that is actually laid out as a 3 x 9 matrix. The reason for this surprising configuration is to enable touch detection using the LEDs capacity to respond to incoming light. The matrix is capable of correctly lighting one LED while measuring light intensity on another at the same time.

What> the f... uture?

Much time and money has been invested by the BBC and its partners to develop a low entry-level software development platform and to put it in the hands of as many people as possible. While handing out one million of these devices for free certainly makes for an impressive, never-seen-before event in the world of electronics, questions can be asked as to its effectiveness. Certainly, some forced micro:bit users, a few thousand maybe, will get inspired and start a career in software or electronics, but how many of these boards will be cast away, end up in a drawer, or on eBay? How many teachers or parents will be able to master the platform and transmit their new knowledge to their pupils? Only the future can tell. One thing is for sure though: with one million micro:bit boards out there the computing world has been enriched with a great, new low-cost microcontroller development board, and we are sure to see many applications built around it, some created not just by children and whizzkids, but makers also. If you do something cool with a BBC micro:bit do let us know — pop it on the Elektor.Labs website and just conceivably it may appear in *Elektor Magazine* or on TV someday.

(150832)

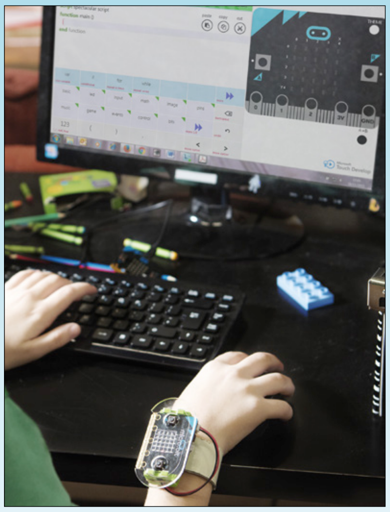


Figure 2. It's wearable Jim: build a smart watch with your micro:bit. (courtesy BBC)

Web Links

[1] https://www.microbit.co.uk/
[2] https://developer.mbed.org/platforms/Microbit/
[3] http://www.bbc.co.uk/makeitdigital

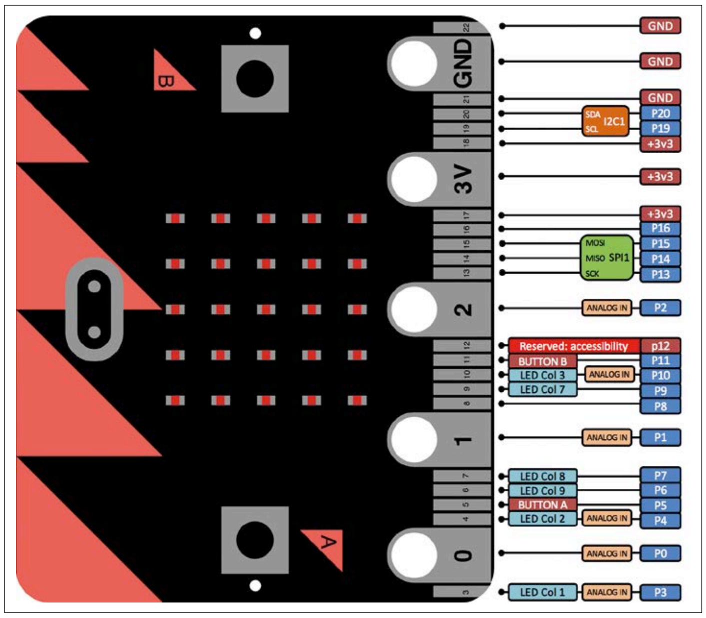
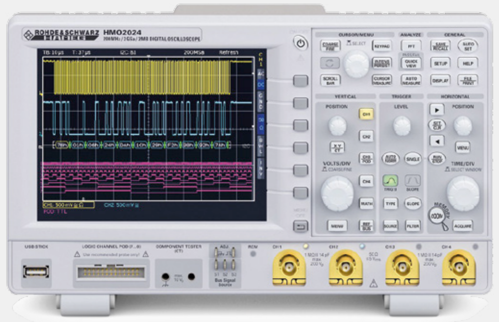


Figure 3. Pinout of the micro:bit's edge extension connector. (source: ARM mbed)



Which 'Scope?

Elektor 3/2016 (May & June), p. 18 (150769)

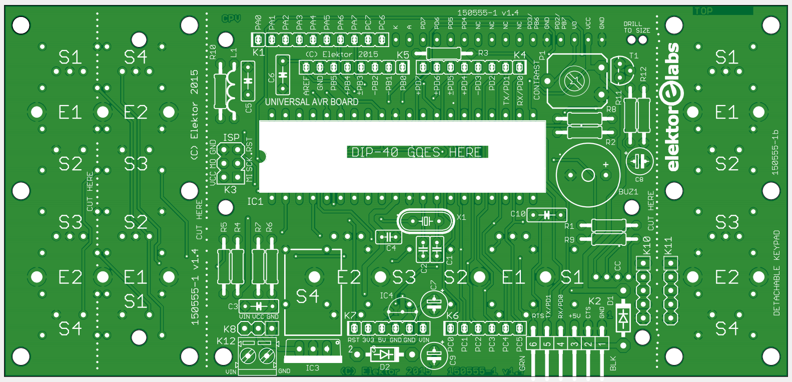
CORRECTION. This article should help engineers choose the right oscilloscope for their needs. Unfortunately an error crept into the article which our friends at Rohde & Schwarz drew our attention to. Under the heading 'Interfaces for remote control' it states that Hameg uses the good old serial interface. Hameg oscilloscopes have for a number of years now been available with a USB interface. In fact here in the Elektor lab we have a number of different Hameg scopes with dual (RS232 + USB) interfaces giving the user freedom to choose whichever interface is best suited to their needs.

Err-lectronics

Corrections, Updates and Feedback to published articles

Platino, the Return

Elektor 2/2016 (March & April), p.62 (150555)



CORRECTION. Two errors crept into this article. First, in both the circuit diagram and parts list we refer to T1 as a BC547C transistor but unfortunately the PCB layout uses the outline for a 2N3902 type transistor. Both transistor types are suitable for this application but they do not have the same pinouts. The emitter and collector leads are swapped. The simplest solution if you are using a BC547C is to mount it on the display side of the board which effectively swaps the collector and emitter connections. Alternatively you can mount the transistor on the PCB component side so that it's rotated 180° to the printed outline.
To make the backlight permanently lit you can simply replace transistor T1 with a short length of wire to link the emitter and collector pads. The second error occurred in Table 1 (Platino solder jumpers). On line JP3 (LCD backlight), position 2 should read PC5 and not PC7. The schematic and PCB are correct.

Room Climate Controller

Elektor 5/2016 (September & October) (140345)

ADVANCE INFORMATION. With the project now released for publication in the English edition of Elektor, the author Goswin Visschers managed to gain more useful experience of the system operation. During the last two years of use the temperature and humidity sensor have inexplicably failed three times. He has subsequently modified the circuit to make use of a different sensor. The new sensor is the type DHT22 available from Adafruit; it has a similar specification and costs about a quarter the price of the original sensor. In order to use the new sensor a few mods need to be made to the board. The DHT22 does not use an I²C interface but has instead a One-Wire interface. To accommodate this, R13 must be replaced by a 10 kΩ resistor. The sensor's DATA pin now goes to the SDA pin of the connector (which connects to RC4 on the controller). The SCL pin is now unused.
It's also necessary to make changes to the software. The sensor data does not need to be converted in software any more which cuts down on the code. This allows another RS232-function to be included in the same program.
In the menu under the RS232 option to control output 1 you can use the commands 'O1=1[CR][LF]' and 'O1=0[CR][LF]' to turn it on and off respectively (ditto for output 02). The serial character sequence with the status information is sent out every second, regardless of which menu option is selected.
The latest software version will become available for download at the project web page, with the publication of the September & October 2016 edition.

Which 'Scope?

Elektor 3/2016 (May & June), p. 18 (150769)

FEEDBACK. I have a UNI-T DSO which has a tiny 320 × 240 display and it can be a little frustrating to use at times. Luckily the device also has a mini USB connector and sends out the samples via this interface so you can view them more comfortably on a PC monitor. I have written a program which allows you to view and scroll through over 10,000 samples. The vertical resolution is 9 bits. The program is written in Python and runs on a PC under Windows or Linux and also on ARM-based systems (ODROID, Raspberry Pi 2). I would like to make the code and associated documentation available to your readers. An example screenshot is included along with the documentation.
Hermann Hamann

Hi Herman
You made a neat job which takes advantage of the USB port capabilities of the UNI-T-DSO and certainly adds to the instrument's versatility. Many thanks for your offer; we will gladly pass your program on to our readers. It's now available for download from the web page relating to the original article (www.elektormagazine.com/150769).
Editor.

Retronics: Elektor High-Power AF Amplifier (1986)

Elektor 3/2016 (May & June), p. 124 (150738)

FEEDBACK. Dear Dr. Scherer, dear Jan, I was pleased to read in *Retronics* your reminiscences on your 1-kW audio amp design from 1986, published in Elektor Germany under the name *Gigant*. You are probably already aware that the name Gigant (Giant in English, *Ed*.) has a special place in audio power amp designs in Germany. Back in 1961 the company RIM, based in Munich introduced a design and a kit of parts to build a tube mixer-desk amp design which they called Gigant and featured in their yearly publication on DIY electronics. The design used an output stage with two EL34 power pentode tubes giving an output power of 30 watts which could be reduced to 12 watts by switching to a lower operating voltage. A few years later the lower power setting was raised to 15 watts but no changes were made to the circuit design. Sometime in the late 1960s the output stage design was changed to use a fixed grid bias rather than the automatic bias and the supply transformer rating was beefed up; the amp now called 'Gigant S' has an output power of 45 watts on tap. The possibility to switch to a lower power setting had by this time been dropped from the design. I am fortunate enough to own an example of this amp.
In 1969 the company Dynacord introduced their Gigant stage-amplifier producing a whopping 160 watts from four EL34s in the output stage. This was quite impressive for its day but the use of valves only extends to the amplifier's output stage, all the low-level signals are handled by transistors. I have four of these amps at home. Then of course we come to your own Gigant design offering 500 watts per channel. I haven't built this amp but was struck by the similarity of its design to the Crescendo amplifier which appeared in December 1982. It gave me the idea to boost the power of my own (otherwise standard) Crescendo amp by increasing its supply rail voltage. I added windings to the two toroidal power supply transformers to give me an extra 5 V on the secondary winding. The driver stage which was at ±70 V now has a supply (unstabilized, in contrast to your Gigant) of around ±77 V, giving an operating voltage for the Hitachi power FETs identical to the Gigant design. Lo and behold the result is that my Crescendo now delivers more than 300 watts per channel with low distortion and remains completely stable in operation!
So I would just like to thank you for your original 1000-watts amplifier design which although I didn't build myself, inspired me to tinker (successfully!) with my own amp.
Uwe Menrath

Hi Uwe,
Many thanks for the background information and historical perspective which we were not aware of. Thomas only started swinging a soldering iron back in the 70s when anything with more than three legs such as a tube was way too complicated for him ;-).
We are glad the trick you used to jack up the supply voltage on your Crescendo worked so well!
Thomas Scherer

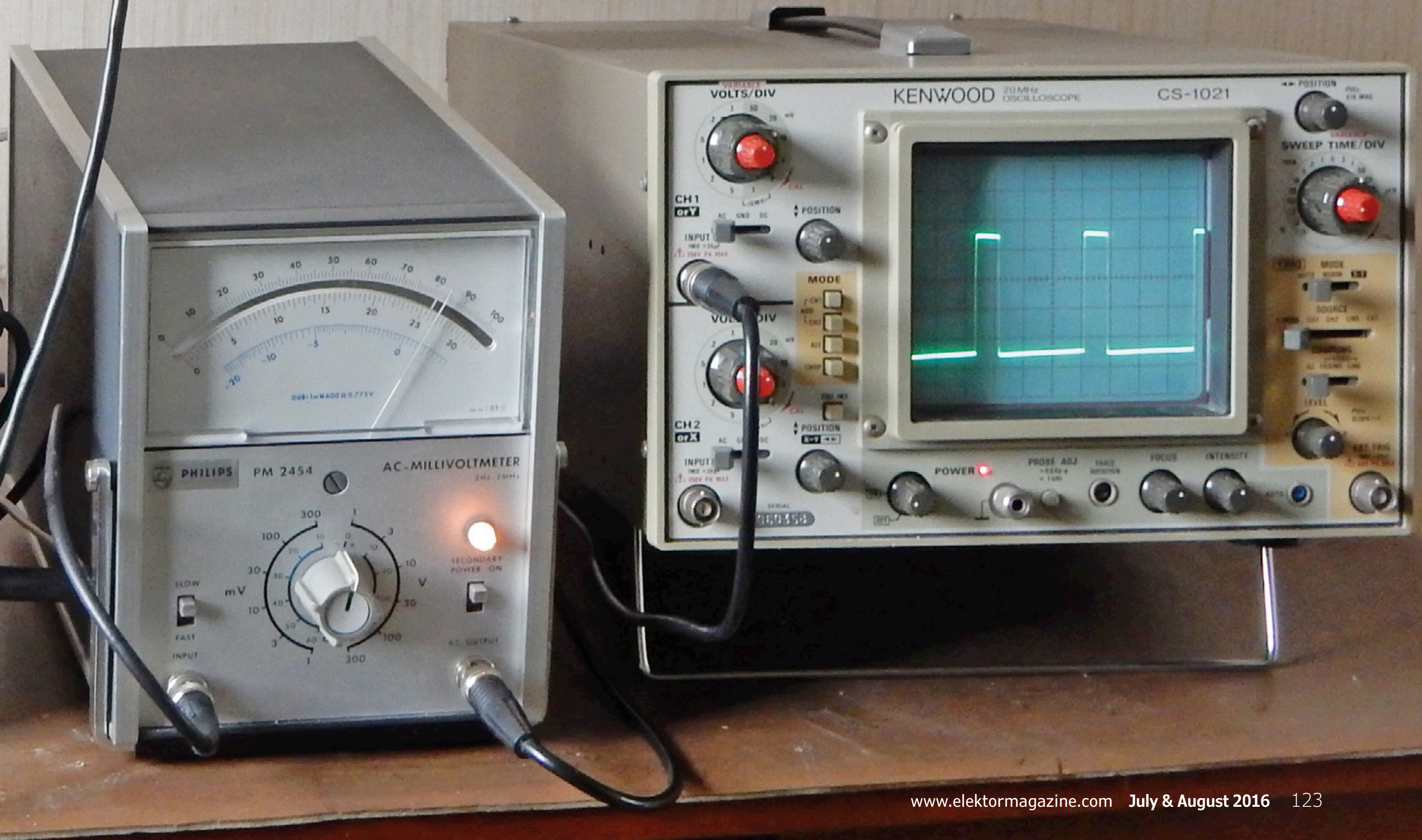
UITG.NIV.
~
I.M.
VERZWAKKER
dB
AANPASSING
Ω
LEK
UITG.
FREQ.
Hz
NET
0,8 A
NOS
T.D. RADIO
GEN. MEETSET 469
VERZWAKKER
ING.
INGANG
BELASTING
DRAAGGOLF I.M.
FUNCTIE
ECHT
HARM.
I.M.
BVM
0,8 A
NET
NOS
T.D. RADIO
ONTV. MEETSET 469

A Custom Audio Test Set (vintage 1969) from the NOS

In the post-war years, the *Nederlandse Radio Unie* – which later became the *Nederlandse Omroep Stichting* or *NOS* [Netherlands Broadcasting Foundation] – was a pioneer in making very high quality audio recordings, and in broadcasting those recordings without any loss of quality. Their attitude was that they were the source of the audio, and any form of distortion was unacceptable. This article aims to show that NRU/NOS electronic engineers were serious about the business of keeping that horrid 'D' as low as possible.

By **Theo Bouman** (Netherlands)

Striving for the lowest amount of distortion — 'D', or 'nonlinearity' in all-technical jargon — meant that the quality of the recorded and transmitted signals had to be checked on a regular basis. Sometimes that was not technically possible because the equipment they needed was not commercially available, at least not yet.
Even the manufacturers which supplied much of the equipment purchased by the NOS were not able to meet the NOS requirements in some cases. That stimulated a number of engineers to find their own solutions. The audio test set described here, developed between 1966 and 1969 and produced in NOS's own workshop, is a good example. To ensure the desired quality, all components of the test set were first measured and then selected to be within 0.5% of the specified value.
The author of this article, who was also the developer of the test set, worked in the Engineering Department of the NOS at that time as an instrumentation engineer and is now retired as well as an Elektor *Retronics* fan.



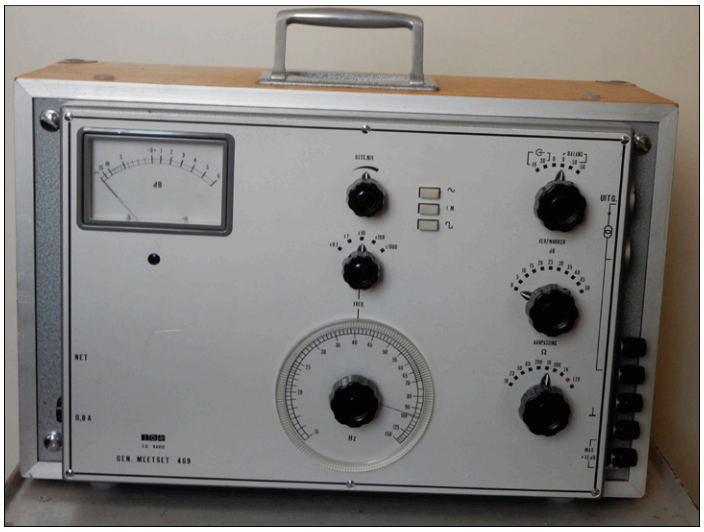


Figure 1. Transmitter set in original case.

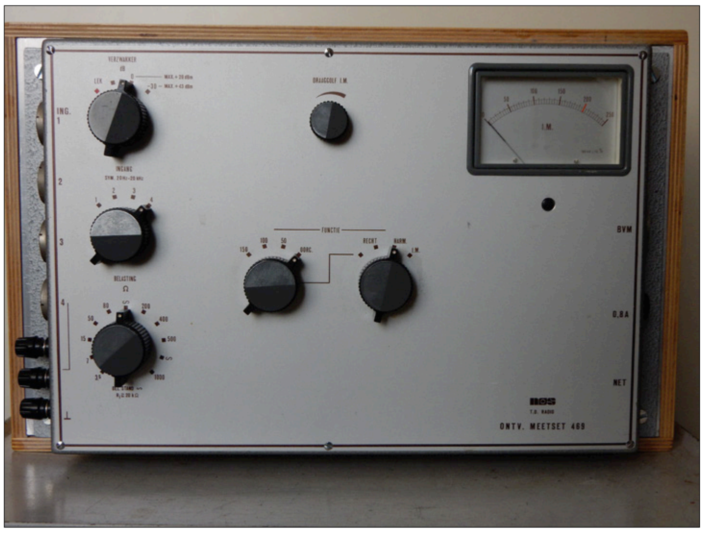


Figure 2. Receiver set, front view.

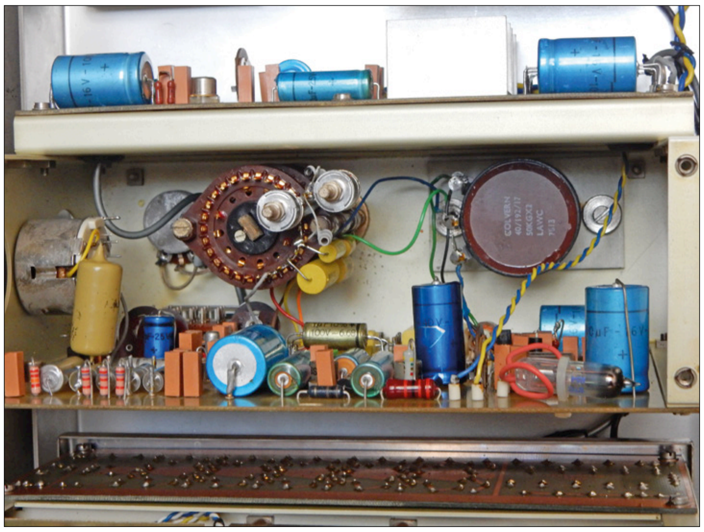


Figure 3. Transmitter set, rear view: the glass tube in the lower-right corner of the board is the thermistor, which is included in the feedback loop.

The first semiconductor-based test set

Along with several studios in Hilversum, the NOS maintained a number of broadcast studios at other locations in the Netherlands, including the Concertgebouw in Amsterdam, Studio Den Haag, the Kurhaus in Scheveningen, and the regional broadcast centers of Maastricht and Groningen. For setting up and measuring existing and newly installed audio systems in the various studios, they needed compact and convenient test equipment. The instruments in use up to the end of the 1960s were all tube based and therefore very heavy, and it was not easy to carry them from one location to the next. The new test set was the first of its kind to be made with semiconductor devices, which considerably reduced the weight.
It consisted of a transmitter(-side) set, a receiver(-side) set, an AC millivoltmeter and an oscillograph ('oscilloscope' in new money, *Ed.*).

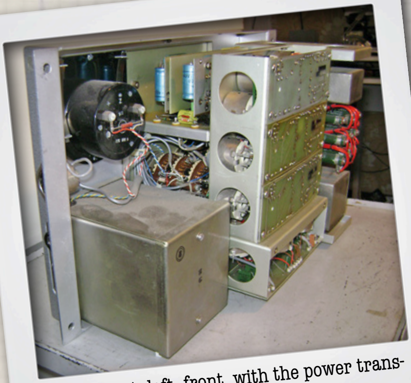
This portable test set was housed in four wooden cases: a transmitter case (**Figure 1**), a receiver case (**Figure 2**), a case containing the AC millivoltmeter and the oscillograph, and an accessories case for the necessary cables and test leads. The accessories case also contained two boxes of resistors and capacitors, which were intended to allow various adjustments to be made on site if the measurements indicated that things were not right. The cases measured 42 x 27 x 24 cm (width x depth x height). The transmitter case weighed about 11.5 kg, and the receiver case weighed about 13.3 kg. That was a big difference in weight compared to the previous test sets, which were at least 50 percent heavier. By the way, the oscilloscope in the introductory photo is not the original but instead a replacement. Unfortunately the original "oscillograph" was nowhere to be found, not even in the basement of the Video and Audio Museum, which prides itself on having an example of virtually every type of equipment. (possibly a Philips GM56xx or PM32xx; The Retronics Collection may have one, *Ed*.)
Another very practical feature of this audio test set was that it combined all the functions necessary for checking audio systems: frequency characteristic, gain, phase, and impedance matching.

Specifications and features

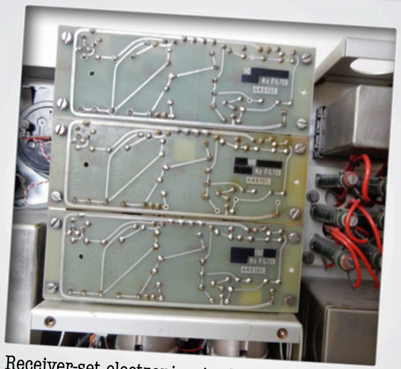
The transmitter set electronics (**Figure 3**) comprises a sinewave generator with a range of 15 Hz to 150 kHz, adjustable in five steps; a 100-Hz sinewave oscillator; a sine-to-squarewave converter; a precise step attenuator with 5-dB increments and a range of 50 dB; and a fixed 50-dB attenuator (for 100 dB in total). The attenuation network is balanced. To compensate for the low output level of the sinewave generator and to provide impedance matching and a balanced output, it has an amplifier and a transformer (see the schematic in **Figure 4**). The output circuit has an external impedance switch with selectable impedances from 10 Ω to 1 kΩ. There is also a two-level "leakage" setting. The transmitter set has a reference level meter calibrated at 0 dBm. This calibration was done very precisely for every set in use.
To allow the sinewave generator to be used for the various distortion measurements, after some experimentation we managed to reduce the distortion to about 0.02%. That was necessary because the amplifiers developed by the NOS labs had very low distortion levels, and the distortion of the measurement

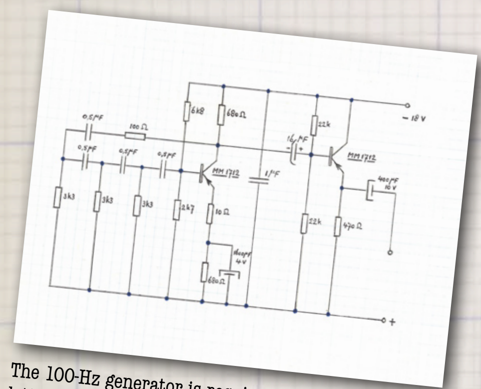
NOS 1960's Small Series Engineering in Pictures

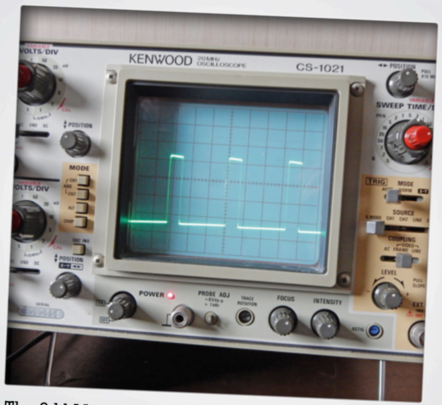
[By Jan Buiting] Where NOS stands for **Nederlandse Omroep Stichting**, rather than "new old stock"! This photo compilation aims to show the levels of craftsmanship in electronics and metalworking achieved by enthusiastic engineers at the Dutch NOS labs in the early 1960s. Remember, the NOS was a foundation so funds were not unlimited.
Quite noticeable is the frequent use of components in the circuitry that can be traced down to Philips Netherlands, specifically the famous blue electrolytics and the yellow polyesters (HiRel. Series — very expensive at the time). That's not surprising considering the NOS' base was in an area in Holland where most of the national broadcasting came from, popularly called the "Gooise Matras", with the Philips Telecommunication Industry works in Hilversum and Huizen at — shall we say — cycling distance. On the other hand, the transistors in the system are predominantly American (MM, 2N3704/06 etc.) and the precision pots, from Colvern UK.

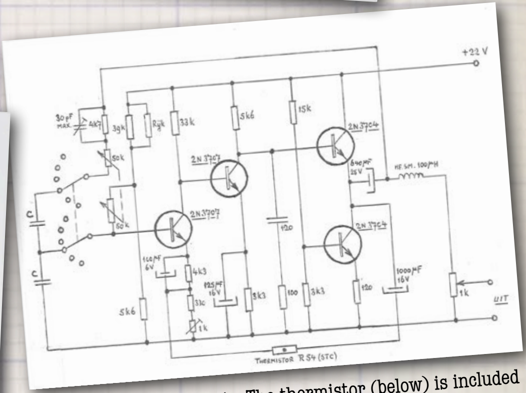

Receiver set: left, front, with the power transformer shielded in mu-metal.

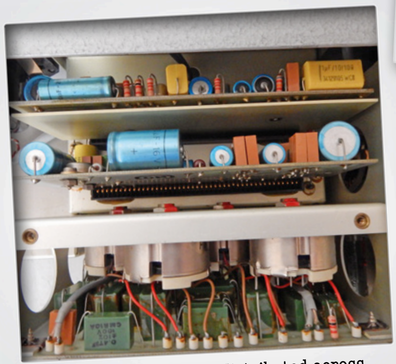
Receiver set, top view, with the function switches topmost.

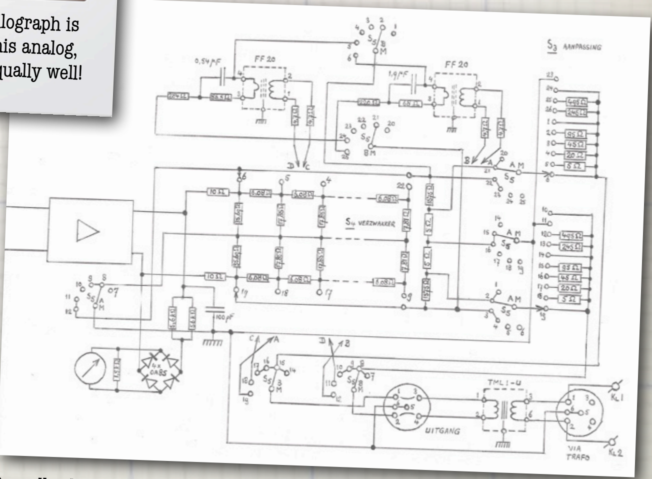

Receiver-set electronics: in the middle, the three hum filters for 50, 100 and 150 Hz.


The 100-Hz generator is required for the intermodulation (IM) measurement.


The Odd Man Out: the original 1960s oscillograph is sadly missing from the author's set but this analog, late 1980s Kenwood 'scope does the job equally well!


Tone generator schematic. The thermistor (below) is included in the feedback circuit. Not every thermistor did the job; it had to be specially selected to achieve the lowest distortion.


Receiver-set electronics distributed across printed circuit boards.


Overall schematic of Generator Case. Note the fully balanced circuitry in the attenuator.

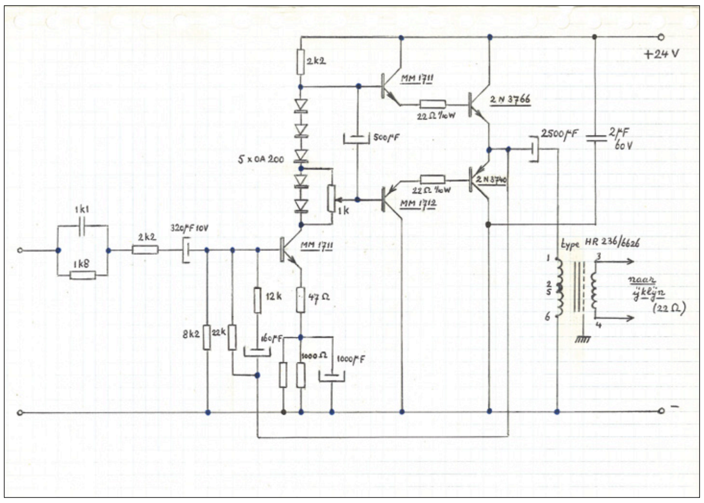


Figure 4. Tone generator amplifier: a specially developed circuit for raising the generator's signal to the right level and converting to suitably low output impedance.

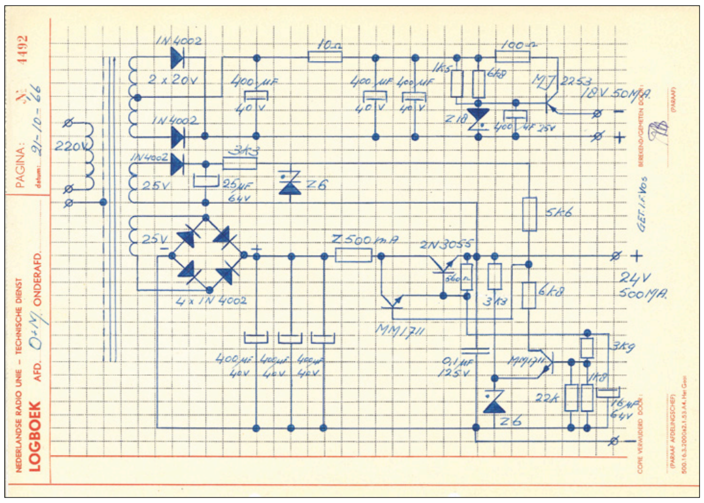


Figure 5. This schematic for the measuring-case power supply dates from 1966, i.e. the start of the development stage. The author's initials for approval can be seen; MB = M. [Theo] Bouman.

signal has to be lower than the distortion of the amplifiers to allow their distortion levels to be measured accurately.
The 100-Hz sinewave oscillator was needed for intermodulation measurements. Those were mainly done on mechanical recording devices, such as tape recorders and vinyl recorders. The sine-to-square wave converter produces an asymmetrical square wave, which was used for phase tests (see head photo again). That was one of the ways to check the phase of connections (microphones, input lines and output lines). Of course, there is also a power supply unit, with 220 VAC input and output voltages of 18 VDC and 24 VDC (see the schematic in **Figure 5**). The power supply unit is nicely enclosed in a metal box made from Mu metal to block the magnetic field. The step attenuator is also well shielded with Mu metal and other means. The signal generator has a second output from an isolation transformer, which is necessary for making proper measurements on unbalanced circuits.
The receiver set has four inputs with 'Tuchel' sockets (**Figure 6**) to mate with Tuchel plugs (**Figure 7**) and a selector switch for a load resistance of 3, 5, 7, 15, 80, 200, 400, 500 or 1,000 ohms, or infinite resistance. A 30-dB attenuator can be switched in for output levels above +20 dBm. The first selector switch is for measuring the noise level at 50 Hz, 100 Hz and 150 Hz. Measurements can also made with an A-weighted filter. The A-weighted filter gives an impression of the perceived noise level with the average human auditory response. The second selector switch (right-hand knob in Figure 2) has three positions: "Straight", "Intermodulation" and "Harmonic". The distortion measuring filter is designed for a frequency of 1 kHz and suppresses the fundamental frequency by 130 dB. The meter is intended for calibration with intermodulation measurements.

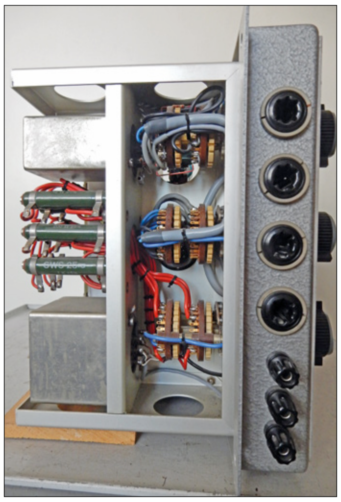


Figure 6. Receiver set, side view with the four Tuchel inputs.

Figure 7. Tuchel plugs.

Millivolts and decibels

All measurements were read from the Philips PM2454 AC millivoltmeter. The measurement results were recorded in decibels, which meant relative to the output level. For example, if the output level was +6 and the meter was set to "Straight", then the distortion level could be expressed as –50, –60 or –65 dB. The result was always relative to the measured level, which meant relative to +6 or +12 dB, or in the case of the power amplifier relative to +42 dB. If you wanted to express that in percentage, you had to work on the basis of –40 dB is 1%, –46 dB is 0.5% and so on.

Results

By about 1969 a total of twenty sets had been produced. They were also used by the audio instrumentation department of the television broadcasting group. In addition there were test sets in the various NOS instrumentation rooms (audio, AM broadcast and FM broadcast), so they were used as both mobile and stationary test sets. All of my fellow engineers at the time were very enthusiastic about the test sets. There were virtually no problems or faults with them during the years when they were in use (1969 to 1985). The set shown here, which still works perfectly, is apparently one of the few surviving examples from that era.

(160032)

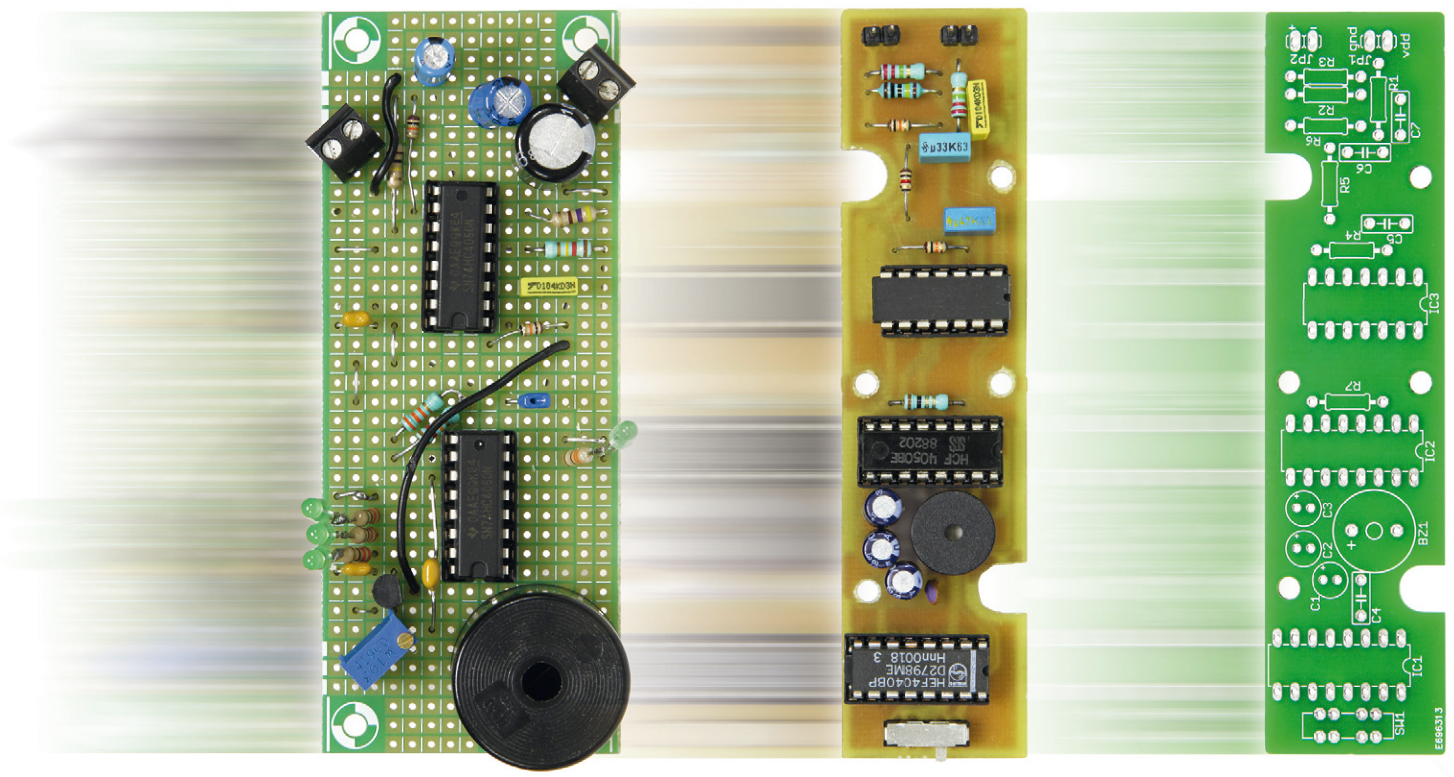
LEARN DESIGN SHARE

elektor PCB Service

offered in collaboration with EURO CIRCUITS

Generate your own PCB using the Elektor PCB Service

Affordable **High Quality** **Reliable**



The Elektor PCB Service is the most extensive fully customized service for printed circuit board production in Europe. With convenient online tools allowing you to visualize and analyze your design before you order and pay .

- For beginners, there is the **NAKED-Prototype Service**: This produces single and double-sided PCBs without solder masks.
- For a more advanced service, there is the **PCB Visualizer** that shows you how your PCB will look after production, with a **PCB Checker** performing a DRC for you and the **PCB Configurator** that lets you customize your order details.

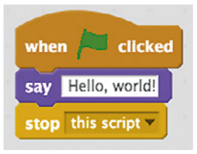
Smart menus and select options guide you through the ordering process. You can see in advance exactly what our machines can produce so there won't be any surprises!

So start your next project here:

www.elektorPCBservice.com

Compiled by **Robert van der Zwan**

Wake Up Australia!



Bill Morgan is an Electronic Design Engineer from Sydney, Australia. He also is a member of the Elektor family as long as he can remember. His message is: "Wake up Australia! We can no longer rely on digging resources out of the ground." Bill wants to start a nationwide initiative before resources and time run out. "Let's teach kids coding, starting when they are 5 or 6 years old." To achieve this goal, Bill is embarking on a new journey, starting from 'Scratch', a visual programming language that makes coding fun. We concur and will put his initiative in the spotlights shortly.

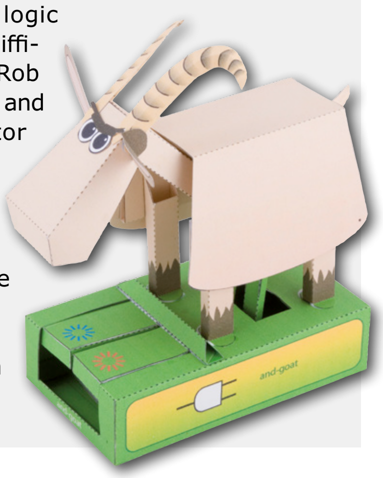
A Yearly Occurrence?

Elektor Magazine is known for its monthly editions in French and German, and the bimonthly ones, in Dutch and English. And sure enough there's also a weekly edition, our e-zine. Now, it's time to contemplate a yearly edition! We keep you posted of course. In the meantime, drop a line to editor@elektor.com if you have any suggestions.

READ ONLY MEMORY

Elektor magazine and its parent publishing company boast a long and rich history. In this space we picture a gem from the past.

Knowing the basics of a CPU or MCU is knowing something about microscopically small switches and logic gates. Switches are binary and either in the on or off position. However, the function of logic gates is somewhat more difficult to grasp. That's why Rob Ivens raised his Logic Goats and farmed them out in Elektor Magazine's July & August 2008 edition. For example, the AND Goat only nods his head if *both* buttons beneath his head are pressed. Want to know more about OR, NOT and XOR type Goats? You can still google 'logic goats'.



Aachen University and Innovative Electronics

RWTH Aachen University is working on an innovative way of teaching information technology (IT). This innovation is based on electronics still in development, meaning: on the basis of multiple objects that can be placed on a touchscreen the size of a table. It is here where Elektor Labs Germany comes in.

TABULA is an innovative project that revolutionizes the way students accumulate knowledge and competence about IT. The project combines a table sized touchscreen with so-called 'tangibles', i.e. objects that can be moved freely over the 'table' and containing electronics themselves. By moving the tangibles, the information on the touchscreen alters, hopefully yielding positive feedback on the 'table screen'.

The tangibles have other advantages too, like students being able to use more than one tangible at the time on the touchscreen, thus stressing multiple relations between separate objects. The so-called PERCS technology developed at RWTH Aachen University also tackles the problem of detecting objects permanently, not only when they are in motion ('PERCS': Persistently Trackable Tangibles on Capacitive Multi-Touch Displays).

"Using the touchscreen, students can work on concrete concepts that are commonly known in IT. For example, learning how to design an algorithm to insert objects into a binary tree, or a flow oriented way of programming filters used in image processing," says professor Jan Borchers from the Human-Computer Interaction Center (HCIC) at RWTH Aachen University.

PEOPLE NEWS ● The man giving Elektor Magazine its colorful look, Giel Dols, turned 60 last May, in Limbricht (and yes, each of these doors led to a cup of coffee and a local delicacy) ● Ferdinand te behalf of Elektor, a project with which RWTH Aachen University and Elektor want to revolutionize the IT ● Tom Verboven is developing Custom Relationship Management software so that we can serve our

Elektor Working on for Teaching IT

The German division of Elektor Labs are currently working on designing prototypes of the tangibles. The tangibles will come with some local intelligence, especially meant to relate their position to a central processing unit in the table, probably via Bluetooth LE. The tangibles are equipped with an optical sensor.

RWTH Aachen University would like to concentrate its work on the IT proper, especially data structures and flow-based programming. A renowned German e-learning company from Aachen, inside, is also an integral part of the TABULA team.

http://hci.rwth-aachen.de/tabula

which was noticeable on every door at our offices
Walvaart is responsible for the TABULA project on
way young (and older...) people learn the principles of
customers even better ... ...

EXPERT PROFILE

Elektor works closely together with more than 1,000 experts and authors for the publication of books, articles, DVDs, webinars and live events. In each installment of Elektor Word News we put one of them in the limelight.

Name: **Viacheslav Gromov**

Age: **16**

Education: **Highschool student**

Currently working on:
a solar-powered control unit for a greenhouse, culminating into a book on IoT

Who is Viacheslav Gromov?
I am a 16 years old Highschool student. Alongside my studies, I'm working as a freelance author on electronics. I've been writing magazine articles as well as books for a couple of years now, first and foremost on digital topics such as MCUs and FPGAs. But I am interested in the analog world as well. For example, at Elektor I published a course on ARM.

What made you start?
Both my grandfathers brought me into contact with mechanical stuff during my very early years in Russia. I got interested in electronics when I moved to Germany. When in Germany, I noticed an electronics store just around the corner or our house, called 'Wultschner Elektronik', where I could be spotted several times a day. In the store I could find all the components I needed. A real paradise! That's when I started to buy books on electronics, as well as visiting exhibitions.

What will be the most key electronics development?
It is my firm belief that research into microsystems based on biological, organic semiconductors will lead us to a breakthrough in terms of efficiency. The same goes for neural networks, which are way ahead of the current structures we use to process information. We have to bring ourselves to a new way of thinking. But when we reach that point, this new technology will bring us all kinds of advantages.

What makes Germany different from the US in terms of electronic innovation?
It is rather sad to acknowledge that German research as well as the German industry is increasingly lagging behind, both in terms of electronics and technology in general. As far as electronics is concerned, it is not difficult to find the centers of gravity right now. They are to be found in both Silicon Valley and Asia. This trend will continue in the foreseeable future...

Suppose you get $500 to buy stuff in the Elektor Store: what's it going to be? Why?
At any rate, I would get myself some books. But I would also get myself some FPGA boards, ranging from the ones to start with to Red Pitaya — which also can be used very well for developing a FPGA. ⏮

(160030)

Hexadoku The Original Elektorized Sudoku

Traditionally, the last pages of Elektor Magazine's SHARE section are reserved for non-engineering activities like reading about old equipment and solving a puzzle with an electronics slant: welcome to Hexadoku! Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of three Elektor book vouchers.

The Hexadoku puzzle employs numbers in the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

Correct entries received enter a prize draw. All you need to do is send us **the numbers in the gray boxes**.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for three Elektor Book Vouchers worth **$70.00 / £40.00 / €50.00 each**, which should encourage all Elektor readers to participate.

Participate!

Ultimately August 1, 2016, supply your name, street address and the solution (the numbers in the gray boxes) by email to:
hexadoku@elektor.com

Prize winners

The solution of Hexadoku installment 2/2016 (May & June) is: **ED54B**.
The €50 / £40 / $70 book vouchers have been awarded to: Guy Gaens (Belgium), Allan Drew (United Kingdom) and Ola Sandin (Sweden).

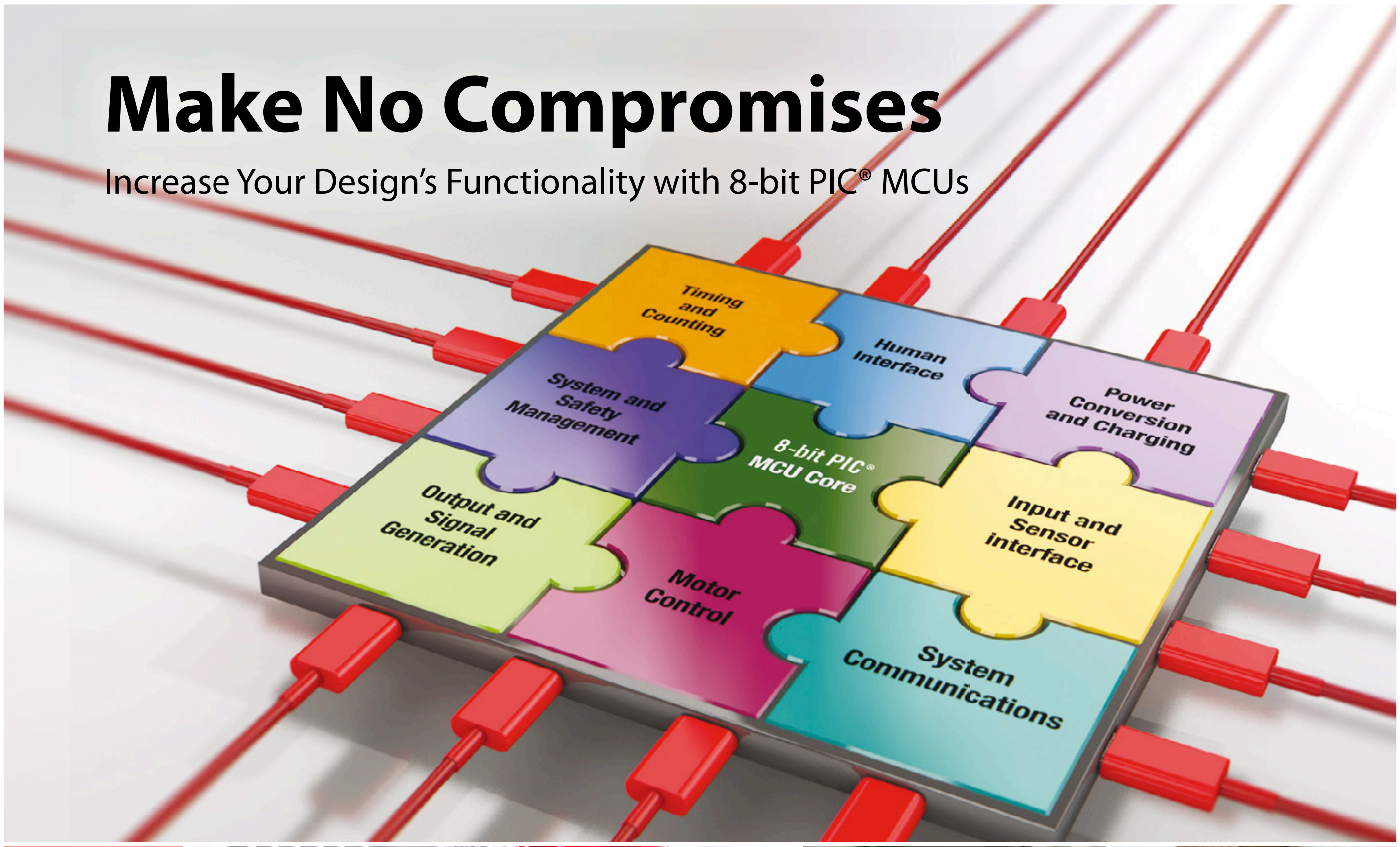
Congratulations everyone!

	5	B	C			6			3			A	F	7	
8	6			1	A					5	F			D	4
2				B	3					7	A				6
0	▨	▨	▨	▨	▨	4	5	8	1						3
	E	0		3		5			7		2		8	1	
	F	8			1					B			7	E	
9			6	7			2	A			5	3			0
			D			E			8			4			
			5			B			E			8			
3			1	0			6	4			C	E			A
	A	D			8					2			1	0	
	C	E		A		1			5		8		2	4	
D						3	E	5	A						1
6				9	7					3	E				C
E	3			F	6					4	0			5	2
	9	4	7			0			2			D	A	3	

1	6	4	9	2	5	E	F	3	7	8	0	B	C	D	A
F	A	D	7	B	3	9	6	1	C	E	2	8	4	0	5
E	0	B	C	8	1	4	7	9	A	D	5	6	F	2	3
2	8	3	5	A	0	C	D	F	4	6	B	E	1	7	9
4	C	F	0	9	2	3	5	A	E	B	1	7	8	6	D
3	1	9	B	7	E	6	8	0	F	5	D	4	2	A	C
A	D	E	8	C	B	F	4	2	6	3	7	5	9	1	0
5	2	7	6	0	A	D	1	4	8	9	C	3	B	E	F
6	F	0	A	D	4	5	9	B	2	1	3	C	7	8	E
7	3	5	2	E	F	1	A	C	D	4	8	9	0	B	6
8	9	1	E	3	C	2	B	5	0	7	6	A	D	F	4
B	4	C	D	6	7	8	0	E	9	F	A	1	3	5	2
9	7	A	F	1	8	0	2	6	3	C	E	D	5	4	B
C	B	2	1	4	D	A	E	7	5	0	9	F	6	3	8
D	E	6	4	5	9	7	3	8	B	2	F	0	A	C	1
0	5	8	3	F	6	B	C	D	1	A	4	2	E	9	7

Make No Compromises

Increase Your Design's Functionality with 8-bit PIC® MCUs


Timing and Counting
Human Interface
Power Conversion and Charging
System and Safety Management
8-bit PIC® MCU Core
Input and Sensor interface
Output and Signal Generation
Motor Control
System Communications

In embedded system design, reality demands that compromises are made at every phase. Tradeoffs between performance, functionality and cost often prevent you from bringing your best ideas to market. We believe there's a better way. That's why we've architected our latest 8-bit PIC® microcontrollers (MCUs) with flexible, "core independent" blocks of hardware intelligence that react quickly, consume very little power and are much more code-efficient than a software-based approach. In essence, Core Independent Peripherals help you easily combine many complex system functions onto a single MCU, increasing speed and flexibility, while reducing power consumption and cost. Design with 8-bit PIC MCUs, and you won't have to compromise.

Enable system functions with:

- Maximum flexibility
- Minimum latency
- Reduced cost

FLEXIBLE INTELLIGENCE MADE EASY
8-BIT PIC® MICROCONTROLLERS

microchip DIRECT
www.microchipdirect.com

MICROCHIP

www.microchip.com/8-bit

PROTEUS 8.3

ECAD to MCAD made easy

Data Exchange with STEP/IGES

The Proteus Design Suite now includes full support for data exchange with Mechanical CAD packages via the STEP/IGES file formats. This allows you to better visualise your design and helps quickly solve fixtures, fittings and casement problems.

Import 3D STEP/IGES models for your parts and visualise inside the Proteus Design Suite. Export your completed board to Solidworks or other MCAD software.

Visit www.labcenter.com
Tel:+44 01756753440 E-Mail info@labcenter.com

Labcenter Electronics Ltd, 21 Hardy Grange, BD23 5AJ

Labcenter Electronics www.labcenter.com