Edition 6/2019 | November & December 2019 | www.elektormagazine.com

elektor

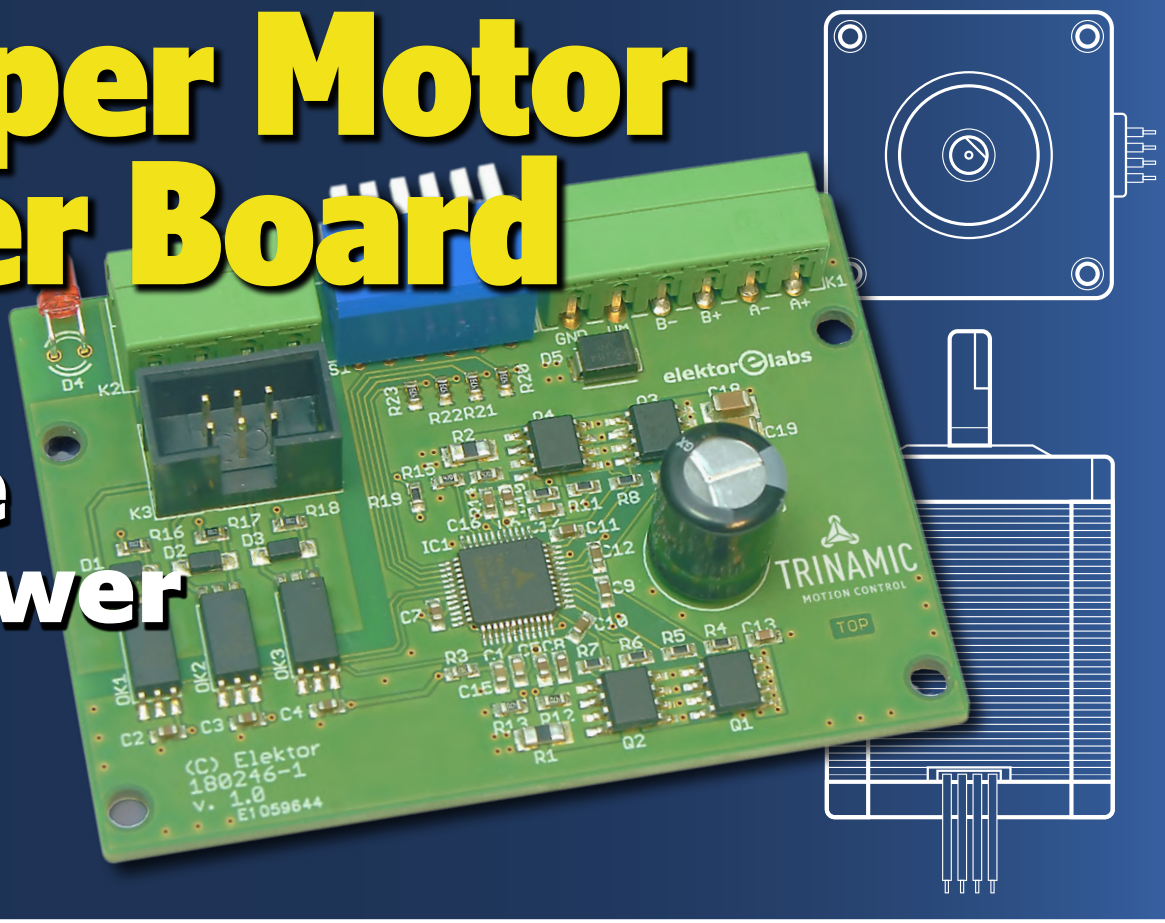
elektorlabs

DESIGN BUILD SELL ELECTRONICS

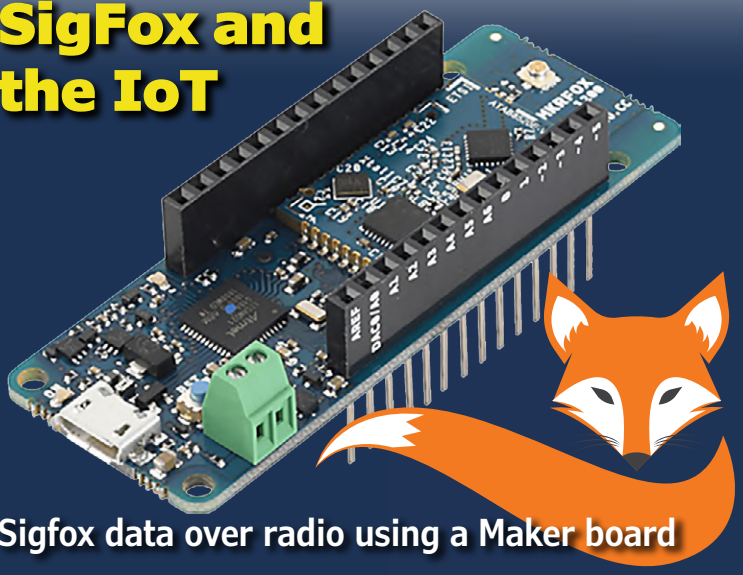
magazine

Stepper Motor Driver Board

Silence and power in one



SigFox and the IoT



Sigfox data over radio using a Maker board

Communication using UDP Packets

No sweat for the ESP8266



1-30 MHz Concept Receiver ⊳ A 3-Dollar Tiny Wi-Fi Switch That Works ⊳ 8-Channel Timer with OLED Display ⊳ Arduino Temperature Recorder ⊳ Baluns ⊳ Communication using UDP Packets ⊳ Elektor's First Video Game Console ⊳ The JesFS Flash File System ⊳ Having Fun with Development Boards from Infineon ⊳ Improved Radiation Meter — Improved! ⊳ Let There Be (Good) Light! ⊳ Raspberry Pi for Automation ⊳ Review: Raspberry Pi 4 ⊳ SigFox and the IoT (1) ⊳ Soft Start for Amplifier... and more

4K DISPLAYPORT SPLITTER

THE SIMPLEST PART OF YOUR PROJECT

DELOCK
we move the world

Mirror or expand the monitor signal to 3 displays in 4K resolution.

- Dual Mode capable (DP++)
- DisplayPort 1.4 Multi-Stream Transport
- supports HDR
- supports HDCP 1.4 and 2.2



BESTSELLER

Order no.: DELOCK 87737

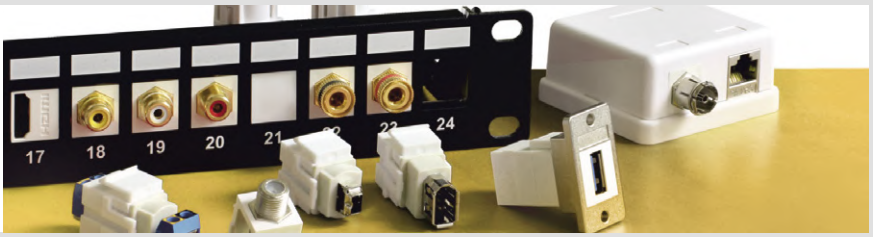
£ 90,39 (£ 74,71)

4K UHD

Our Keystone assortment from

DELOCK
we move the world

Discover now ▶ www.rch.lt/to



LoRa Antenna, 868 MHz

Omnidirectional, with tilt joint

- SMA plug
- Frequency range: 860 - 870 MHz
- ZigBee 868 MHz, Z-Wave 868 MHz
- Antenna gain: 3 dBi
- Impedance: 50 Ohm
- Efficiency: 45.91
- VSWR: 1,5





BESTSELLER

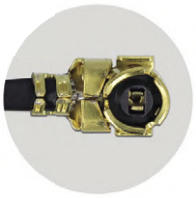
Order no.:

DELOCK 89769 **£ 9,46** (£ 7,82)

DELOCK
we move the world

LoRa Antenna, 863-928 MHz

- MHF plug
- LoRa, NB-IoT, ZigBee, Z-Wave (908, 921), ISM, GSM
- Antenna gain: 1.68 dBi
- Impedance: 50 Ohm
- Adhesive mounting
- Cable type: 1.13







BESTSELLER

Order no.:

DELOCK 12540 **£ 7,12** (£ 5,88)

DELOCK
we move the world

Payment Methods: VISA MasterCard Maestro PayPal

DAILY PRICES! Price as of: 25. 9. 2019

- Top price/performance ratio
- Over 110,000 selected products
- Reliable delivery - from Germany all over the world

Order Hotline: +44 203 808 95 25

www.reichelt.co.uk

reichelt elektronik – The best part of your project

The statutory cancellation provisions apply. All prices are stated in £ including statutory value added tax, plus shipping charges for the entire shopping basket. Our Terms and Conditions (available at https://rch.lt/TERMS or on request) apply exclusively. Actual appearance of products may differ from photos. We accept no liability for misprints or errors; prices subject to change. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Germany), Phone: +44 203 808 95 25

Elektorlabs Magazine
Edition 6/2019
Volume 45, No. 498
November & December 2019

ISSN 1757-0875 (UK / US / ROW distribution)

www.elektor.com
www.elektormagazine.com

Elektorlabs 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:
Margriet Debeij
Phone: +49 170 5505 396
E-mail: margriet.debeij@elektor.com

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

Copyright Notice
The circuits described in this magazine are for domestic and educational use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, DVDs, software carriers, and article texts published in our books and magazines (other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning and recording, in whole or in part without prior written permission from the Publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The Publisher disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from schematics, descriptions or information published in or in relation with Elektor magazine.

© Elektor International Media b.v. 2019
www.eimworld.com
Printed in the Netherlands



Do Try This At Home

I believe one of Elektor's first articles linking DIY electronics to 'kiddie' electronics was the Gameboy Digital Oscilloscope (GBDSO) from almost 20 years ago. It was a blast. Basically, you plugged a module into a Nintendo Gameboy game console and hey presto there's a simple portable oscilloscope. Initially we staged a Dutch-ish auction on the web to 'test the market', that is, check what price our customers were willing to pay for the product. Eventually the Elektor GBDSO reached sales volumes in the thousands across a period of more than ten years. If that GBDSO were an Olympic athlete, it would be in the company of Crescendo, EEDTs, Junior Computer, Filmnet Decoder, FT232-BoB, Pizza SMD Oven and Pico-C.
Since it will resonate widely within the e-community served by Elektor, a 'home modded' product [xyz] with enormous popularity is sure to attract weird questions from upright readers (mostly German), like "Where do I get hold of a known-good [xyz], with full warranty, in mint condition, at the lowest price?" to which we feel inclined to reply "Promise your kid(s) the new Upgrade Plus model", or "Dumpster-dive at the local thrift shop". Likewise, *Q*: "Did you contact the manufacturer on this?". *A*: "Techies don't normally do that, they read Elektor."
I would unhesitatingly recommend to all you electronicists out there to hack, fry, disembowel, blend, explore, rebuild or repurpose the tons of 'closed' electronics out there. It's cheap, often free and in plentiful supply. Do rescue stuff from skips and dumpsters filled by a generation that discards complete, plastic bicycle lights because of two exhausted button cells. The electronics inside, however trivial, is educational in providing clues to the design and construction process. If you like it, repair it. If you don't, formulate your criticisms and then Widlarize it — but safely and responsibly. Reverse engineering is good engineering. Do it creatively and with respect.

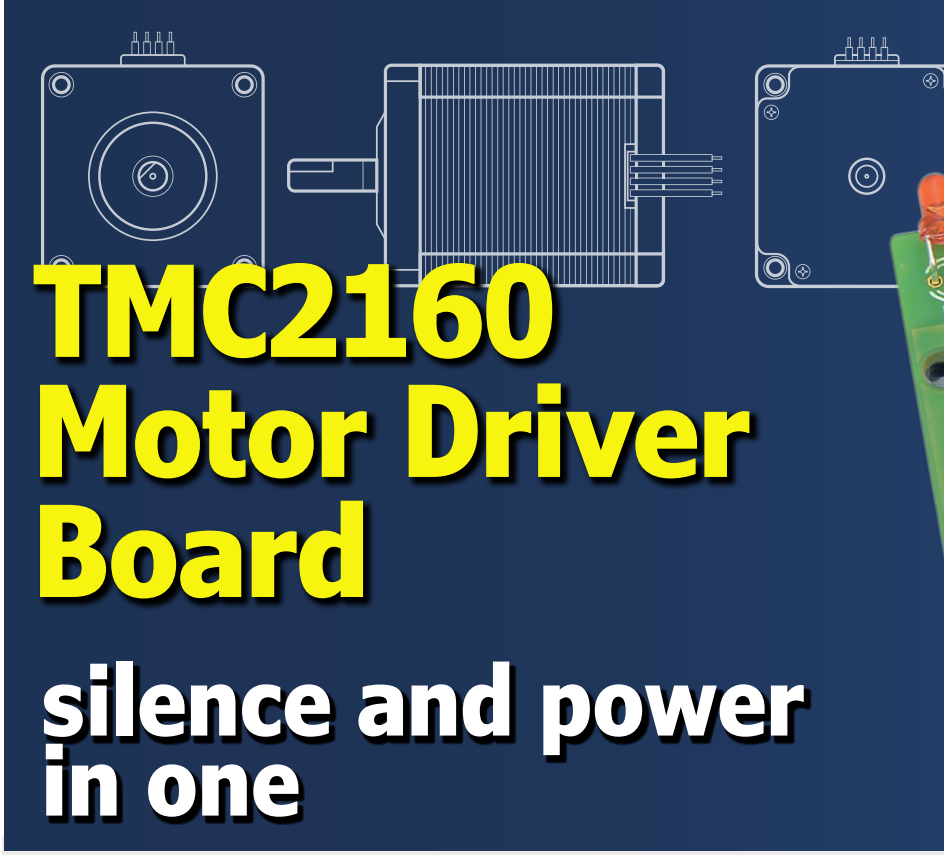
Jan Buiting, Editor-in-Chief

The Circuit

Editor-in-Chief:	Jan Buiting
Translators:	David Ashton, Jan Buiting, Martin Cooke, Ken Cox, Arthur deBeun, Andrew Emmerson, Mark Owen, Julian Rivers
Membership Manager:	Denise Bodrone
International Editorial Staff:	Eric Bogers, Mariline Thiebaut-Brodier, Denis Meyer, Jens Nickel
Laboratory Staff:	Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens, Jan Visser
Graphic Design & Prepress:	Giel Dols, Jack Jamar
Publisher:	Don Akkermans

This Edition

Volume 46 – Edition 6/2019
No. 498 November & December 2019



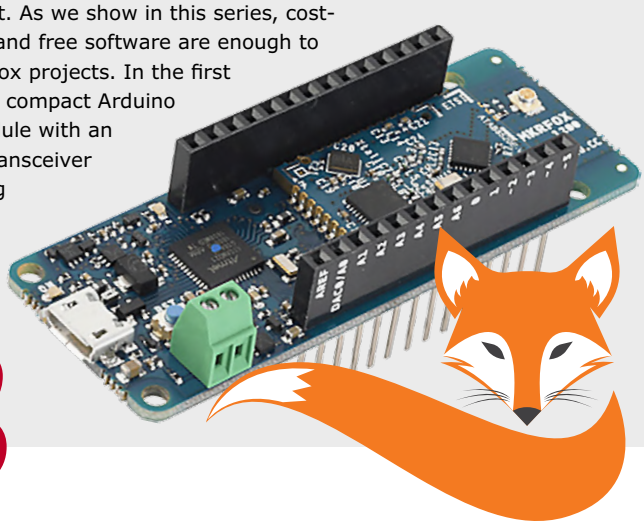
TMC2160 Motor Driver Board

silence and power in one

SigFox and the IoT (1)

Part 1: Sigfox data over radio using a Maker board

Wireless standards such as Sigfox and LoRa are available whenever battery-powered sensors need to transmit data to an Internet gateway. Sigfox scores high with a uniform and well-developed network. All the user must do is program and log on the sensor nodes in order to visualize their output data on the Internet. As we show in this series, cost-effective hardware and free software are enough to realize the first Sigfox projects. In the first part we present the compact Arduino MKR FOX 1200 module with an integrated Sigfox transceiver and a corresponding motherboard.



28

Regulars

15 Baluns
Peculiar Parts, the series

35 Elektor Labs Pipeline

48 HomeLab Helicopter
Wondrous Things Electronic
Spotted From Above

64 Err-lectronics
Corrections, Feedback and Updates to published articles

101 Retronics
Elektor's First Video Game Console — Restart after 42 years

106 Elektor Store Highlights
Earth Listener — Function Generator — Lab Power Supply

110 Elektor Store
what's available @ www.elektor.com

114 Hexadoku
The original Elektorized Sudoku

Features

36 Having Fun with Development Boards from Infineon

40 Let There Be (Good) Light!
Quality criteria for room lighting with LED lamps

58 Raspberry Pi for Automation
Maker-System powers end-products

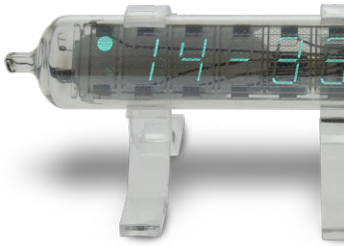
96 Review: Raspberry Pi 4
All new but still good?

Projects

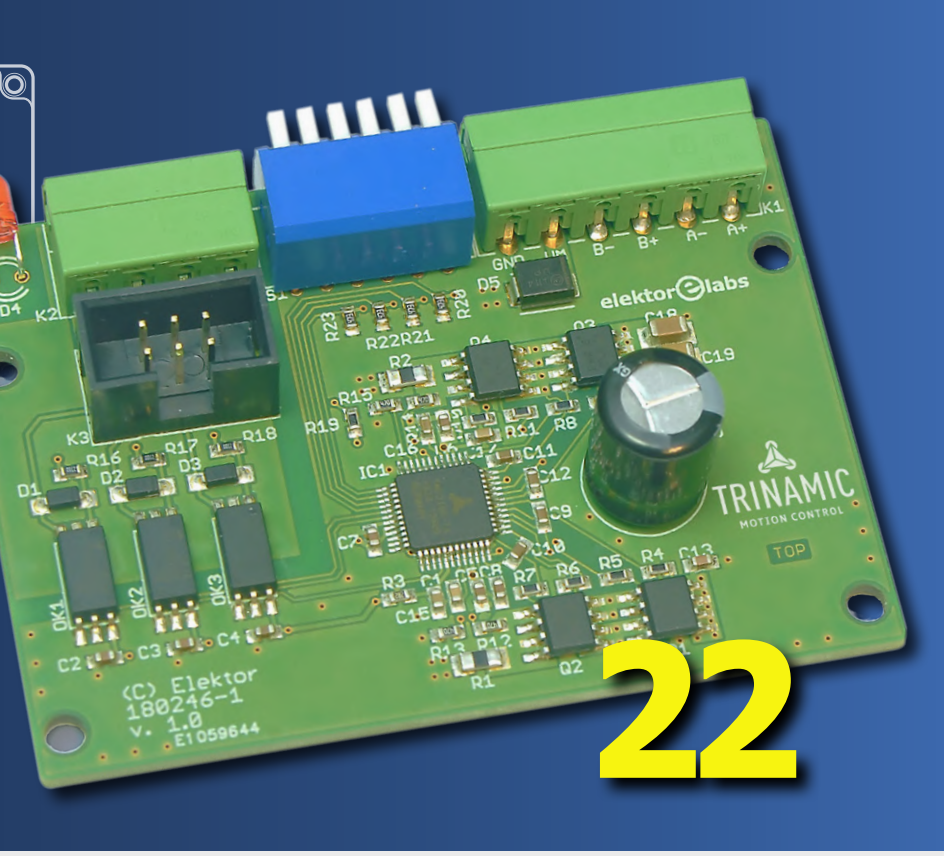
6 Soft Start for Amplifiers
Saves fuses and electronics

16 Arduino Temperature Recorder
Log around the clock!

19 Sound Blink
Prototyping a portable DIY speaker with LED light show



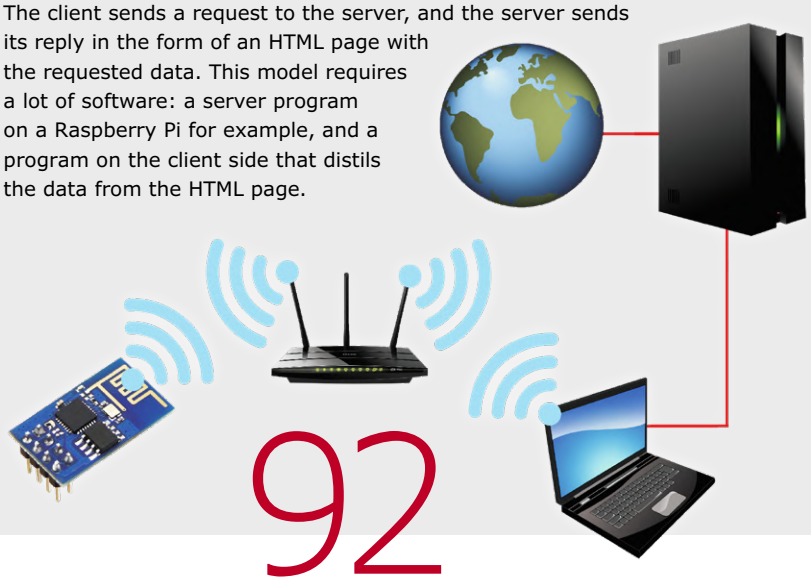
VFD Tube Clock
Old wine in new (vacuum) bottles



Communication using UDP Packets

No sweat for the ESP8266

To exchange data over a network, a client/server model is usually used. The client sends a request to the server, and the server sends its reply in the form of an HTML page with the requested data. This model requires a lot of software: a server program on a Raspberry Pi for example, and a program on the client side that distils the data from the HTML page.



70 VFD Tube Clock
Old wine in new (vacuum) bottles

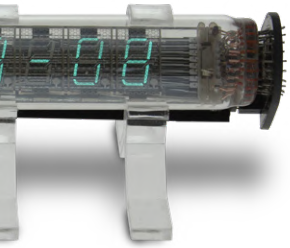
77 A 3-Dollar Tiny Wi-Fi Switch That Works
Creative dealings with the ESP8266

80 The SCCC Project (5)
Homebrew soft-core processor and C compiler

84 1-30 MHz Concept Receiver
AM/SSB/CW, with plug-in resonators, filters or crystals

92 Communication using UDP Packets
No sweat for the ESP8266

108 8-Channel Timer with OLED Display
Revised with easier to obtain parts



22 TMC2160 Motor Driver Board
Silence and power in one

28 SigFox and the IoT (1)
Part 1: Sigfox data over radio using a Maker board

52 The JesFS Flash File System
BIG DATA 4 small controllers

66 Improved Radiation Meter — Improved!
A hardware update for the alpha, beta and gamma counter

Next Editions

ElektorLabs Magazine Edition 1/2020

Geiger Counter Board • Speed Control for AC Motors • Send Data with Sigfox • A Graphics Library for Microcontrollers • ESP32 Multitasking • The New Watt Scale • First Steps with RISC-V • How to: Pushbutton Debouncing • Review: Thermal Imaging Camera • Electronics for Beginners Course • Tea Timer • Small Circuits & Tricks.

ElektorLabs Magazine edition edition 1/2020 covering January & February 2020 is published on 9 January 2020. Delivery of printed copies to Elektor Gold Members is subject to transport. Contents and article titles subject to change.

Elektor Industry Edition 3/2019

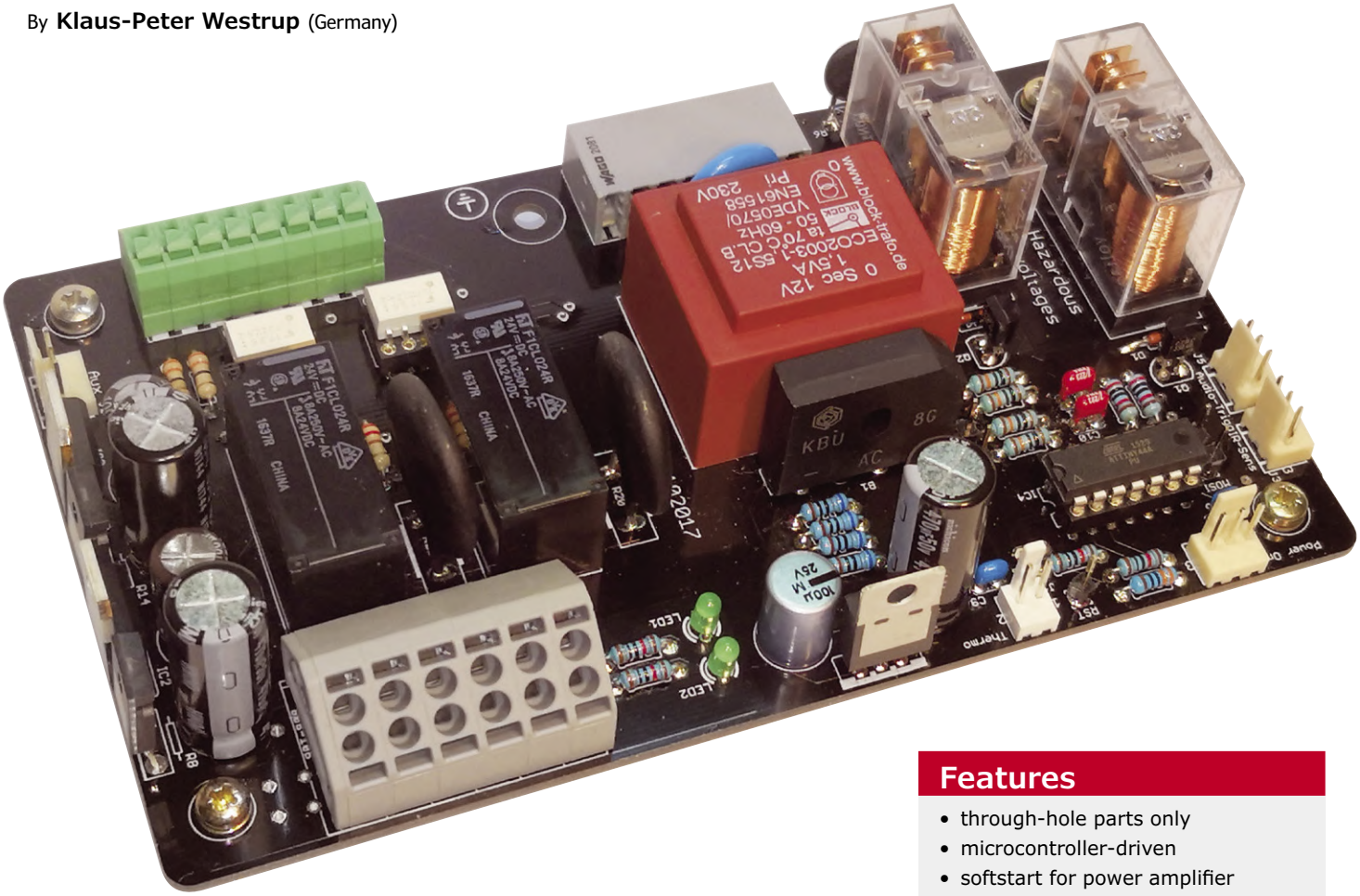
Elektor Industry issue 3/2019 is a special edition for the **Productronica 2019 trade show in Munich, Germany (November 12-15, 2019)**, and will focus on a range of innovative technologies, research, hardware and software, including IoT, 5G, sensors, CAD, SMT, PCB manufacturing, and more. Plus, you'll find fresh instalments of the magazine's regulars like Infographics and Industry Store.

Elektor Industry issue 3/2019 will be published early November 2019 to Elektor Gold members in print and to Elektor Green members as a PDF download. The edition will be available for purchase at www.elektormagazine.com.

Soft Start for Amplifiers

saves fuses and electronics

By **Klaus-Peter Westrup** (Germany)



Features

- through-hole parts only
- microcontroller-driven
- softstart for power amplifier
- IR remote control
- automatic on and off switching

No worries, it's happened to everyone: you switch on an electrical device and the mains fuse blows. In most cases that does not mean the equipment in question is faulty; the cause is usually much more prosaic: the switch-on current surge made the fuse melt prematurely. This inrush current (or surge current) occurs in traditional power transformers as well as in switch-mode power supplies, although it has different causes in either type of power supply.

In the case of power supplies having a power transformer, the peak inrush current is caused by the magnetic core saturation, while in the case of switching power supplies (SMPSU) it is caused by the discharged DC link capacitor.

The inrush current is very high and taxes many parts. It is therefore not surprising that devices and/or their power circuits almost always break down at the time of switching on — just like the death of an old-fashioned (and now banned) incandescent lamp the instant you flip the switch.

Manufacturers of electrical and electronic

equipment (and readers of ElektorLabs magazine) are all too familiar with this phenomenon. And the remedy is also known — somehow the inrush current must be limited.

Power to the power transformer!

The higher the induction in the iron core of the transformer, the smaller the air gap, and the smaller the copper losses, the higher the current surge. Contrary to what you may think, this peak current is the highest when the mains voltage is switched on exactly at the time of the zero crossing.
The reason is revealed by a little maths. Once switched on and all switch-on symptoms over, the phase shift between the mains voltage and the magnetic flux is 90°, as described by

$$\phi(t) = U_{\max} \int \sin(\omega t)\, dt + C$$

This equation is derived from Faraday's Law. It follows that:

$$\phi(t) = -\frac{U_{\max}}{\omega} \cdot \cos(\omega t) + C$$

The magnetic flux thus reaches its negative maximum at the zero crossing of the mains voltage. In continuous operation of the transformer, the constant C equals zero.
However, this is not the case at the time of switching on, because the magnetic flux must start at zero (provided that the iron core does not have residual magnetism). So when we switch on the transformer at time t = 0, the starting condition of the magnetic flux is defined by the constant C, and that constant itself depends on

- the value and polarity of the iron core's residual magnetism;
- the phase of the applied AC voltage;
- the properties of the core material.

In summary, we can write:

$$\phi(t) = \phi_{\max} \cdot \sin(\omega t + \alpha) + \phi_{rest} + \phi_{\max} \cdot \sin(\alpha)$$

with $\alpha = \theta - \frac{\pi}{2}$

Assuming that φ_{rest} = 0, then we find:

- when we switch on the mains voltage at the moment that it is maximal: $\varphi = \varphi_{max}$;
- but when we switch on the zero transit: $\varphi = 2\text{-}\varphi_{max}$.

This double size magnetic flux sends the transformer core in (magnetic) saturation, so that the inductance of the coil drops almost to zero, leaving the current to be limited only by the ohmic resistance of the winding and the power cord.
The switch-on current peak lasts for a maximum of 10 ms. In practice, we can expect a time span of 3–6 ms because the core does not go into saturation abruptly but takes some time to do so. However, we should not forget that it takes about 4–10 periods of the mains voltage before the magnetic flux — and with it the coil current — have assumed their 'stationary' values.

Possible solutions

In order to avoid the problem of large inrush current surges, we could somehow try to switch on the transformer only when the mains voltage has reached its peak value — then we avoid the wretched inrush peak.
That by itself is correct but for it to function properly, the residual magnetism of the core at the switch-on time must always be zero, and that is something that cannot be guaranteed in practice. However, there are circuits that measure this residual magnetism and can operate without current limiting resistors. However, these circuits are quite complicated.
Another possibility is to dimension the transformer in such a way that it does not reach saturation so quickly. This works, but leads to higher costs and greater electrical losses, while also increasing the size of the transformer, which in most cases will be undesirable.
For practical reasons we leave the inrush current peak for what it is. After all, it doesn't damage the transformer due to its short duration — when we're not talking about extreme powers here. We solve the problem through inrush current limiting. As we will see, the process can be carried out both passively and actively.
Table 1 gives an impression of the inrush currents that occur when transformers are more or less heavy. Clearly some form of inrush current limitating is necessary from powers of about 300 VA onwards in order to prevent a fuse exchange circus to put it that way.



Passive inrush current limiting

For passive inrush current limiting, 'fat' resistors or thermistors are usually applied. Thermistors are simply resistors whose resistance depends on the temperature. We all know them as NTC and PTC resistors.
When it comes to the limiting of inrush currents, only NTC resistors are consid-

Table 1. Switch-on current surge for various transformers

Power (VA)	Mains voltage (V)	Copper resistance (Ω)*	Inrush current (A)
200	230	8	36
300	230	5	54
500	230	3	81
800	230	1.5	130
1000	230	1	163

* Approximate resistance values

TDK

Inrush current limiters	B57236S0***M0**
ICLs	S236

Electrical specification and ordering codes

R_{25} Ω	I_{max} (0...65 °C) A	C_{test} 1) 230 V AC µF	C_{test} 1) 110 V AC µF	R_{min} (@ I_{max}, 25 °C) Ω	Ordering code
2.2	6	200	800	0.057	B57236S0229M0**
2.5	5.5	200	800	0.067	B57236S0259M0**
3	5	300	1200	0.077	B57236S0309M0**
4.7	4.6	300	1200	0.102	B57236S0479M0**
5	4.5	300	1200	0.108	B57236S0509M0**
8	3.7	300	1200	0.152	B57236S0809M0**
10	3.5	300	1200	0.180	B57236S0100M0**
12	3.2	300	1200	0.215	B57236S0120M0**
16	2.9	300	1200	0.262	B57236S0160M0**
20	2.8	300	1200	0.284	B57236S0200M0**
25	2.5	300	1200	0.357	B57236S0250M0**
50	1.9	300	1200	0.594	B57236S0500M0**
80	1.6	400	1600	0.861	B57236S0800M0**
120	1.5	400	1600	0.990	B57236S0121M0**

Figure 1: Specifications of TDK's S236 series thermistors.

ered, i.e. resistors with a negative temperature coefficient. They have a higher resistance in the cold state than in the warm state.

In the case of a purely passive inrush current limiting, the thermistor is in series with the load. When switched off (i.e. at room temperature), the resistance is such that the inrush current is limited to a safe, manageable value. As a result of the current passing through the thermistor, the device heats up (it dissipates power) and its resistance decreases to ultimately no more than a few ohms and the current reaches its nominal (stationary) value. This solution with thermistors is generally considered to be reliable and cheap.

Important parameters in the selection of the thermistor are the initial resistance (in the 'cold' state) and the current carrying rating. The value of the resistor should be chosen to limit the current flowing through the load to a value which does not yet make the fuse work (and which cannot damage the load).

In the following, we assume a 500-VA power transformer for dimensioning, which is fused with I_{smax} = 10 A by a device fuse present in the device in question.

We calculate the cold resistance of the thermistor from:

$$R_{NTC} = \frac{U_{eff} \cdot 1,414}{I_{s\max}} - R_{trafo} =$$

$$\frac{230V \cdot 1,414}{10A} - 3 \quad = 30$$

The cold resistance of our thermistor should be 30 Ω. Now we have to calculate the current flowing through the thermistor in stationary (continuous) operation (we assume that the transformer is under maximum load):

$$I_{NTC} = \frac{vermogen}{U_{eff}} = \frac{500VA}{230V} = 2,2A$$

For our 500-VA transformer we arrive at 2.2 A.

Finally, we need to estimate the maximum energy the thermistor has to convert into heat immediately after being switched on. Assuming that the switch-on pulse is approximately rectangular-shaped, then the product 'power times time' corresponds to the amount of energy. We find:

$$E = (R_{NTC} + R_{trafo}) \cdot I^2 \cdot t =$$

$$(30 \quad + 3 \quad) \cdot (10A)^2 \cdot 6ms = 20J$$

The data sheet of the TDK S236 series of thermocouples (see **Figure 1**) does not mention this maximum allowable energy as such. Instead, we find a maximum allowable load capacitance Ctest that can be connected to the current limiting thermistor. In the case of switching power supplies, the value of the DC link capacitor may be used for this purpose. We then calculate:

$$E = \frac{1}{2} \cdot 300 \quad F \cdot (230V \cdot 1,414)^2 = 15,8J$$

The load rating of a single thermistor is too low; we have to switch to a 'beefier' one or connect two thermistors in series (note: thermistors must not be connected in parallel). We have chosen for the latter option: two 16-Ω thermistors in series.

Disadvantages

Although passive inrush current limiting is often found in consumer electronics, there are some disadvantages to it.

In the first place: although the warm resistance of the thermistor is only 1/20 to 1/10th of the cold resistance, it still dissipates a considerable amount of power, in our case at least about 8 W, and this is of course undesirable for environmental reasons alone. In addition, the thermistor heats up: at nominal load, its temperature increases to (in our case) no less than 150 °C.

Consequently the device in question must not be switched on again immediately after it's been switched off: The thermistor quickly takes a minute or more to cool down to such an extent that it can once again act as a limiter.

Parts and soldering joints in the immediate vicinity are also affected by this heat buildup; especially plastic parts must be mounted at a sufficient distance from the thermistor(s).

A relay provides a solution

In **Figure 2** we have outlined a commonly used (active) solution to the power dissipation problem. We see here a power supply with upstream thermistor. A relay contact that is closed shortly after switching on is connected in parallel with the thermistor. The thermistor is then bridged so that no more power is dissipated in it. Of course, some power is lost in the relay (about 1 W) but that is much less than the minimum 8 W for the version without the relay.

The switch-on electronics are often powered by a small auxiliary transformer. This disconnects the switch-on electronics from the device power supply, so that even a short power failure leads to a safe reset of the switch-on electronics. Another advantage of this design is that the power switch of the device can be replaced by a low-voltage switch through a second relay.

And now, in practice

After all these (hopefully interesting) theoretical considerations it's time to roll

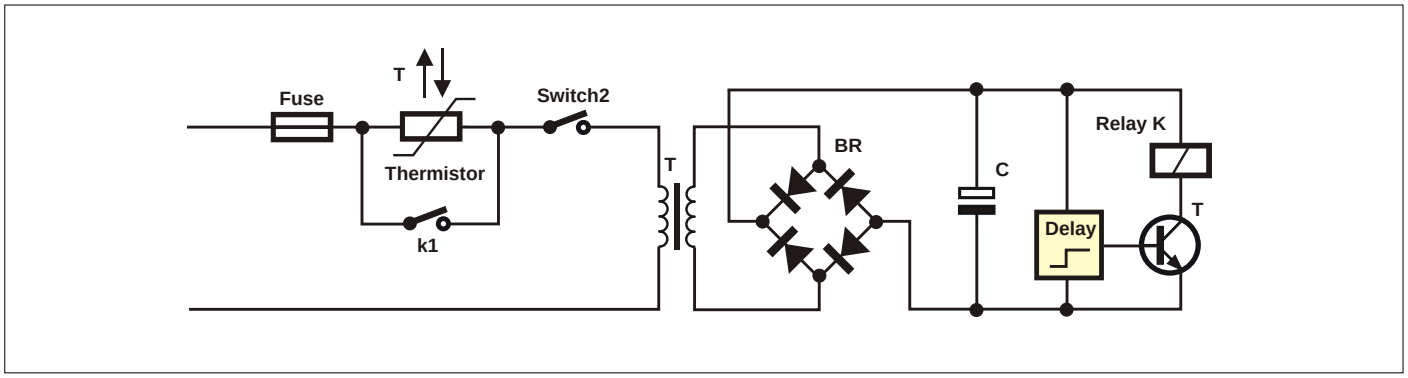


Figure 2: Principle of a softstart circuit using a thermistor and a relay.

up our sleeves. The soft start circuit in this article was originally designed for an audio power amplifier with a switching power supply. Admittedly, opinions on the use of switching power supplies in audio amplifiers are divided, but a. SMPSU has some important advantages over a classic linear power supply with a toroidal transformer. On the one hand, the efficiency is higher than with a linear power supply, and on the other hand, the smaller dimensions permit building smaller and lighter power amplifiers, which can also easily be built into a loudspeaker case Another advantage is that the output voltage is controlled, ensuring a safe maximum value of the output voltage is not exceeded or undershot, even with varying loads. And if the topology of the controller is chosen correctly, it is possible to obtain an extremely 'clean' supply voltage.

Special switching power supplies are now being developed for audio applications [1].

Block diagram

All considerations eventually led to a softstart unit with microcontroller supervision. A microcontroller offers an excellent opportunity to realise a number of expansions without too much hardware fuss, such as an audio-triggered autostart function which turns off the amplifier when no audio signal is present for a while, and which automatically switches on the amplifier when an audio signal is available. There is also an infrared receiver allowing the volume to be adjusted (via a motor potentiometer). **Figure 3** shows the block diagram of the final circuit.

The heart of the circuit is formed by the microcontroller — and although it is prominently present, in practice a relatively modest ATTiny44 proved enough. The mains voltage comes from the left inside and powers the softstart unit through its own small transformer. At the same time this mains voltage is conveyed, via a relay/thermistor circuit, to the actual (switching) power supply of the amplifier shown in the upper right-hand corner.

This provides the power supply voltages V1+ and V1– for the amplifier which are additionally protected via relay/thermistor combinations. From this the voltages V2+ and V2– are derived in a conventional way, ready for powering a preamplifier. In addition, the power supply also provides an auxiliary voltage that's used to control the motor potentiometer.

The detailed schematic

...is pictured in **Figure 4**. The various components of the block diagram are easy to recognize. The top left hand area(K9, K1 and K11) is the power part. The mains voltage for the power supply of the softstart unit itself is connected to K1. The switched supply for the switched-mode power supply arrives via K11.

K9 has no real function but only serves to comfortably loop through any protective earth (PE) line.

The power supply voltages for the amplifier provided by the switching power supply arrive at K3 as V+, GNDA and V–. The well-known linear regulator pair 7818/7918 (IC2 and IC3) produces an extra stabilized supply voltage of ±18 V

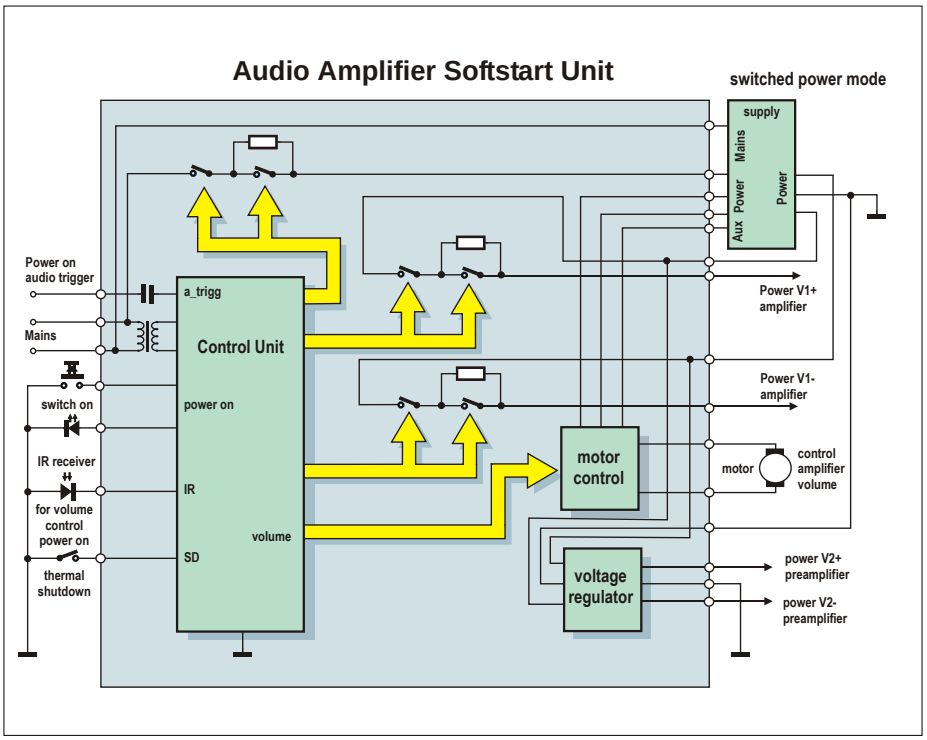


Figure 3: Block diagram of the softstart unit.

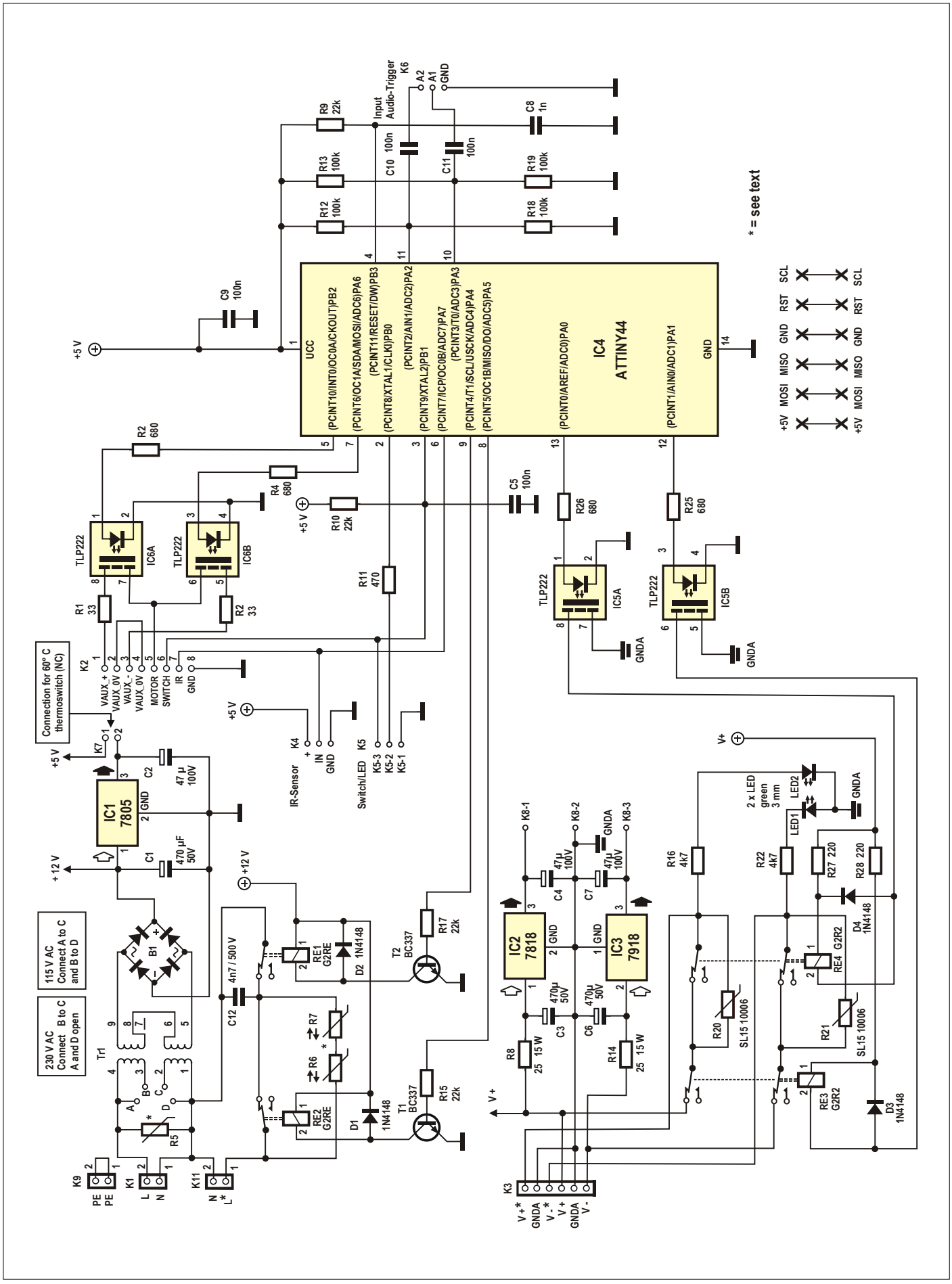


Figure 4: The detailed schematic.

for a preamplifier if present. V+ and V– pass through the relays R3 and R4 to the amplifier as V+* and V–*. The two green LEDs serve as status indicators.
The amplifier power supply used by the author also provides an auxiliary voltage VAUX to control the motor of the motor potentiometer for volume control. This voltage comes in at connector K2 and is transferred to the motor by the two semiconductor relays in IC6 as needed. The manual on/off switch (push button) is connected to K5 together with an on/off LED. The IR receiver is connected to K4.

The IR receiver

The option to connect an IR receiver and control the amplifier remotely is of course very attractive. However, it turned out that there were a couple of pitfalls. The main problem is of course which remote controls (i.e.: which data protocols) the software should be compatible with. There are numerous IR remote controls in circulation, as well as numerous variations on the IR protocol.
For this reason, the author abandoned the idea of using a specific remote control and tuning the software accordingly. Instead, the software for the softstart unit has been designed so that it can — within certain limits — learn some commands from commonly used IR remote controls.
Celebrating the motto “keep it simple”, the software supports no more than three commands, which in practice are sufficient to operate an audio power amplifier:

- on/off;
- volume +;
- volume –.

A commercial IR receiver is used here as its cost defies any attempt at home construction. In addition, these modules (**Figure 5**) decode the received IR signal and provide their output with a ready-to-use demodulated signal that can be fed directly to the microcontroller. Atr [2] you will find a lot of useful information on this subject.
Finally, a thermostat module can be connected to connector K7 (if desired) to switch off the amplifier when the temperature in the casing exceeds about 60 °C).

The software

The software is written in assembler for an ATTiny44, using ATMEL Studio 7. Thanks to the extensive hardware functionality of the ATTiny44 and the spacious memory, a compact solution was found for the softstart unit, especially when we take into account its possibilities. For the record: we don’t go into the code in detail here; if you like it, you can download the firmware and find out for yourself.
One of the most interesting features of the ATTiny is the so-called ‘Capture Function’ which we embraced for IR reception. With this function, the µC loads a copy of the counter position of the 16-bit timer to the Capture input pin in the so-called capture register, on every edge of the signal. In this way an extremely accurate (order of magnitude: microseconds) time measurement is possible.
Figure 6 shows the state diagram of the software. The microcontroller controls a total of four relays (K1–K4 in the image, RE1–RE4 in the diagram) for the switch-on process. In Figure 6 ‘button’ is the pushbutton for manual switch-off (see below). The process is sequential.
First, relay RE2 activates thermistor R6/R7. After a short delay, the circuit is closed by relay RE1 and after the delay ‘wait1’, relay RE2 is switched off again so that the thermistor is bypassed. The switching power supply for the amplifier is now directly connected to the mains but is currently unloaded.
Now the supply voltage is fed to the amplifier. First, relay RE4 connects both thermistors R20 and R21 into the positive and negative power lines of the amplifier. This is done to load the ‘fat’ reservoir electrolytic of the amplifier in a current-limited fashion. Doing so protects the components concerned and prevents the SMPSU from getting stuck in a short-circuit loop. After another short delay, the supply voltage is transferred ‘hard’ by relay RE3.



Figure 5: The author’s IR receiver module.

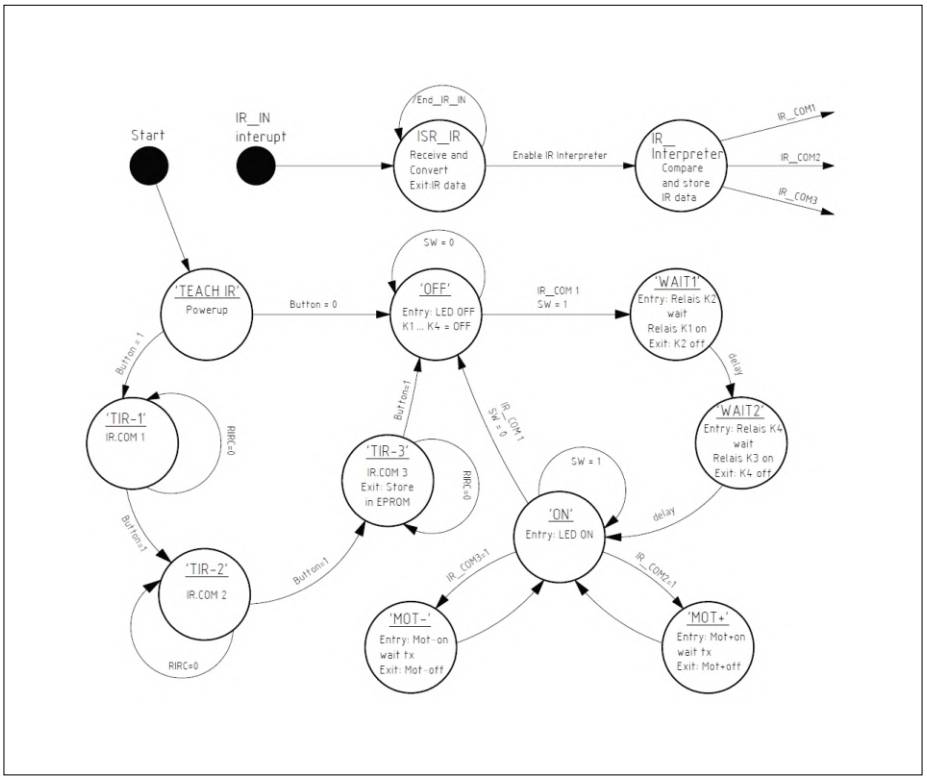


Figure 6: Firmware state diagram.

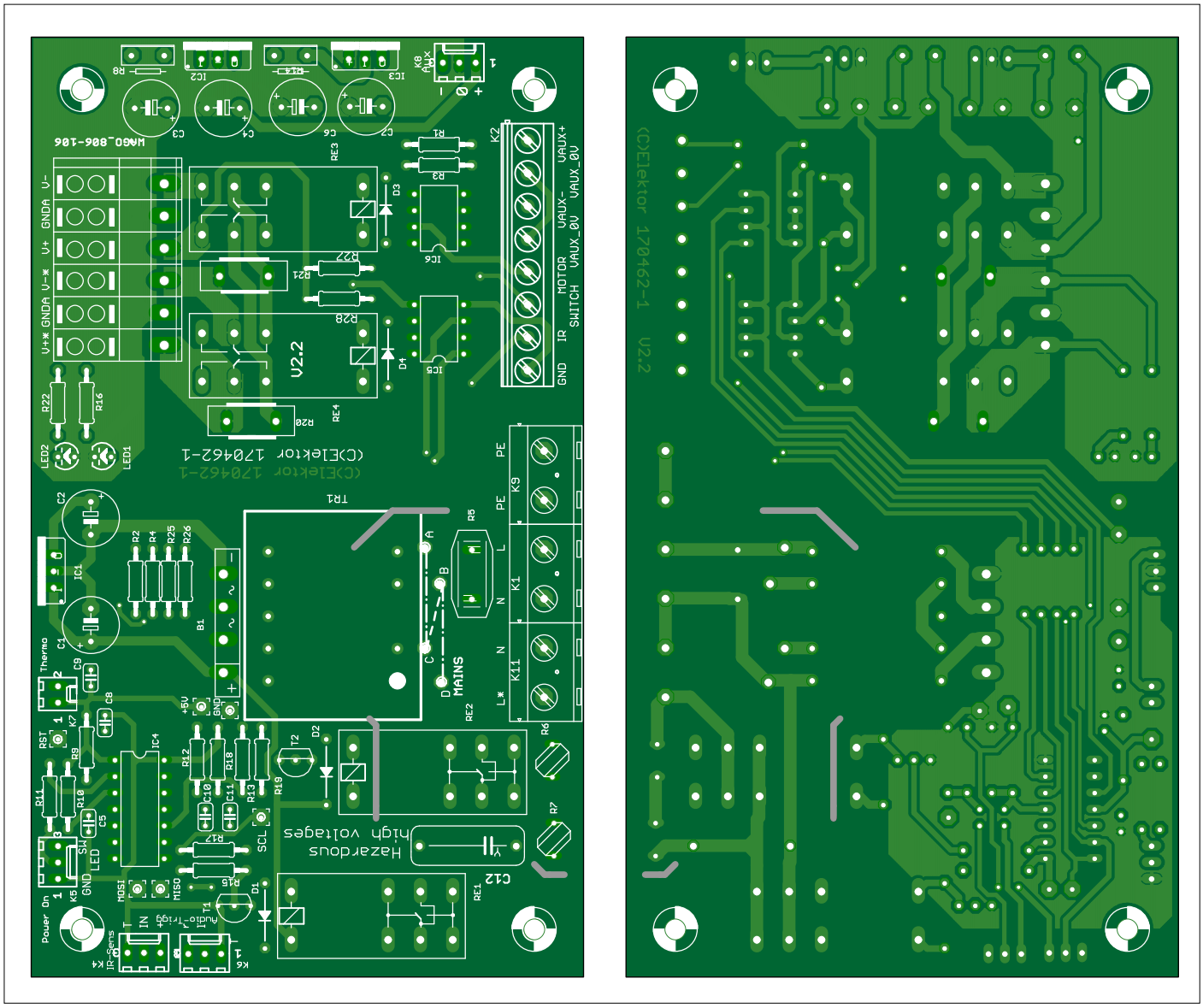


Figure 7: Printed circuit board designed for the softstart unit.

no more THUMP! at power-on

Teaching commands

The 'learning mode' for the IR remote control is activated by holding down the ID key while plugging in the softstart unit. The softstart unit remembers the IR signal and stores it in EEPROM. The LED indicates successful reception. To learn the second command, press the pushbutton again; the same for the third command.

In total three commands can be stored. The first command is the on/off command, the second the volume-up command ('louder') and the third, the volume-down command ('softer').

Final notes on the hardware

The hardware has an autostart function. As soon as an audio voltage is applied to the differential input of the microcontroller (pins 10 and 11), the amplifier switches itself on as described above. The switching threshold is approximately 20 mV. If no input signal is detected for a period of about 5 minutes, the amplifier switches itself off again. The standby consumption of the circuit is less than 1 W in this case. The audio signal proper is connected to connector K6.

The author uses a type SMPS300RE switching power supply from Connex Electronics. This provides an extra auxiliary voltage of ±12 V, which is used to control the motor potentiometer of the amplifier. For this purpose, the two bidirectional MOS relays in IC6 convey either the positive or the negative auxiliary voltage to the motor. The current is limited by resistors R1 and R2. Each volume command switches the motor on for 500 ms in the desired direction of rotation.

When the amplifier is turned on, it can be turned off by pressing the pushbutton.

Construction

A (double-sided) PCB was designed for the softstart unit — see **Figure 7**. And

COMPONENTS LIST

Resistors

R1,R3 = 33Ω carbon film, 5%, 0.25 W, 250V
R2,R4,R25,R26 = 680Ω carbon film, 5%, 0.25W, 250V
R5 = varistor, Epcos B72210S2271K101 (for 230VAC mains voltage) *
R5 = varistor, Epcos B72210S2151K101 (for 115VAC mains voltage) *
R6.R7 = NTC 16Ω, 2.1W, Epcos B57236S160M (for 230VAC mains voltage) *
R6 = NTC 16Ω, 2.1W, Epcos B57236S160M (for 115VAC mains voltage) *
R7 = wire bridge (for 115VAC mains voltage) *
R8,R14 = 25Ω, 1%, 15W, T)-126, Caddock 915
R9,R10,R15,R17 = 22kΩ carbon film, 5%, 025W, 250V
R11 = 470Ω carbon film, 5%, 0.25W, 250V
R12,R13,R18,R19 = 100kΩ carbon film, 5%, 0.25W, 250V
R16,R22 = 4.7kΩ carbon film, 5%, 0.25 W, 250V
R20,R21 = thermistor, ICL NTC, 10 Ω, -20% to +20%, radial, Ametherm SL15 10006
R27,R28 = 220Ω metal film, 1%, 0.6 W, 350V

Capacitors

C1,C3,C6 = 470µF, 50 V, 5mm pitch (13x21 mm)
C2,C4,C7 = 47µF,100V, 5mm pitch
C5,C9,C10,C11 = 100nF, 50V, X7R, 0.2" pitch
C8 = 1nF, 100V, X7R, 0.2" pitch
C12 = 4.7nF, X1/Y1, 500V, Y5U, 15mm pitch

Semiconductors

D1–D4 = 1N4148
B1 = bridge rectifier 600V, 4A, GBU405
LED1,LED2 = green, 3mm
T1,T2 = BC337
IC1 = 7805 (+5V, 1A)
IC2 = 7818 (+18V, 1A)
IC3 = 7918 (–18 V, 1A)
IC4 = ATTiny44-20PU microcontroller, programmed, Elektor Store no. 170462-41 *.
IC5,IC6 = TLP222A-2 double optocoupler

Miscellaneous

K1,K9,K11 = 2-way PCB screw terminal block, 0.3" pitch, 630V
K2 = 8- way PCB screw terminal block, 0.2" pitch(300V)
K3 = 6-way 2-wire PCB terminal, Wago 806-106 *
K3 = matching 6-pin header, Wago 806-906 *
K3 = 6-way PCB screw terminal block, 0.2" pitch, 300V, e.g. Multicomp MC000039 *
K4,K5,K6,K8 = 3-pin pinheader, 0.1" pitch w. matching cable parts
K7 = 2-pin pinheader, 0.1" pitch w. matching cable part
Shrink connector 1-pole, minimum 14 pieces
TR1 = safety transformer, 1.5VA, primary 2 x 115V, secondary 2 x 6V, e.g. Block AVB1.5/2/6
RE1,RE2 = power relay, SPDT, 12VDC, 16A, through-hole (Omron G2R-1-E 12DC)
RE3,RE4 = power relay, DPDT, 24VDC, 8A, non-locking (Omron G2RL-24-CF-DC24)
GRV IR receiver (Seeed)
60°C thermostat, e.g. Thermorex TK24-T02-MG01-Ö60-S50
PCB no. 170462-1 (v2.21) from Elektor Store

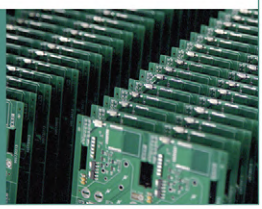
Advertisement

PCBWay

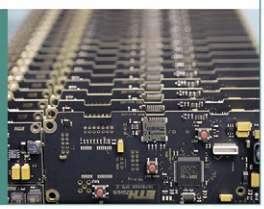


As a manufacturer based in Shenzhen, China, PCBWay helps individuals and companies overcome electronic technology challenges through their professional services in PCB prototyping and assembly. Whether its small-volume or large production, the company delivers its expert PCB services, all under one roof.

CUSTOM PCB SERVICE
24-hour production time
ONLY $5
FOR 10 PIECES



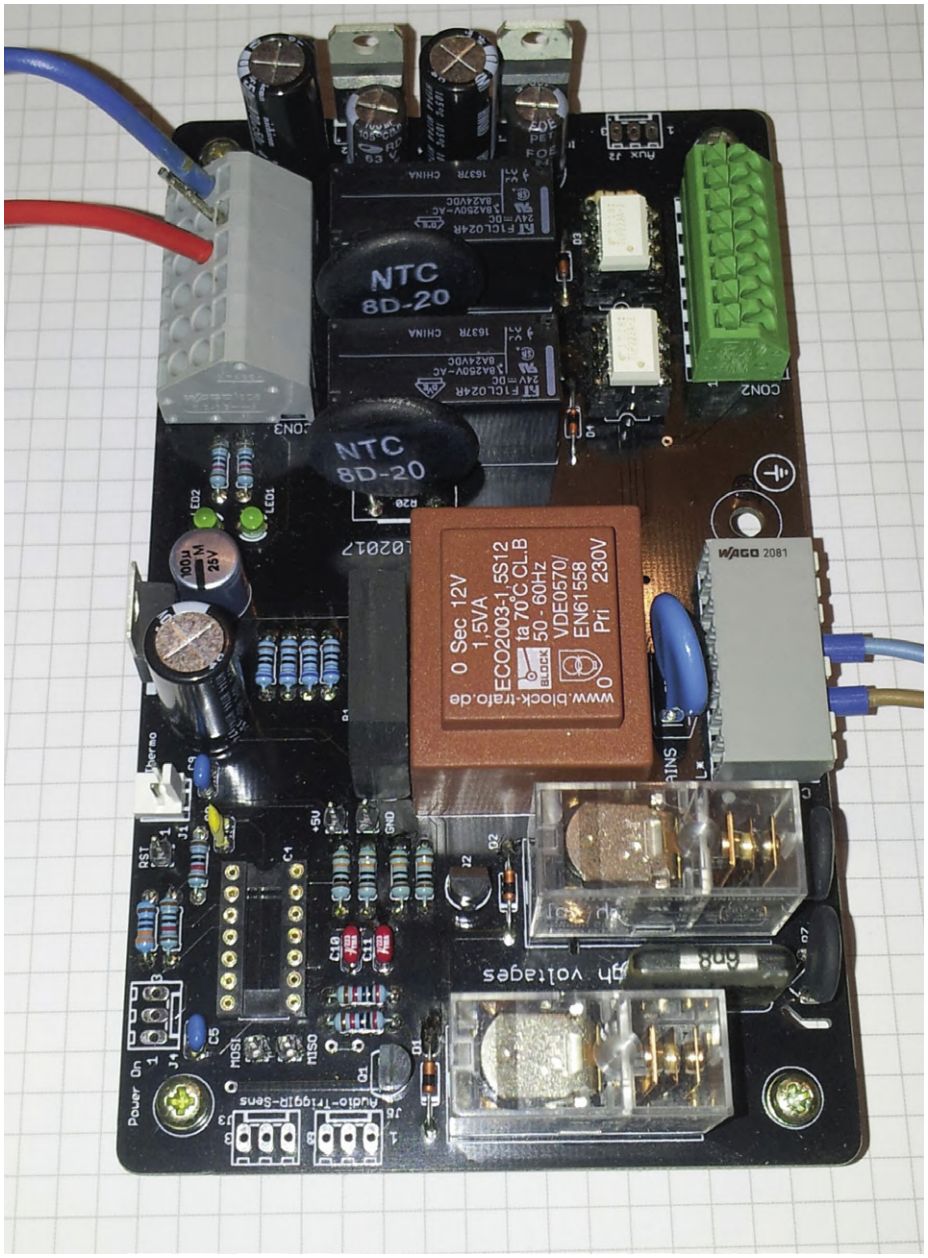
SMT & THT ASSEMBLY
Free worldwide shipping + Free tooling fee
START
FROM $30



What really differentiates PCBWay is its production efficiency. While the other competitors take 7-10 days to deliver, the company can complete the PCB orders in 24 hours. Further, PCBWay also offers 24-hour customer support allowing the clients to receive assistance whatever time of day or night.

As one of the most experienced PCB manufacturers and SMT Assemblers in China, PCBWay prides itself as the best business partner in every aspect of their clients' PCB needs.

Get an instant quote : www.pcbway.com Email : service@pcbway.com



@ WWW.ELEKTOR.COM

→ **Unpopulated printed circuit board**
www.elektor.coml/170462-1

→ **Programmed microcontroller**
www.elektor.com/170462-41

→ **USB ISP-Programmer Stick for AVR**
www.elektor.com/18438

you will no doubt be pleased that this is a real 'old-fashioned' PCB with through-hole components only.
We have somewhat 'internationalised' the author's original design — i.e. made it suitable for both 230 VAC and 115 VAC mains (US: 'line') voltage. This means that you must be careful when selecting and assembling some parts.

- First, the transformer. That's a type with two primary windings. For 230 VAC, points B and C must be connected on the PCB (use insulated wire!) while points A and D remain open. For 115 VAC use, point A must be connected to point C and point B to point D (again: use insulated wire!).
- Next, varistor R5. Its value follows the AC line voltage: 230 V or 115 V. Take a good look at the parts list here!
- Then, thermistors R6 and R7. These have the same value (see parts list), but for 230 VAC both thermistors must be mounted, while for 115 VAC only R6 is mounted and R7 is replaced by an (insulated!) wire jumper.

Finally, connector K3 deserves some attention. The author has used a Wago PCB, which is adequate but may be difficult to obtain outside of Germany. Instead, you can also use a 6-way, 0.2 inch pitch PCB screw terminal block, e.g. the Multicomp type from the Components List. Note: if you choose the Wago PCB terminal, you should also order the corresponding Wago 6-pin header!

The PCB is of course available in the Elektor Store and the same goes for the programmed ATTiny44 (see the box: **@ www.elektor.com**). However, if you want to program your own controller, feel free to do so. The firmware can be downloaded from the project page with this article [3] (free of charge, as you would expect from us). You can then program the ATTiny44 with an AVRISP, which is connected to the corresponding PCB pins on the PCB (5V, MOSI, MISO, GND, RST and SCL).
Last but not least: parts of the PCB carry life-threatening voltages; therefore, build in the whole project in a touch-safe manner! All that remains for us to do is to wish you lots fun building the softstart and hope you can enjoy long and undisturbed listening.

(170462-03)

Web Links

[1] Switching power supplies for audio amplifiers: www.audiophonics.fr

[2] Infrared receivers (and other interesting items): www.electronics-base.com

[3] Project page for this article: www.elektormagazine.com/170462-03

Baluns

Peculiar Parts, the series

By **Neil Gruending** (Canada)

In 1939 the Empire State Building used a 'balun' in a television antenna feedline for the first time. It's almost 100 years later and baluns are more important now to modern life then ever. But we need to talk about balance before taking a closer look at baluns.

A balanced transmission line system is important because it greatly reduces unwanted radiation and received interference. The trick to keeping everything balanced is to make sure that the conductors are the same length and impedance so that the current flowing one conductor has an out of phase current flowing back in the other conductor. When the currents are exactly 180 degrees out of phase, all of the electric fields will cancel out, eliminating any radiated signal. For example, ladder line does this by keeping the conductors a constant distance apart and dipole antennas use equal length conductors in opposite directions.

But the problem is when you will eventually need to connect an unbalanced, single-ended signal to the balanced transmission line such as using coax cable feedline for a dipole antenna. In this case the dipole element connected to the coax centre conductor will radiate all of the current, but the element connected to the coax shield will only radiate a fraction of the opposite current. The rest of the current flows down the outside of the coax shield causing unwanted radiation due to the impedance mismatch [2]. A balun (BALanced UNbalanced) can be used here to correct the mismatch and direct the coax shield current out the dipole element.

The technical definition of a balun is any 3-port device with a ground-referenced single-ended input and a differential or balanced output like in **Figure 1**. By design, the unbalanced input is matched to the input line impedance and the outputs are equal and 180 degrees out of phase with a balanced impedance. The ratio of the unbalanced and balanced impedance is the impedance ratio which is usually 1:1 or 1:4.

Baluns can either be magnetically coupled, capacitively coupled or a combination of both like in **Figure 2**, but magnetic coupling is the most popular. It's made by winding two separate windings onto a magnetic core to and the ratio between the primary and secondary windings controls the balun impedance ratio. Magnetic coupling is generally used to up about 1 GHz where capacitive coupling becomes more effective. The actual structure of the balun also depends on whether it's a balanced voltage or current output balun, and the frequencies used.

Baluns have a lot of uses in RF circuits but you will also find them in used in long distance high speed signalling like Ethernet, USB and HDMI. They can even be used to interface high speed differential ADCs to single-ended signals while also adding voltage gain at a lower noise level then a traditional differential amplifier. But the best thing about baluns is that you can wind them yourself to try one out for your next project. You can even use them backwards to convert balanced to unbalanced signals!

180575-F-01

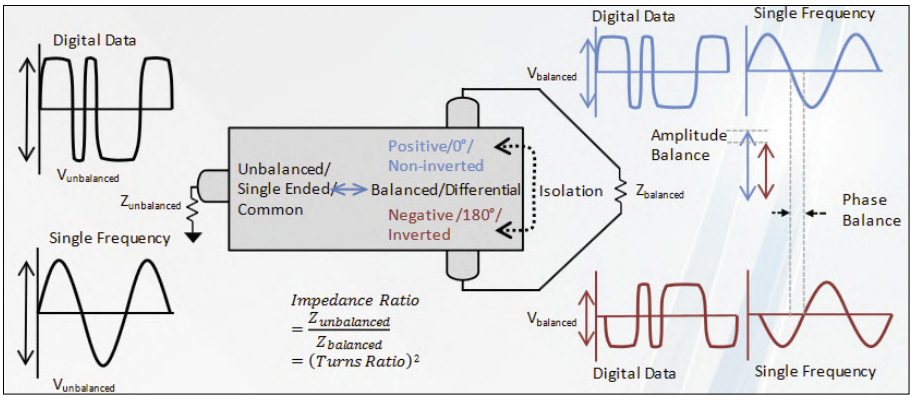


Figure 1: Balun operation [1].

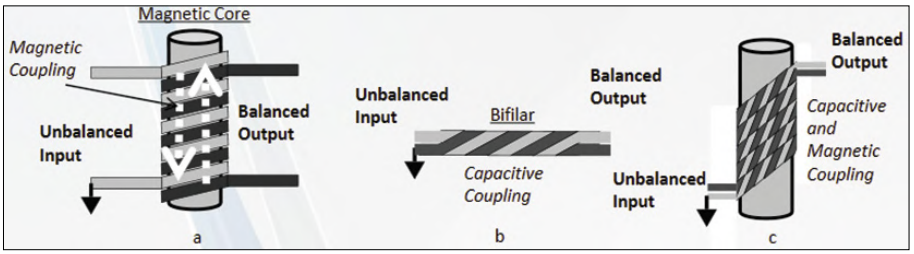


Figure 2: Balun construction [1].

Web Links

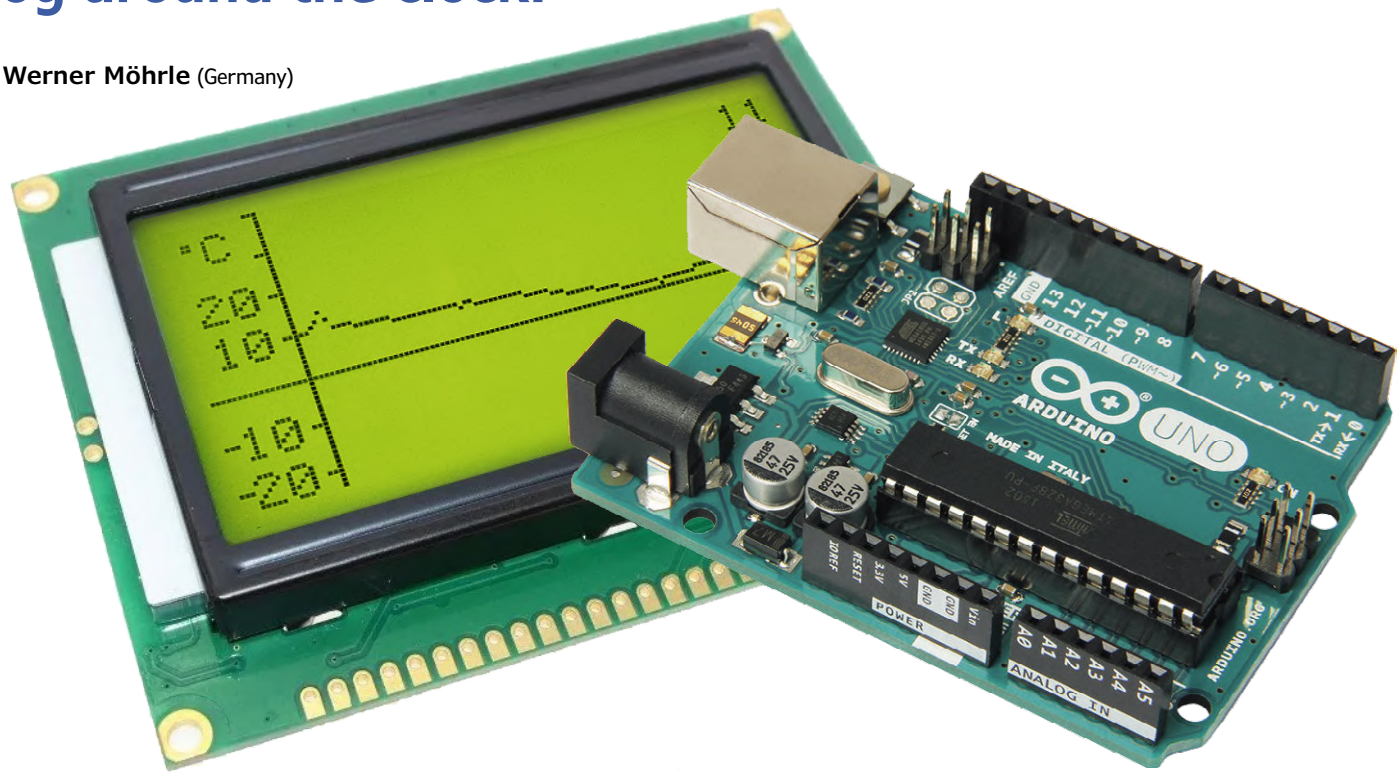
[1] Balun operation: www.markimicrowave.com/Assets/appnotes/balun_basics_primer.pdf

[2] Balun construction: www.eznec.com/Amateur/Articles/Baluns.pdf

Arduino Temperature Recorder

Log around the clock!

By **Werner Möhrle** (Germany)



Using the Arduino Uno, you can build all manner of really neat small projects at modest cost. This one is a simple temperature recorder that shows on a display screen the values measured during the past 24 hours. Of particular note is the software, which manages all of the operations without recourse to external libraries.

The recorder described in this article measures the temperature over the course of a day and displays the readings on a graphic display screen. When a new measured value is added, the displayed curve is shifted one pixel to the left, so that the oldest value disappears and the newest reading appears. The interval between measurements can be changed easily to record more rapid temperature changes, for example in process control applications.
The circuit in **Figure 1** shows the (just) three active components (and a few passive parts) of this miniaturised temperature recorder.

The recorder described in this article measures the temperature over the course of a day and displays the readings on a graphic display screen. When a new measured value is added, the displayed curve is shifted one pixel to the left, so that the oldest value disappears and the newest reading appears. The interval between measurements can be changed easily to record more rapid temperature changes, for example in process control applications.

Programmable temperature sensor

The DS1631 temperature sensor takes temperature readings between -55 °C and +125 °C, in the range 0 to 70 °C with an accuracy of ±0.5 °C. The resulting measurements are stored in a register with a resolution of 9 to 12 bits (according to the user's needs) and can be read out from there. Thanks to its supply voltage range between 3.0 and 5.5 V, the sensor is suitable for interfacing to a broad variety of microcontrollers. Another remarkable feature in these times is that the DS1631 is available not only in SMD format but also as a DIP package. The DS1631, whose address is programmable using pins A0 to A2, communicates with the controller via an I^2C bus. The internal circuitry of the sensor in **Figure 2** shows configuration registers that are written to by the controller and a read-only temperature register from which the controller reads the temperature detected. The sensor is controlled by the Arduino temperature recorder software, in continuous mode and with a resolution of 12 bits.
With the two registers TH and TL you can also specify a hysteresis range, enabling the DS1631 to be used as a thermo-

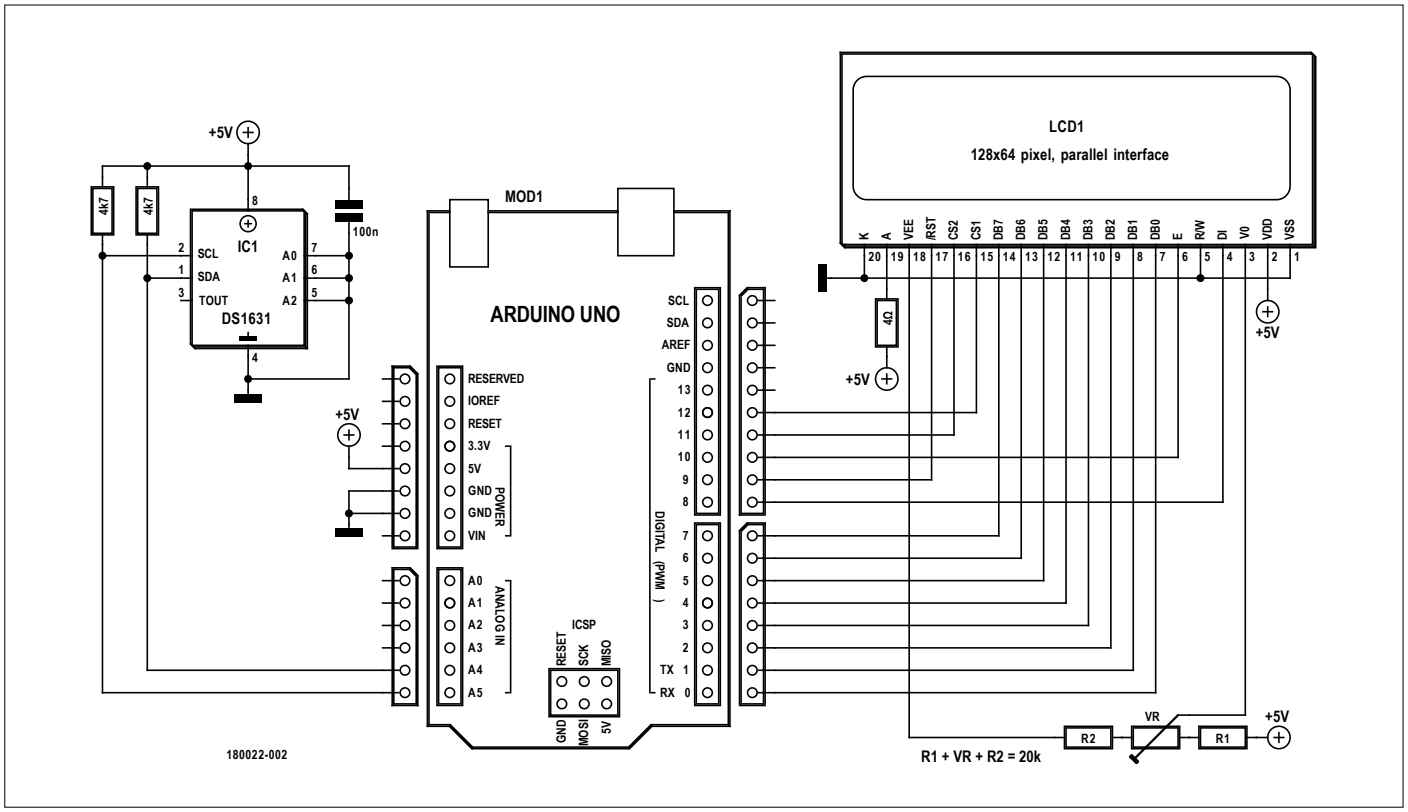


Figure 1. The three-component circuitry of the temperature recorder.

stat (with switching output TOUT). However, this feature is not used in the current project.

A suitable display

These days it is common practice to employ graphic displays equipped with a serial bus (I^2C or SPI). However, given that the temperature sensor used here is already provided with an I2C bus and occupies only three I/O pins of the Arduino, there are enough pins available for parallel control of the display. This does not offer us any notable advantages or disadvantages and you would save only a few pence.

Suitable displays with 128x64 pixels are available from plenty of manufacturers (Vishay, Lumex, Winstar and so on), which sell for less than €10 / £9 / $11 on eBay and other websites. Size and screen colour choice are entirely up to you; your only major task is to make sure that the pin assignment (see **text inset**) matches, as there are several connection variants.

The graphic display (**Figure 3**) shows a series of 108 temperature readings in a coordinate system. If these 108 values correspond to 24 hours, the temporal resolution of the display amounts to 800 s, barely a quarter of an hour. The current temperature is shown in the upper right-hand corner of the display.

Data central

The three components of the Arduino Uno microcontroller initiate the temperature measurement process, capture the temperature values determined and transform them into pixels positioned correctly on the display.

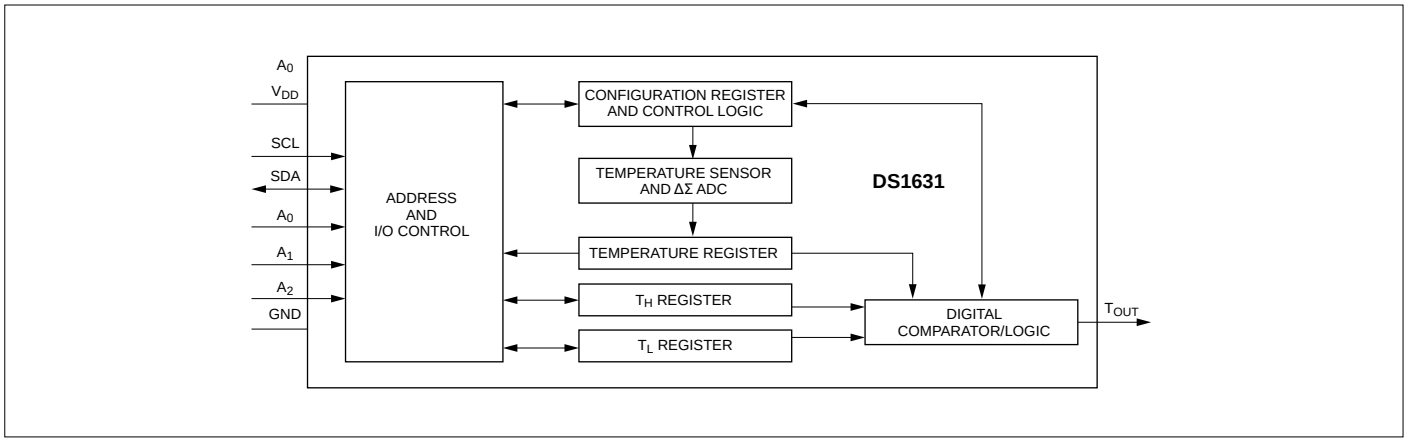


Figure 2. Internal functions of the temperature sensor (source: Maxim Integrated).

Display pinout		
Pin	**Symbol**	**Function**
1	VSS	Supply voltage 0 V
2	VDD	Supply voltage +5 V
3	V0	Operating voltage LCD driver
4	D/I	H: Data input, L: command code input
5	R/W	H: Data read, L: data write
6	E	Enable (negative edge)
7 - 14	DB0 - DB7	Data bus
15	CS1	Chip select for IC1 (segment drivers 1 to 64)
16	CS2	Chip select for IC1 (segment drivers 65 to 128)
17	RES	Reset (active L)
18	VEE	Output negative voltage
19	A	LED backlight anode
20	K	LED backlight cathode

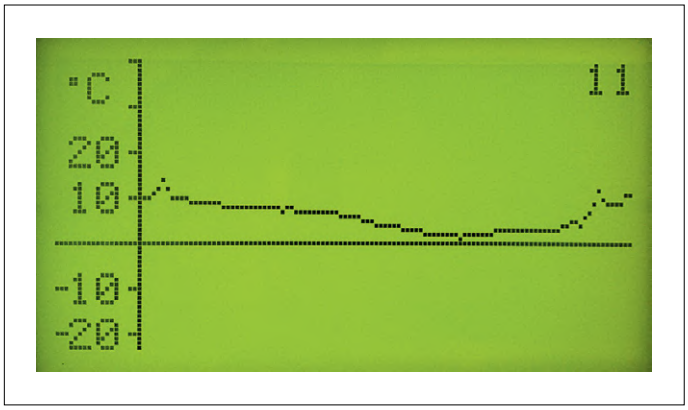


Figure 3. Temperature display showing the 108 readings logged during the preceding 24 hours.

A Sketch with no Libraries

The first thing you will notice in the fully-commented Sketch [1] is that it does not rely on external libraries at all. All read and write operations involving the sensor and the display are performed using features already provided by the Arduino IDE. The most important functions can be found at the end of the Sketch. The initialisation of the controller, temperature sensor and display is carried out in the `setup` function. In the main routine `loop`, the data of the temperature sensor are read out (`read_temp`) after being called by the clock generator (`meas_flag==1`) and the temperature value measured is calculated from this (`cal_temp`). The variable is set to zero if the displayable limit values of -23 °C and +40 °C are exceeded or not reached. From this the value of the ordinate axis (`cal_measval`) is calculated. The corresponding data for loading the display are determined with the new measured value (`set_meas(measval)`) and finally the display is updated using this. In preparation for the output of the next measurement, the pixel data of the temperature values are then shifted one pixel to the left, so that the oldest measured value disappears.

Timing in this Sketch is provided by the interrupt service routine, to be found at the very end of the Sketch. It is here that the interval between measurements is set at 800 s, so that the total of 108 displayable measurement intervals in 108×800 s = 24 h are processed just once. To test the Sketch, you can shorten the interval to, say, 2 s.

Control of the sensor is also handled towards the end of the Sketch. The 2-wire hardware interface of the controller (A4 for SDA and A5 for SCL) is used. The function `write_to_config` describes the configuration register. The configuration register is accessed using `TWDR=0xAC` and `TWDR=0x0C` selects the continuous 12-bit mode. After this, the function `start_convert` initiates the measurement mode (`TWDR=0x51`).

Function `read_temp` ensures that the two bytes of the register holding the current temperature value are read sequentially (`TWCR=0x84`) and stored in the variables `temph and templ`. The `cal_temp` function then converts the two binary values into the signed float-variable `temp`.

The 64 horizontal columns of the 128x64 dot matrix of the display are divided on a temperature scale of −23 to +40 °C, the 128 vertical rows in 19 dots for the representation of the temperature scale and in 109 dots for the temperature curve. The display itself is controlled by a segment driver for the left half (CS1) and one for the right half (CS2), divided into eight superimposed pages with 64x8 bits each.

A quick overview of operations

Array `num[11][5]` determines how the numeric value of the current temperature shown in the upper right-hand corner of the display is displayed, i.e. which pixels should be black. The array `dmask[8][128]` stores the entire display content, so as to be able to re-display the previous readings shifted one point to the left after each new measurement.

The following functions are used to control the display. Functions `set_ytics` and `set_ylabels` draw the axes labelled with the temperature scale. Function `shift_left` shifts the previous temperature curve one pixel to the left. Function `set_meas` determines which data and at which point the newly measured value is loaded into `dmask`. Function `load_gdisp` loads the display with the data stored in this array. Finally `load_num` enters the last measured value as a number in the upper right-hand corner, if there are no measured values in this area (otherwise it is moved further down).

The software, with comments, can be downloaded in the usual manner from the Elektor project page [1]. ⏮

180022-02

Web Link

[1] Elektor project website: www.elektormagazine.com/180022-02

@ WWW.ELEKTOR.COM

→ Arduino Nano
www.elektor.com/arduino-nano-3

→ Book: Make Sensors
www.elektor.com/make-sensors

Sound Blink

Prototyping a portable DIY speaker with LED light show

By **Adhith M.** (India)



Sound Blink is an effort to create a portable speaker that's a cut above most party sound boxes. It comes with lots of functions which make it unique, also as a crafted project with many parts made by modifying easily available materials. The same for the electronics, basically, being a combination of off the shelf and home made.

This 50-watt portable speaker — that fits in a gym bag — uses multiple speaker drivers within a wooden enclosure to deliver superior sound quality. The customisable curved LED matrix on the front of Sound Blink provides both a light show and an audio equalizer. Other useful functions like wide voltage charging, onboard music controls, charging level indicators, AUX input & output make Sound Blink more user friendly.

Audio drivers and crossovers

The multiple speaker driver setup includes a pair of passive radiators, woofers, and tweeters to enhance the low, mid and high audio frequencies **(Figure 1)**. The woofers and tweeters at the rear of the unit are connected through a pair of 2-way audio crossover filters, which split the audio signals as received from the amplifier. The passive radiators on the top and bottom of the enclosure vibrate to the pressure difference caused by the woofer cone movement. This enhances the bass response to a great extent. The combined effect of all drivers along with the wooden enclosure were the core aspects for achieving clear & distinct sound quality.

Design of the wooden enclosure

Planning the entire process and choosing the right materials was way more time consuming than the actual build. The approach of using simple techniques like jigsawing makes replication convenient for those without access to 3D printers and CNC routers. However, this approach did challenge my abilities to find unconventional solutions.

The build commenced by making the circular enclosure from plywood. I preferred a curved LED lighting setup considering it would offer a larger viewing angle and better aesthetics (a.k.a. 'looks'). However, due to this particular shape I followed the layered enclosure technique. The plywood cutouts were stacked to form a single pile structure. Next, each layer was fixed using wood glue and brad nails **(Figure 2)**.

Moving on to the LED matrix, this consists of horizontally and vertically mounted 12-V RGB LED strips. Using its self-adhesive back, the assembly was fixed on a piece of flexible OHP sheet. It was later fixed around the outer enclosure using screws **(Figure 3)**. The strips were wired in a specific arrangement



Figure 1: The woofers and tweeters at the rear side.

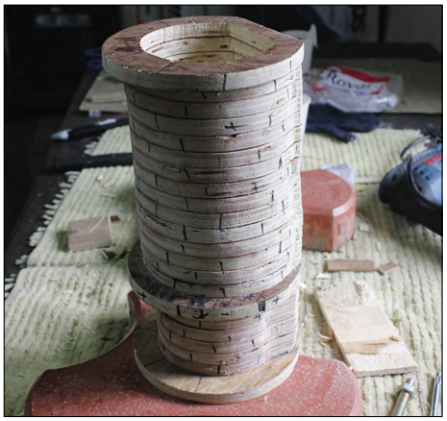


Figure 2: Wooden enclosure made from stacking plywood cutouts vertically.



Figure 3: The curved LED matrix mounted at the front.

for the desired lighting effects and colour combinations. The surrounding acrylic casing concealed the setup while allowing light to pass through. It's made by first heating a piece of white acrylic sheet using a heat gun and then rolling around a length of PVC tubing. The result is also covered with a black sun protector film to make it resemble a curved LCD screen **(Figure 4)**. Later, this casing was also secured to the enclosure with screws.

Dual-function LED indicator

The LED music equalizer readout built into Sound Blink is one of the project's key features. A slightly modified VU meter (volume unit meter) circuit allows this to be done **(Figure 5)**. It's based on the well-known LM3915 IC and a few external components. Simply put, based on the audio signal intensity at the input this IC sequentially drives a series of 10 internal comparators. These comparators then drive 10 rows of the LED matrix to form the equalizer readout. This particular circuit designed specially for Sound Blink is a modified version of the default application found in the LM3915 datasheet. It also has an internal peak detector circuit (based on the LM358) for better sensitivity and overall a more accurate indication. By using Darlington transistors type MPSA64 at the output side rather than the regular BJTs, the current load on the LM3915 outputs was greatly reduced. This reduces unwanted heat dissipation and prevents any overheating issues with the IC.

Modular!

A total of eight different electronic modules was used in this project, the lineup is pictured in **Figure 6**. Excepting the battery pack and the VU meter they are ready-made products from online suppliers:

- 50-W class D amplifier board with Bluetooth 4.0 [1]
- Buck-boost module for voltage regulation (2 pcs.) [2]
- 4S1P-style 5.8 Ah lithium-ion battery pack
- BMS (battery monitoring system) [3]
- Constant current-constant voltage (CC/CV) buck converter [4]
- Wi-Fi LED driver [5]
- VU meter circuit

All modules were secured in the lower half of the enclosure using screws, and concealed by another acrylic casing.

Charging and powering

Another noticeable feature of Sound Blink is its wide charging capability from reasonably powerful DC power adapters supplying any voltage from 5 V to 30 V. This covers most 'power bricks' including phone and laptop chargers. The buck-boost converter with 18 V out, followed by the CC/CV buck converter with 16.8 V, 800 mA out, makes this possible. A battery monitoring system (BMS) connected to the battery pack provides essential functions like balancing of the Li-ion cells, over/under-

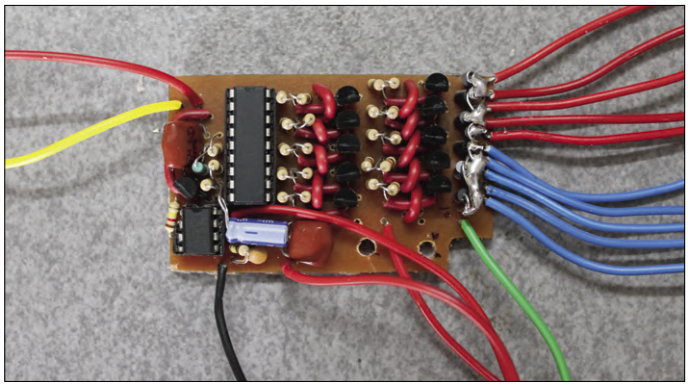


Figure 5: The VU meter circuit board.

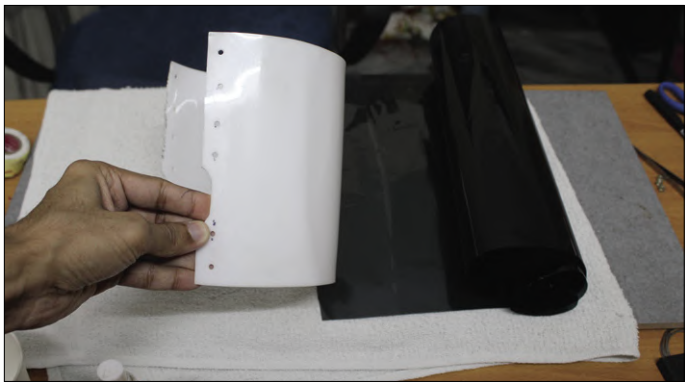


Figure 4: The white acrylic casing and the sun protector film.

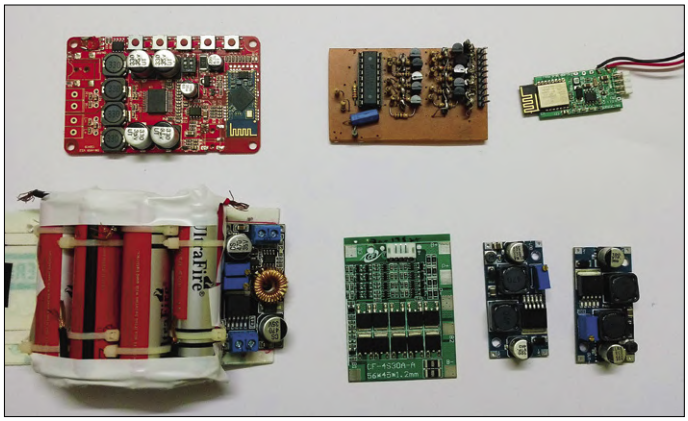


Figure 6: The eight electronic modules used.



Figure 7: The switch controls are in a box mounted on the rear of Sound Blink.

charge protection, and short circuit protection. This makes the battery pack efficient and safe to use.

On Wi-Fi

In good party fashion the Wi-Fi LED driver offers a lot of functions like a 16-million colour wheel for lighting, default and custom lighting effects, timers, colour selection through camera, and much more. All these functions are controllable through a Wi-Fi network and a smartphone app. Both the driver and the VU meter are powered at 12 V by a buck-boost converter.

Controls and user interface

With so many functions and options on offer on Sound Blink, the number of switches and controls is large. A switch box containing three DPDT and ten SPST switches, and a 47-k potentiometer is arranged at the rear bottom side of the speaker **(Figure 7)**. It includes a dedicated switch to toggle power straight from the DC adapter (line-in mode) or from the battery pack. The advantage: close to a power outlet, Sound Blink can be used in line-in mode without exhausting the battery pack. Pressing the dedicated charging switch disconnects the load and simultaneously initiates the charging process of the battery pack. So while using the speaker during charging, a part of the input power goes to charging and the rest is used to drive the speaker. This eliminates the strain from simultaneous charging and discharging of battery cells as seen in a lot of portable devices.

The LED driver when used alone may act as a mood lamp. By coupling it with the VU meter it forms the equalizer. A separate switch toggles between these two functions. The VU meter also offers two different lighting patterns, dot and bar mode. Dot mode drives a single row of LEDs at a time whereas bar mode drives the rows of LEDs sequentially.
A 47-k potentiometer associated with the VU meter is used to adjust the row of sensitivity of the LED matrix.
Since the LED matrix consist of both horizontal and vertical strips, lighting them in different colour combinations allows further colour mixing within the matrix. This is achieved by individually cutting off one or two lines of the R, G and B terminals using 6 separate switches.

Conclusion

Working on this project and going through each stage of the build was a rewarding experience. Since it required various processes like woodworking, electronics, and crafting, I was able to expand my knowledge and skills in these areas.
Those of you interested in exploring details of the project are advised to check out the complete article posted on the ElektorLabs website [1]. Also worth viewing is the video at [6] showcasing the Sound Blink prototype in action. ⏮

190101-01

Web Links

[1] Amplifier Board (with onboard Bluetooth CSR 4.1 audio RX): https://bit.ly/2YWZPbY
[2] Buck/Boost module (two required): https://bit.ly/30yyayt
[3] CC—CV Buck Converter: https://bit.ly/2Jb2eIZ
[4] BMS Module: https://bit.ly/32oMICJ
[5] Wi-Fi RGB LED Controller: https://bit.ly/2LUBbFq
[6] Sound Blink assembly video (author's production) : https://youtu.be/wMS58EtNJO0

eLABS PROJECT

TMC2160 Motor Driver Board

silence and power in one

By **Ilse Joostens** (Belgium)

Stepper motors are popular in computer and microcontroller-based position control applications like robotics, industrial automation, CNC machines, 3D printers and video camera control. Due to their uncomplicated internal construction and their limited number of connection wires, stepper motors look deceptively simple. On the internet, there are even lots of tutorials on how to control these motors using an antique L293D driver IC and an Arduino, but for serious applications everything depends on the quality and the capabilities of the motor driver.

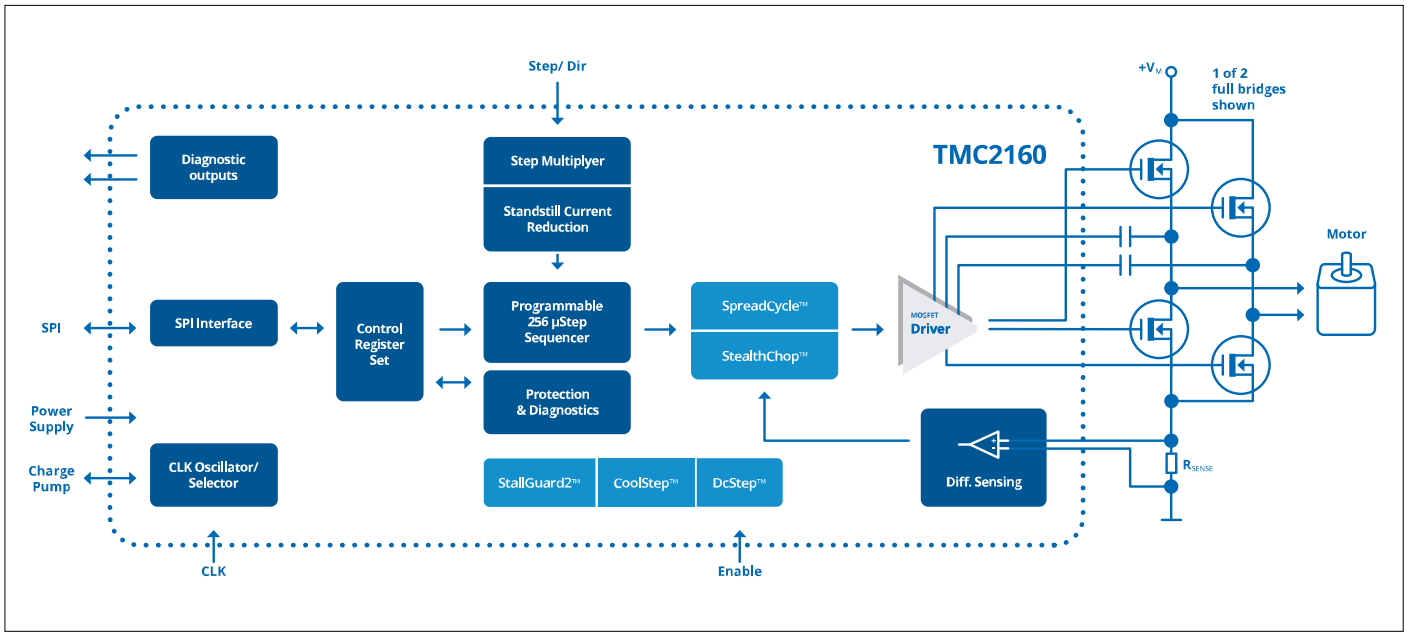


Figure 1. Inside the TMC2160 quite a lot is going on.

Quick Features

- 9 to 36 VDC supply
- Step/Dir interface via optocoupler inputs
- Motor current: 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.1 A RMS selectable
- Microstepping: 1/8 – 1/16 – 1/32 – 1/64 selectable
- SpreadCycle operation without hold current reduction or StealthChop operation with 50%
- Hold current reduction selectable
- Drives NEMA 17, NEMA 23 and some NEMA 34 motors (not at full power)
- Power, motor and control signal connections through pluggable terminal blocks
- Max. step input frequency: 250 kHz (depending on microstepping settings)

Commonly used stepper motor driver types

Stepstick drivers

If you are into hobby projects involving stepper motors or if you are building 3D printers, you are probably familiar with stepstick drivers. They consist of a small PCB with a stepper motor driver IC and pinheader connectors allowing the module to be plugged into a larger board. Stepsticks come in many flavours allowing you to choose from a range of driver

ICs from several semiconductor manufacturers. Some of the better stepsticks are the so-called SilentStepSticks built around Trinamic stepper driver ICs of which the TMC2100 and TMC2130 are the most popular ones.

These SilentStepSticks are a good choice for 3D printing applications [1], especially during lenghty print jobs. When using less advanced drivers, stepper motor noise may become objectionable soon. SilentStepSticks not only allow the printer to run silently but also help to achieve better, smoother and more consistent print results.

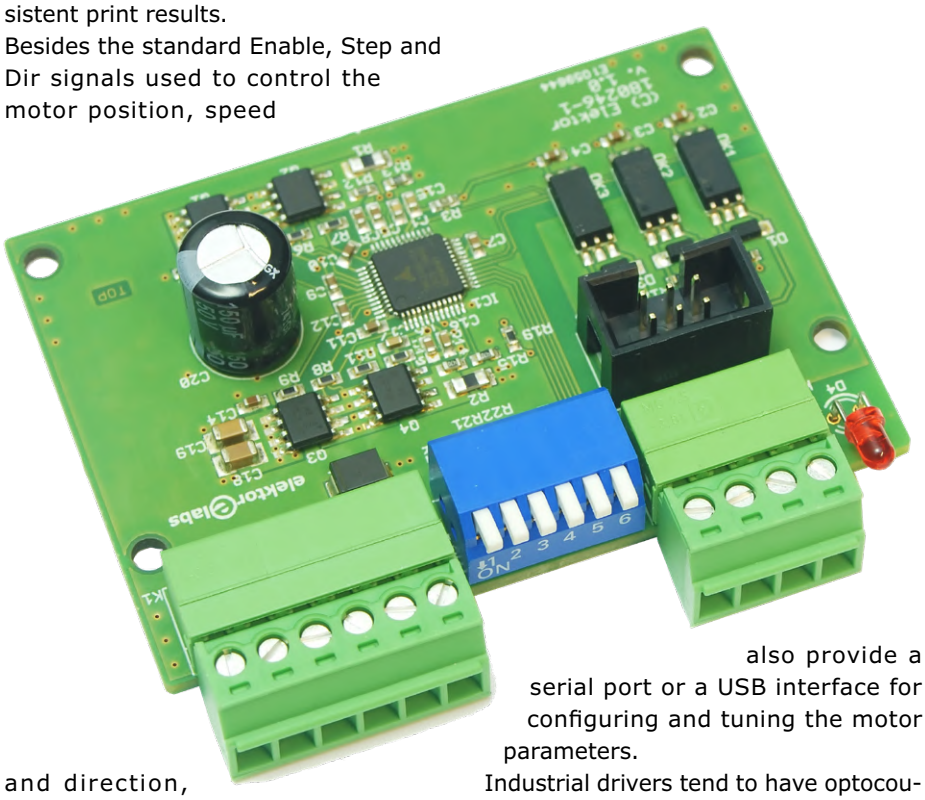
Besides the standard Enable, Step and Dir signals used to control the motor position, speed and direction, the TMC2130 and TMC5160 based stepsticks also have an SPI interface that allows tuning and configuring several driver parameters. Modern 3D printing software like Marlin [2] supports these drivers and can even autotune the motor drivers via the SPI interface during printing for best results.

While stepsticks are small and easily swappable, they also have a number of disadvantages. Due to their very small size, most use driver ICs with integrated MOSFET bridges that tend to overheat on small boards with almost no copper planes, even with a heatsink mounted on top. This is mainly a problem in more demanding applications like CNC.

Furthermore, stepsticks don't offer electrical isolation between the motor power supply and the control signals, making them less suitable for complex motion control systems.

Industrial stepper motor drivers

Industrial drivers are commonly used for a broad range of applications and come mostly in an enclosure with terminal blocks for connecting power, the motor leads and control signals. Often, the enclosure doubles as a heatsink for the electronics inside. Some driver enclosures are even suited for direct mounting on the back of a stepper motor. The configuration of the main driver parameters like microstepping and motor current is usually done using DIP switches. More advanced drivers also provide a serial port or a USB interface for configuring and tuning the motor parameters.



Industrial drivers tend to have optocoupler inputs, providing electrical isolation for the control signals (Enable, Step, and Dir signals). Some drivers use ModBus (or even CAN bus) instead to control motor position, speed and direction.

Broadly speaking two driver types are available: open loop and closed loop. The latter use rotary encoders for motor shaft position feedback to the driver. This results in higher accuracy and compensation if the motor skips steps under certain load conditions.

If you need to drive multiple stepper motors, multiple-axis motor drivers are also an option.

The Trinamic TMC2160 motor driver IC

Our 'Wolverine' CNC gantry robot [6][7] is based on a CNC310 controller board from Eding CNC. As the CNC310 board needs external stepper motor drivers, we considered using drivers based on Trinamic ICs. These would allow the CNC gantry robot to run silently, which is especially important when the machine is used as a pick & place machine for SMT components.



Initially we ran some quick tests with TMC2100 and TMC2130 SilentStepSticks but these were not up for the task and overheated quickly. Instead we looked at the brand new TMC2160 which uses external MOSFETs and was designed for more power-hungry applications (**Figure 1**). A TMC2160 breakout board was available but unfortunately configuring the motor driver parameters was possible via SPI only, something the CNC310 board doesn't support. Also, galvanic isolation between the control signals and the motor power supply was non-existent.

In the end we designed our own TMC2160-based driver with the IC running in standalone mode, and with optocoupler inputs for the control signals. This allows the motor driver parameters to be set using DIP switches.

Most modern stepper motor drivers like the TMC2160 are so-called chopper drivers, which rapidly switch on and off a relatively high voltage (typical eight

PROJECT DECODER

- Motor, PWM, Trinamic, CNC, 3D Printing, Robotics
- entry level / ➡ intermediate level / expert level
- 2 hours approx.
- SMD Soldering
- €75 / £75 / $80 approx.

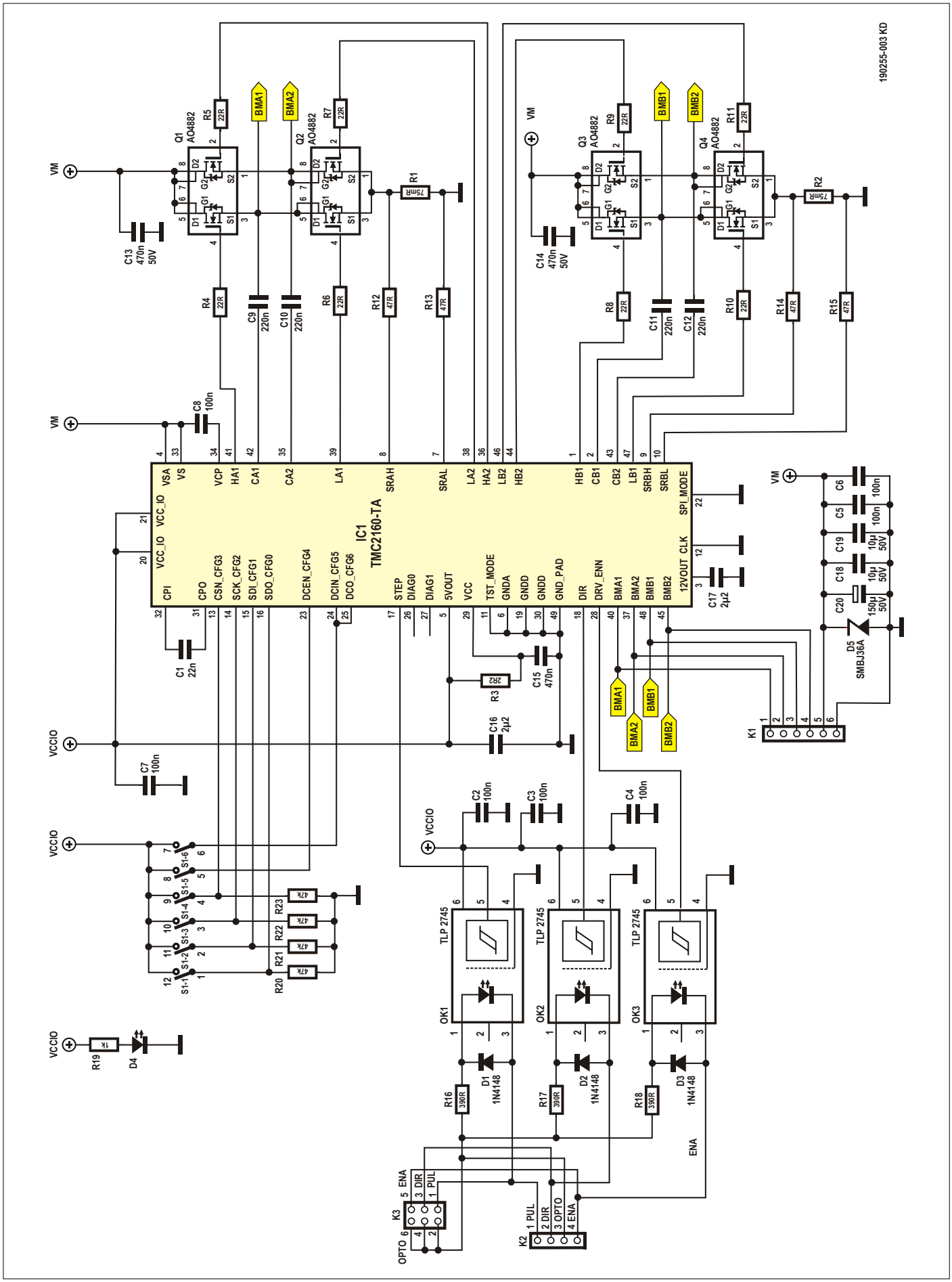


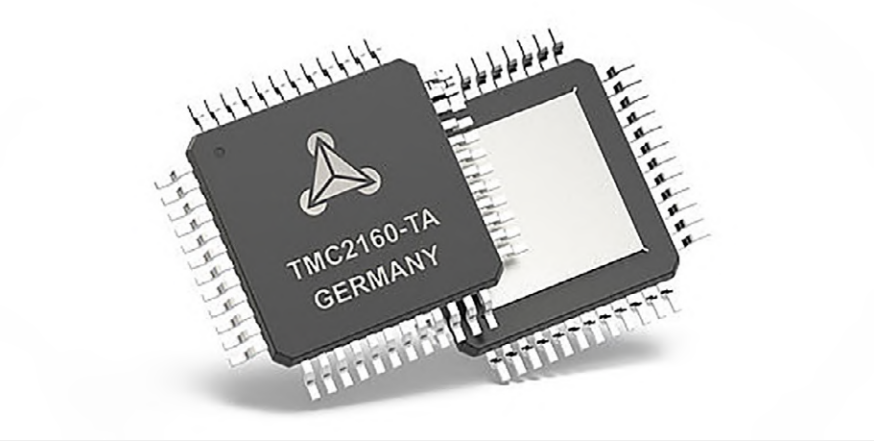
Figure 2. Basically, the circuit consists of motor driver IC1, two MOSFET H-bridges, three optocouplers, and six DIP switches.

Table 1. Microstep resolution for step input.		
S1.2	**S1.1**	**Microstep setting**
off	off	8
off	on	16
on	off	32
on	on	64

Table 2. Run current settings.			
S1.5	**S1.4**	**S1.3**	**Run current RMS / Peak [A]**
off	off	off	1.6 / 2.2
off	off	on	1.8 / 2.5
off	on	off	2.0 / 2.8
off	on	on	2.2 / 3.1
on	off	off	2.4 / 3.4
on	off	on	2.6 / 3.7
on	on	off	2.8 / 4.0
on	on	on	3.1 / 4.4

Table 3. Chopper mode and hold current.	
S1.6	**Chopper Mode and Hold Current**
off	SpreadCycle with no hold current reduction
on	StealthChop2 with 50% hold current reduction

Trinamic transforms digital information into physical motion



When digital information can be transformed into precise physical motion, previously infeasible use cases suddenly become possible, pushing the 4th Industrial Revolution: Advanced robotics, automation, prosthetics, and IoT, to name just a few.

Nevertheless, efficiently turning digital information into physical motion requires more than just transferring data into movement. Motor control technology needs to be easy to implement. It demands flexibility to support evolving device capabilities. It requires learning capabilities to turn repetitive automation into intelligent movement. And it needs to fit increasingly small form factors to handle new use cases.

Trinamic achieves these critical requirements by making the most advanced motion control as easy as 1-2-3. Their long history has taught them how to optimize performance, drive miniaturization, and turn-key motor characteristics into competitive advantages for your devices.

times the rated motor voltage for best results) to the motor windings. This technique allows controlling the average current per phase. Most chopper drivers use a chopping frequency in the 20–50 kHz ballpark. Since the chopping frequency is above the audible range, noise levels are kept low. With a constant fixed chopping frequency, the average output current is controlled by varying the pulse width of the output pulses, a technique known as pulsewidth modulation (PWM).

Besides the classic constant off-time chopper, the TMC2160 also features two sophisticated SpreadCycle and StealthChop2 chopper modes, which ensure noiseless operation combined with maximum efficiency and best motor torque. StealthChop2 is a voltage chopper-based principle. It especially guarantees that the motor is quiet in standstill and in slow motion, except for noise generated by ball bearings. Unlike other voltage-mode choppers, stealthChop2 does not require any configuration. It automatically learns the best settings during the first motion after power up and further optimizes the settings in subsequent movements. An initial homing sequence is enough for learning. StealthChop2 allows high motor dynamics, by reacting at once to a change of motor velocity.

SpreadCycle is an advanced cycle-by-cycle chopper mode. It offers smooth operation and good resonance damping over a wide range of speed and load. The SpreadCycle chopper scheme automatically integrates and tunes fast decay cycles to guarantee smooth zero crossing performance and allows for high dynamics and highest peak velocity at low vibration.

Other advances of the TMC2160 include MicroPlyer step interpolation, resonance damping for mid-range resonances, DcStep load-dependent speed control, StallGuard high-precision sensor-less motor load detection and CoolStep current control for energy savings up to 75%. For more information, please refer to the TMC2160 product page and data sheet [3].

Advertisement



TRINAMIC
MOTION CONTROL

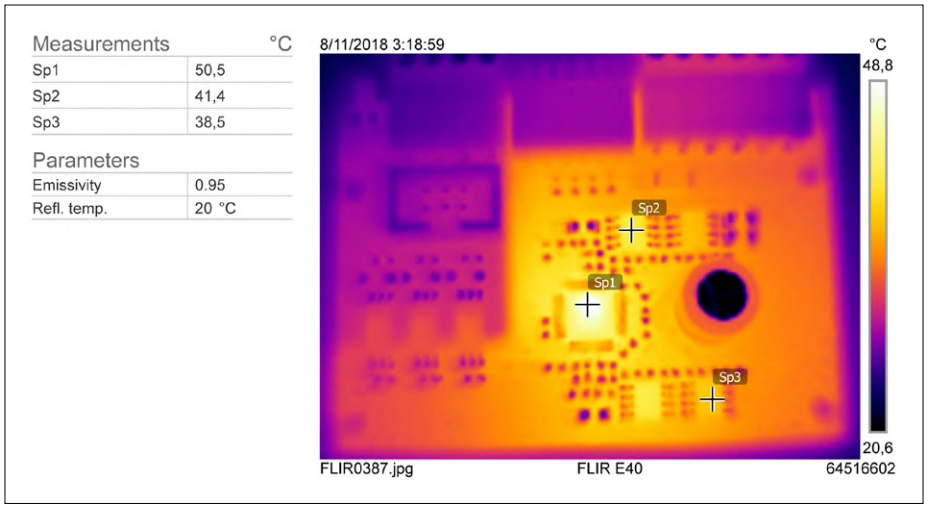


Figure 3. Thermal image of the motor driver board controlling a NEMA 17 stepper motor with a current of 1.6 A RMS in SpreadCycle chopper mode.

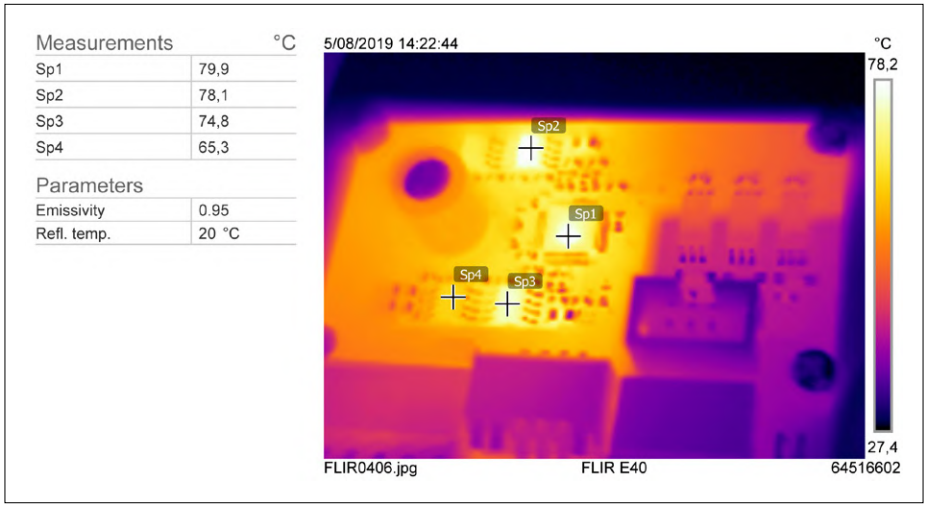
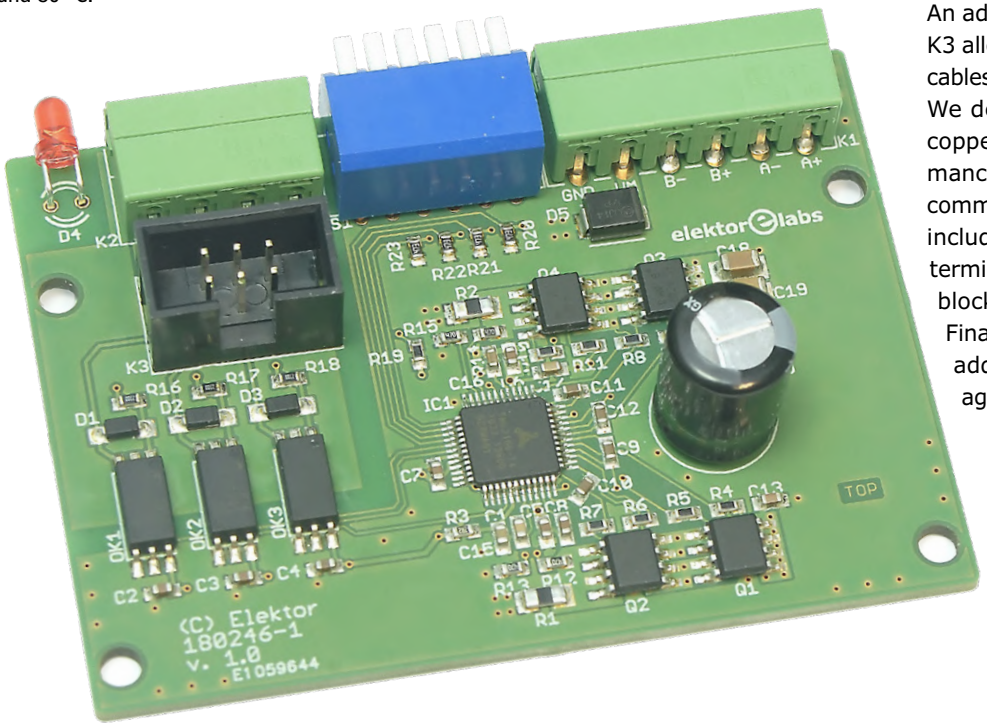


Figure 4. With a current of 3.1 ARMS and the motor pulling a 12 kg weight, the temperature rose to around 80 °C.



About the circuit diagram

See **Figure 2**. Our TMC2160 stepper motor driver board is largely based on the Trinamic TMC2160 breakout board [4]. While most of the connections to the TMC2160 are identical and we used the same parts to build the H-bridges, there are quite a few differences.

To allow the TMC2160 to operate in standalone mode, we tied pin 22 (SPI Mode) to Ground. As a result, pins 13–16 and 23–25 now become configuration inputs CFG0-CFG6 which we connected to a 6-way DIP switch, S1 (see **Tables 1, 2 & 3**). Since CFG4 and CFG5 have internal pulldown resistors, we decided to connect external pulldown resistors to the other configuration pins CFG0-CFG3 as well. CFG5 and CFG6 are both connected to S1.6 since driving them to a different level would result in CFG6 drawing a significant current of 20 mA. This would cause the internal linear voltage regulators of the TMC2160 to produce excessive heat.

Type TLP2745 high-speed optocouplers with digital outputs (OK1-OK3) provide galvanic isolation between the control inputs (Enable, Step and Dir) and the TMC2160 driver IC. The inputs are optimized for 5-volt swings. Higher voltages can also be used if extra resistors are connected in series with the control inputs. Please note that we changed the name of the 'Step' signal to 'Pul' (pulse) to better match the signal names printed on the enclosures of common industrial stepper motor drivers.

An additional 6-pin boxheader connector K3 allows for easy connections using flat-cables to our CNC310 breakout board [7].

We designed a 4-layer PCB with large copper planes for better cooling performance and used a form factor similar to common industrial stepper motor drivers including Phoenix Contact's "pluggable" terminal blocks, a piano-style DIP switch block and a power indicator LED.

Finally, a type SMBJ36A TVS diode was added to protect the circuit from voltage surges on the supply voltage rails.



COMPONENT LIST

Resistors

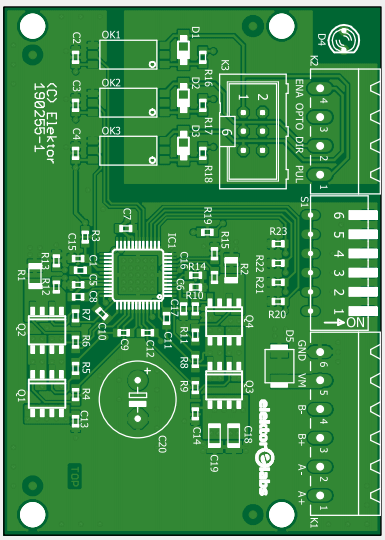
Default: SMD 0603, 1%, 100mW
R1,R2 = 75mΩ, 1W, 1206
R3 = 2.2Ω
R4,R5,R6,R7,R8,R9,R10,R11 = 22Ω
R12,R13,R14,R15 = 47Ω
R16,R17,R18 = 390Ω
R19 = 1kΩ
R20,R21,R22,R23 = 47kΩ

Capacitors

Default: SMD 0603, 50V
C1 = 22nF
C2,C3,C4,C5,C6,C7,C8 = 100nF
C9,C10,C11,C12 = 220nF
C13,C14 = 470nF
C15 = 470nF, X5R, 16V
C16,C17 = 2.2µF, X5R, 25V
C18,C19 = 10µF, X5R, 50V, 1206
C20 = 150µF, 50V, 5000h/105°C

Semiconductors

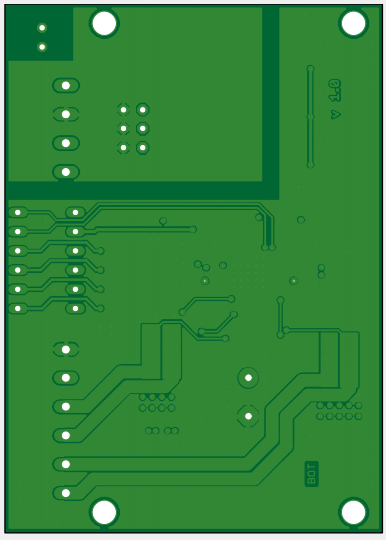
D1,D2,D3 = 1N4148, SOD-123
D4 = LED, 3mm, red
D5 = SMBJ36A
IC1 = TMC2160TA (TQFP48-EP) Trinamic



OK1,OK2,OK3 = TLP2745(TP,E
Q1,Q2,Q3,Q4 = AO4882

Miscellaneous

K1 = MC1,5/6-G-3,81 + MC 1,5/6-ST-3,81, Phoenix Contact
K2 = MC 1,5/4-G-3,81 + MC 1,5/4-ST-3,81, Phoenix Contact



K3 = 6-pin boxheader, 2×3, vertical, 0.1" pitch
S1 = 6-way DIP switch, piano-type, 0.1" pitch

When the driver is connected to a supply rail through long leads, on power-up the parasitic induction of the leads may cause voltage surges in combination with the MLCC capacitors C18 and C19 [5]. Please note that the circuit is not reverse-polarity protected, so be careful when connecting the power supply leads.

We ran some tests with a thermal camera covering various load conditions. With a 24-V power supply, the temperature of the TMC2160 and the MOSFET transistors reached 50 °C and 41 °C respectively, while driving a NEMA 17 stepper motor at various speeds. The motor current was set to 1.6 A RMS using the SpreadCycle chopper (**Figure 3**).

The stepper motor was then replaced with a NEMA 34 motor and we cranked up the supply voltage to 32 V. The current was set to 3.1 A_{RMS} and the motor was pulling a 12-kg (26 lb) kettle bell. Then the temperature of the TMC2160 and MOSFETs rose to 80 °C and 78 °C respectively (**Figure 4**).

When driving larger stepper motors like NEMA 23 and NEMA 34 types, we recommend sticking a heatsink to the bottom of the PCB or to use a small fan for forced air cooling. ⏮

190255-01

@ WWW.ELEKTOR.COM

→ **TMC2160 Motor Driver Board - Bare PCB (190255-1)**
www.elektor.com/19010

→ **TMC2160 Motor Driver Board - Partly Assembled Module (190255-91)**
www.elektor.com/19009

Weblinks

[1] How to make your 3D printer smart and silent with the TMC2130!: www.youtube.com/watch?v=sPvTB3irCxQ
[2] Marlin firmware for 3D printers: http://marlinfw.org/
[3] Trinamic TMC2160 product page: www.trinamic.com/products/integrated-circuits/details/tmc2160-ta/
[4] TMC2160 Breakout board: www.trinamic.com/support/eval-kits/details/tmc2160-bob/
[5] SDGEE #061 Ceramic Capacitors will Blow Up your PCB: www.youtube.com/watch?v=2287hcWFqe8
[6] 'Wolverine' CNC Gantry Robot - Part 1: www.elektormagazine.com/180246-01
[7] 'Wolverine' CNC Gantry Robot - Part 2: www.elektormagazine.com/180246-B-02
[8] Project at Elektor Labs: www.elektormagazine.com/labs/tmc2160-motor-driver-board-190255



SigFox and the IoT (1)

Part 1: Sigfox data over radio using a Maker board

By **Frank Schleking** and **Bernd vom Berg** (Germany)

Wireless standards such as Sigfox and LoRa are available whenever battery-powered sensors need to transmit data to an Internet gateway. Sigfox scores high with a uniform and well-developed network. All the user must do is program and log on the sensor nodes in order to visualize their output data on the Internet. As we show in this series, cost-effective hardware and free software are enough to realize the first Sigfox projects. In the first part we present the compact Arduino MKR FOX 1200 module with an integrated Sigfox transceiver and a corresponding motherboard.

5G networks are currently on everyone's lips. They enable massive amounts of data to be transmitted by radio in the shortest possible time. However, the hardware and software required to set up these networks is colossal. The base station range is short typically and the energy consumption of the network as a whole is relatively high.

What does the situation look like if you only want to transmit just a few sensor and/or position data, for example, for application areas like Smart City, Smart Farming, Smart Transport or Smart Hobby?

The sensors in the field need to function maintenance-free and remain self-supporting for several years with only one battery charge. The range (i.e. distance to the nearest base station) should be several kilometres and in many cases inexpensive data transfer across national borders is required. This is the field of radio-based sensor/actuator networks, now referred to as 0G networks. Such communication systems are currently emerging worldwide, with three major concepts budding in the meantime: Sigfox, LoRa and NB-IoT.

Sigfox 'maker' (MKR) boards

A wide variety of maker boards provides a good entry to the field of Sigfox communication, offering fully functional hardware and versatile software drivers. The development effort is reduced considerably, and all you have to do is develop your own application 'around it'.

Such maker boards are available as plug-in cards for various microcontroller platforms, including the Raspberry Pi. If however Sigfox stations are to be developed that can be powered at extremely low energy levels over several months, even years by a battery pack, then solutions with a Raspberry Pi are out of the question, because these systems are almost always dependent on an AC power supply owing to their high power consumption. Alternatively, large batteries are used in connection with powerful solar generators.

On the other hand, the Arduino MKR FOX 1200 maker board (**Figure 1**) provides a particularly simple and, above all, extremely energy-efficient launchpad. The concept of the well-known Arduino Zero is used and simply supplemented by the complete Sigfox functionality.

All software development tools from the Arduino world can be used, hence a powerful library is available for the actual Sigfox communication which can easily be integrated into your own software.

With the purchase of the MKR FOX 1200 comes an annual subscription to Sigfox, permitting you to operate the board for one year free of charge within the worldwide Sigfox network.

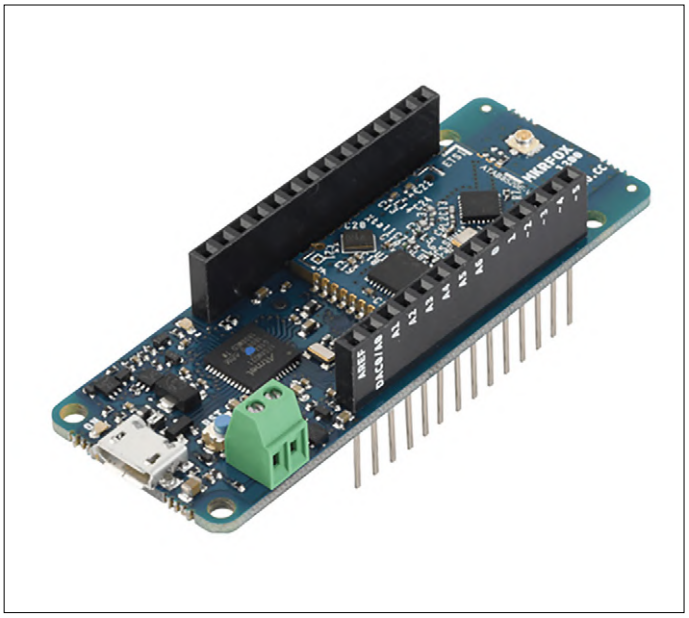


Figure 1: The Arduino MKR FOX 1200 sports a transceiver to implement Sigfox data-over-radio.

Our article series

In the **first part** of this article series we introduce the MKR FOX 1200 hardware. In addition, we have developed a motherboard for the MKR FOX 1200 that provides some interesting additional functions for the user. Furthermore, we will consider the programming of the board and program test and example applications which can be downloaded as usual from the ElektorLabs magazine website [1].

In the **second part of** the series we will focus on Sigfox. We will explain the underlying communication concept and how individual stations are integrated into the worldwide Sigfox network.

Then we will deal with the Sigfox cloud and answer the question of how messages are actually sent and received in the Sigfox backend.

The **third part** of the series we discuss details on reception, evaluating and displaying of Sigfox data from stations worldwide via the Internet. Freeware tools can be used to design and operate your own individually designed dashboards (screen interfaces).

MKR FOX 1200 — the hardware

The official homepage of the MKR FOX 1200 [2] has a lot of additional information (Getting Started, technical information, circuit diagram, FAQ). **Figure 2** shows a block diagram of this board.

The maker FOX 1200 is based on the Arduino Zero with the ATSAMD21G18A-48 microcontroller from Microchip. This chip integrates an ARM-based 32-bit Cortex M0 processor, complemented by microcontroller-typical on-chip peripheral units such as timer/counter, AD and DA converters, UART interface, watchdog, RTC and more.

The Sigfox capability is provided by the single chip SIGFOX RF Transmitter type ATAB8520E. This component contains everything required for operation in the Sigfox network: a dedicated microprocessor, the complete RF generation including the transmit/receive stages and a complete Sigfox protocol stack (cast in hardware). Only the RF output filter and the antenna have to be connected externally.

This Sigfox modem is controlled from the SAMD21 via an SPI bus interface.

The remaining modules on the MKR FOX 1200 board are used for the flexible power supply and for the realization of the USB interface to the PC or laptop.

Because of these two chips, the MKR FOX 1200 has a core feature that is essential for (virtually) self-sufficient IoT stations. Besides high computing power, the board is marked by extremely low power consumption in sleep mode of just 4 µA (SAMD21G) and 5 nA (ATA8520) [9][10]. In most of the very short active operating phases, the current consumption is 32.7 mA in transmit mode and 10.4 mA in receive mode. This allows the powering of your stations from two AA or even AAA batteries for months or years without replacement.

These types of Sigfox stations are also great to run on small solar panels with battery buffering, “24/7/365”. The total current consumption of a station also depends on the connected sensors and actuators. In this area, too, energy-saving designs abound; especially since these units do not have to be oper-

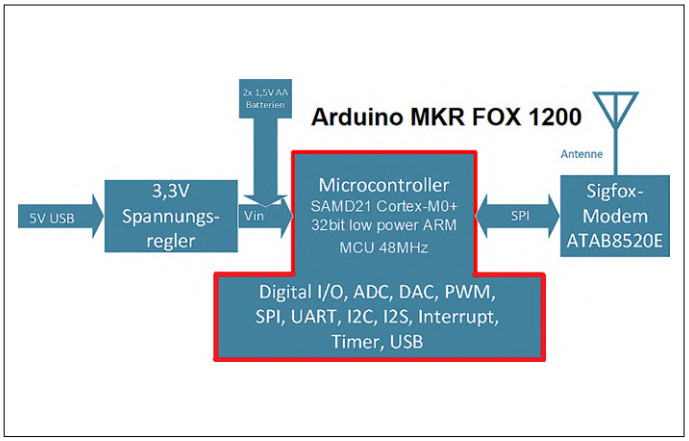


Figure 2: Arduino MKR FOX 1200 block diagram.

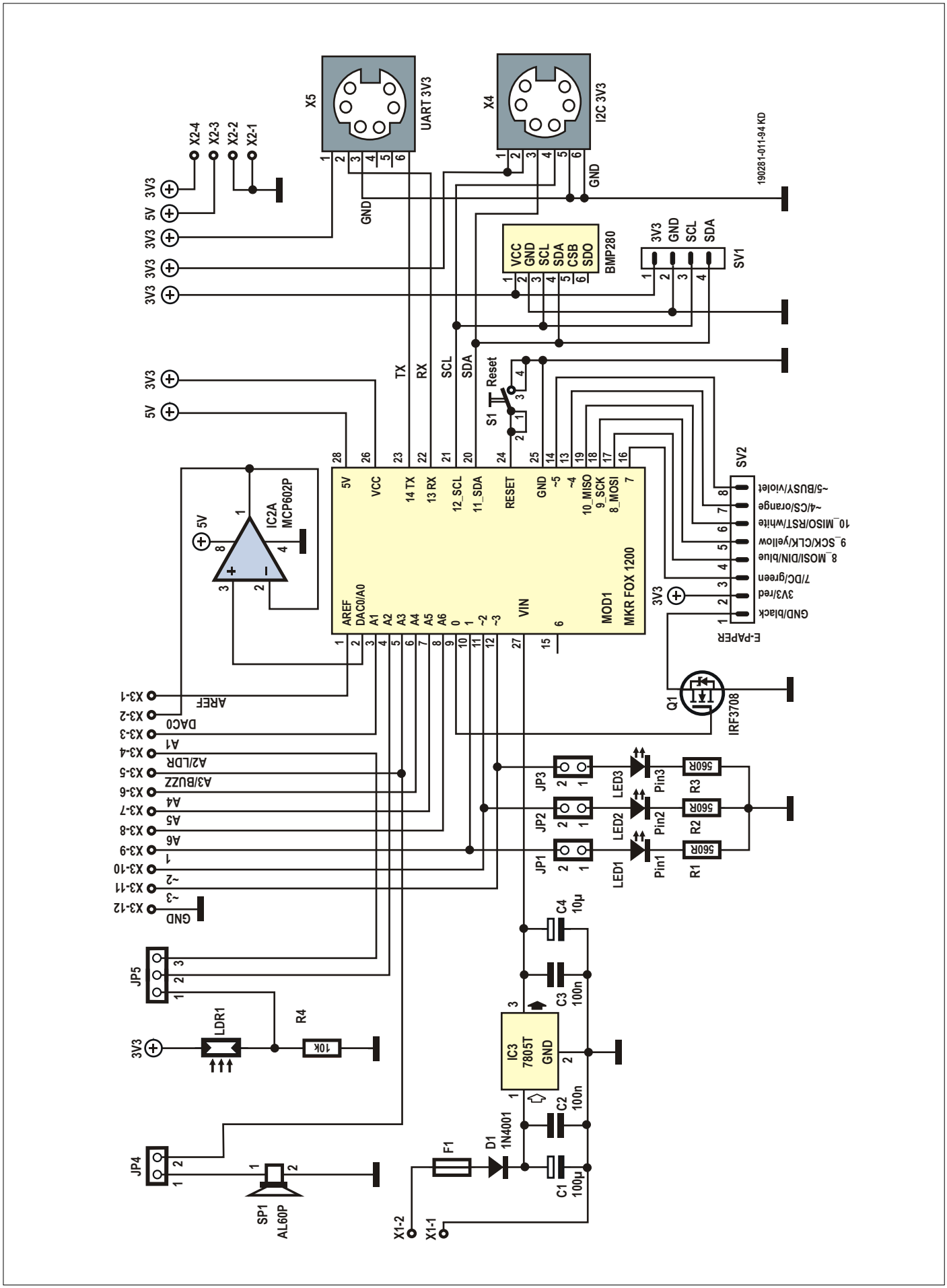


Figure 5: Circuit diagram of the motherboard for the MKR FOX 1200.

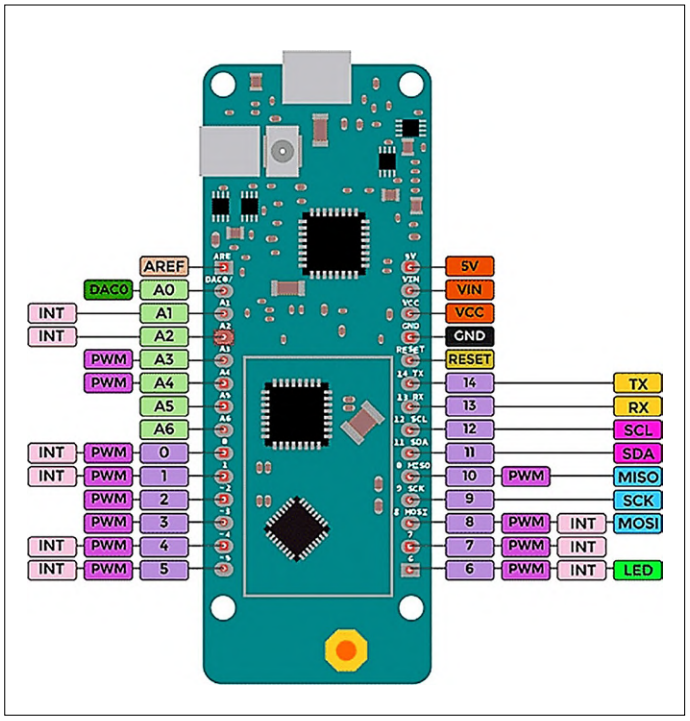


Figure 3: Pinout of the MKR FOX 1200 [3].



Figure 4: The motherboard with the MKR FOX 1200 attached.

ated permanently but get activated only when needed and at short notice.

The basis for the development of an application with the MKR FOX 1200 is the pinout (see **Figure 3**). The microcontroller characteristic of the SAMD21 chip is clearly visible here: The ports of major on-chip peripheral units are bonded out to pins and almost all ports are double or even triple occupied. The respective function of a pin is then determined by the application software.

MKR FOX 1200 — motherboard

In order to effectively put the MKR FOX 1200 into operation and to be able to quickly realize our first own applications, we have developed a motherboard on which the MKR FOX 1200 is simply plugged (**Figure 4**). **Figure 5** shows the corresponding circuit diagram.

The power supply of the MKR FOX 1200 board can now be realized in three different ways**.**

Power supply via USB interface

First, the board can be powered via the USB interface, which is also used to download the program from the computer or to implement the serial communication between the user program and the computer.

The green 2-way terminal block on the board

Here a battery pack (two pieces type AA or AAA connected in series) can be connected to supply the board. The polarity is printed under the terminal block. However, this 3-V supply is not suitable then for supplying the peripheral components on the motherboard.

The VIN connection

Here an external regulated fixed voltage of +5 V can be connected to supply the board. If a voltage is connected here, the power supply is automatically switched off by the USB connection.

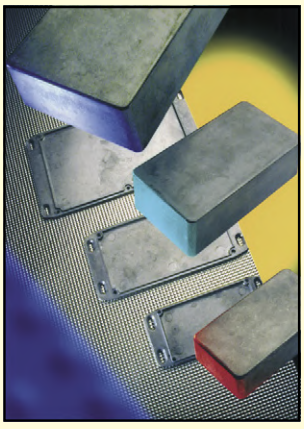
Advertisement

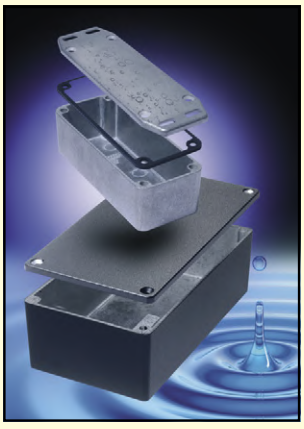
HAMMOND MANUFACTURING®

Die-cast enclosures flanged and waterproof

More than 5000 different enclosure styles:
hammfg.com/electronics/small-case

01256 812812
sales@hammond-electronics.co.uk





The 5 V connection
That's an **output**. Here 5 V (unregulated) comes out of the board if the board is supplied via the USB interface or via VIN. The voltage is passed through directly from the USB interface or from VIN. The maximum output current is 500 mA.

The VCC connector
That's an **output** supplying the following voltages:

- 3.3 V from the on-board voltage stabilizer if the board is powered via the USB interface or via VIN. Maximum output current: 500 mA;
- the voltage from the two batteries if the board is powered by the two batteries.

The operating voltage LED: LED ON
The operating voltage LED (green, next to the USB port) only lights up when the board is powered via the USB interface or externally via VIN. When powered by batteries, the LED is off (to save power).

Table 1: MKR FOX 1200 module pin assignment on motherboard.

Pin No.	Pin Designation	Pin Assignment
1	AREF	Input, standard function — connection to terminal X3-1; inject an external reference voltage, max. 3.3 V
2	DAC0/A0	D/A converter output - via impedance converter at terminal X3-2; output voltage 0 to 3.3V, max. output current: 30 mA.
3	A1	On terminal X3-3; standard function
4	A2	On terminal X3-4; standard functionAlternative: brightness measurement using LDR can be connected through jumper JP5
5	A3	On terminal X3-5; standard function.Passive piezo buzzer can be added via jumper JP4
6	A4	On terminal X3-6; standard function
7	A5	On terminal X3-7; standard function
8	A6	On terminal X3-8; standard function
9	0	Output, switches the power supply for the ePaper display on and off on the board.
10	1	On terminal X3-9, standard function;LED switchable to status display via JP1
11	~2	On terminal X3-10; standard function. LED for status display switchable via JP2
12	~3	On terminal X3-11; standard function. LED for status display switchable via JP3
13	~4	Required for operation of the ePaper display: CS
14	~5	Required for operation of the ePaper display: BUSY
15	6	Not wired; yellow LED on MKR FOX 1200 connected here.
16	7	Required for operation of the ePaper display: DC
17	8_MOSI	Required for operation of the ePaper display: DIN
18	9_SCK	Required for operation of the ePaper display: CLK
19	10_MISO	Required for operation of the ePaper display: RST
20	11_SDA	Required for operation of the I^2C bus: SDA
21	12_SCL	Required for operation of the I^2C bus: SCL
22	13 ~ RX	UART data input. Connected to the 6-pin UART mini DIN socket X5 (addressed via Serial1 independently of the USB interface)
23	14 ~ TX	UART data output. Connected to the 6-pin UART mini DIN socket X5 (addressed via Serial1 independently of the USB interface)
24	RESET	External reset button connection
25	GND	GND connection
26	VCC	3.3 V output voltage (see text)
27	VIN	Input for the stabilized 5 V operating voltage (see text)
28	5V	5 V output voltage (see text)

We feed the MKR FOX 1200 board from the motherboard via a stabilized 5-V voltage, enabling the entire supply of the system to initially take place via a plug-in power supply.
Some port pins of the MKR FOX 1200 are already connected to sensors or actuators on the motherboard, so that even small applications can be developed like:

- a light-sensitive resistor for measuring ambient brightness;
- a passive piezo buzzer for outputting tones;
- an impedance converter for buffered output of an analogue voltage (OPV MCP602P). This impedance converter is used to increase the maximum achievable output current of the DAC0 from 7 mA to 30 mA;
- LEDs connected to three port pins for visualization (in test mode), which can be switched off via jumpers.

As more complex I/O peripheral units, we use an extremely power-saving 2.9-inch' ePaper display and the I2C bus for local on-site display. To the latter we have connected the BMP280, a barometric absolute air pressure sensor that also measures the ambient temperature. In addition, the I²C bus is routed out through a 6-pin mini DIN socket, so that further I²C components can be easily connected here.
The serial UART interface is also connected to a 6-pin mini DIN socket.
An external reset button completes the equipment of the motherboard.
Table 1 shows an overview of the exact wiring of the I/O pins of the MKR FOX 1200.

Attention please

In contrast to most other Arduino boards, the MKR FOX 1200 board runs off **3.3 V**. Consequently the I/O ports only work with input/output voltage levels of 3.3 V!
Higher voltages at the I/O ports can damage the board. It should also be noted that the power output stage of the Sigfox modem can be damaged during transmission attempts without an antenna.
If the ePaper display, the I²C bus and/or the UART interface are not required, and the corresponding freed I/O port pins can be used normally in the standard function.

The software

How can we implement the software for our sensor node? Four steps are necessary:

- install the Arduino IDE on the computer and adapt it to the board;
- install the additional libraries required for operating the peripheral units;
- write a set of general routines that can be used to control the periphery: operating the AD and DA converters, operating the I²C bus, operating the UART interface, etc.
- turn to the 'special functions' for implementing Sigfox communication.

Installing the Arduino IDE

The latest version of the Arduino IDE (development environment) can be downloaded directly from the Arduino homepage [6]. To install the IDE permanently on your computer, select the version Windows ZIP file for non admin install in the right box, it's approximately 108 MB.

Installation and configuration for the board is trouble-free [7] [8]. You only have to know the respective board; here the MKR FOX 1200 with SAMD21 controller. The driver library for the SAMD boards can be installed in the Arduino IDE via Tools à Board à Board Administrators.

Required libraries

When first creating Arduino programs (sketches), you can rely on many ready-made functions, grouped together in libraries. This reduces the development effort considerably and you can achieve very fast executable results.
In addition to the libraries already contained in the Arduino IDE, the Internet contains many other function collections for various external peripheral units (displays, sensors, etc.), which can easily be linked to your own project.
For the operation of the external peripheral units used here, the following additional libraries must be installed:

- for the ePaper display: the GxEPD and Adafruit GFX libraries;
- for the BMP280 air pressure sensor: the library BMx280MI;
- in the second part of the series we install the Sigfox library.

The first two libraries can be found in our software package [1]. The Sigfox library can be installed later via the library manager of the Arduino IDE (Tools ➞ Manage Libraries...).

Advertisement



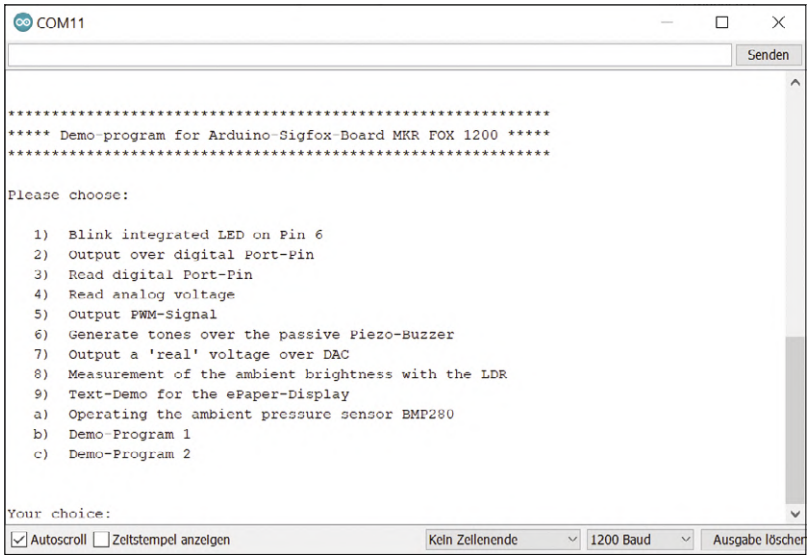


Figure 6: Menu of the demo and test program.

A simple guide to installing Arduino ZIP libraries can be found at [5] for example.

Operating software for the MKR FOX 1200

A good reference to the Arduino command set can be found under [4]. We have developed a small Arduino sketch to try out some basic features of the board (included in our software package [1]).

At the same time this sketch serves as a test program for the motherboard. The individual software modules can serve as starting points for further in-house developments.

Figure 6 shows the menu of this program; the individual functions can be called via the serial interface. To do this, first start the Arduino IDE.

Now open the demo sketch in the IDE. Do not load the sketch onto the board yet, but first start the IDE-internal serial monitor by pressing Ctrl + Shift + M.

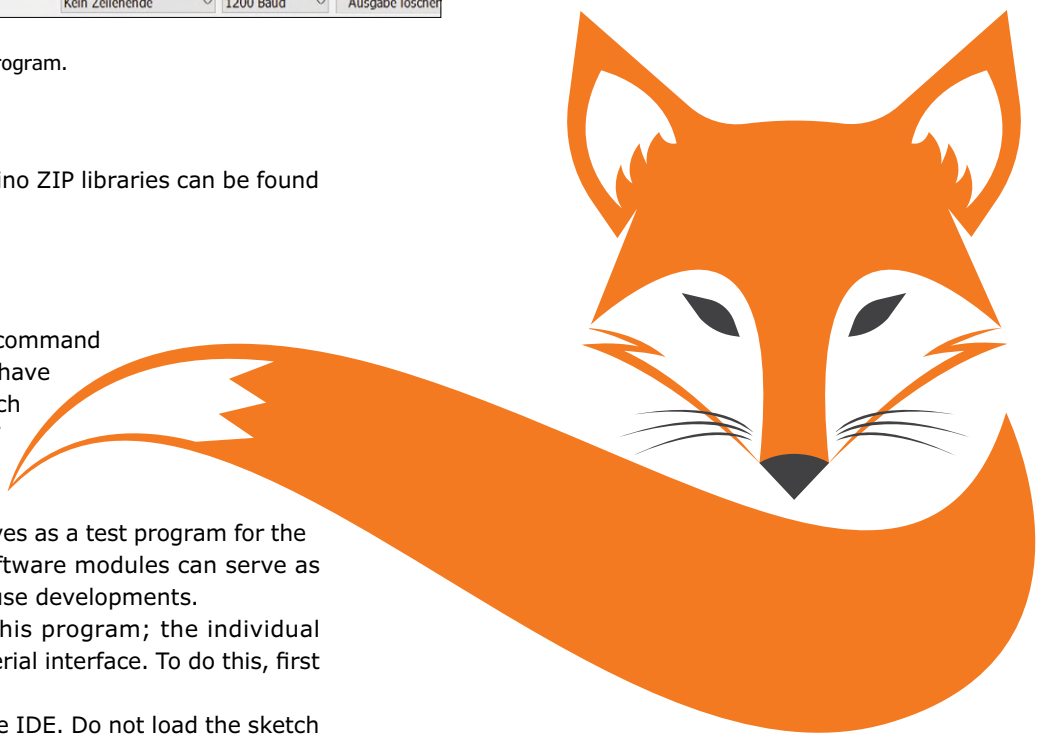
Now have the sketch compiled, downloaded and started. Switch to the Monitor window (Ctrl + Shift + M). The menu from Figure 6 should now appear there.

Now you can select one of the demo examples (entries in the upper monitor line always end with RET).

The Sigfox communication software

In the next part of the series, we will discuss the use of the special functions for setting up and handling Sigfox communication. ⏮

190281-02



Web Links

[1] Software download: www.elektormagazine.com/190281-02

[2] Official MKR FOX 1200 homepage: https://store.arduino.cc/arduino-mkrfox1200

[3] MKR FOX 1200 pinout: https://arduinoplus.ru/ref/raspinovka-plat/raspinovka-arduino-mkr-fox-1200/

[4] Arduino command set: www.arduino.cc/reference/en/

[5] Install additional libraries: www.arduino.cc/en/guide/libraries\

[6] Arduino IDE latest version: www.arduino.cc/en/main/software

[7] Arduino IDE installation: https://funduino.de/anleitung

[8] Arduino IDE installation and introduction: www.youtube.com/watch?v=4Ih39hGcPzg

[9] SAMD21 microcontroller family datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/SAMD21-Family-DataSheet-DS40001882D.pdf

[10] Sigfox transceiver datasheet: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-9409-Smart-RF-ATA8520E_Datasheet.pdf

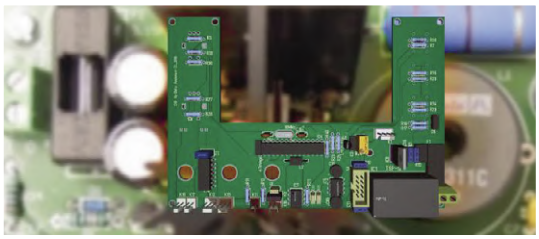
Elektor Labs Pipeline

By **Clemens Valens** (Elektor Labs)

Sensors, batteries and good connectivity are important for successful IoT applications. Add to that a neat user interface (UI) and you're on top of the world. Here are great examples of projects related to these subjects and more.

Add a User Interface with Presets to Your Power Supply

A few years ago Elektor published a 30-V, 3-A adjustable benchtop power supply dubbed UniLab2. To keep the total outlay down it had a fairly Spartan user interface. This Elektor Labs project offers a slick user interface for the UniLab2 by adding rotary encoders and a 4-line LCD. On top of that, you get handy presets for 3.3-V, 5-V and 12-V output voltages.



@ Elektor Labs: www.elektormagazine.com/labs/1836

DIY Spot Welding for Battery Packs

It appears that Europe is way, way behind China when it comes to the production of batteries. Billions will have to be invested to keep up. Although this Elektor Labs project is a drop in the battery builders' ocean, it may prove highly useful for those who don't want to wait for investors and decision makers to catch up.



@ Elektor Labs: www.elektormagazine.com/labs/1817

Build a Pin Diode Based Radiation Detector as an MQTT Client

More than 30 years after the accident that destroyed the nuclear power-plant in Chernobyl (Ukraine), the radiation it created still has measurable consequences in regions hundreds of miles away. This project came to life when Sweden launched a nationwide project in 2018 to analyse dried mushrooms to determine the accumulated radiation level in the soil.



@ Elektor Labs: www.elektormagazine.com/labs/1553

Build a PIR Sensor That Can Send Email

Cheap, powerful and equipped with Bluetooth and Wi-Fi, the ESP32 has quickly become a popular development platform for homegrown IoT applications. Here is one that transforms an ordinary PIR sensor into an almost intelligent, communicating entity. This alarm may be less suited for people who check their email only once a month. ⏮

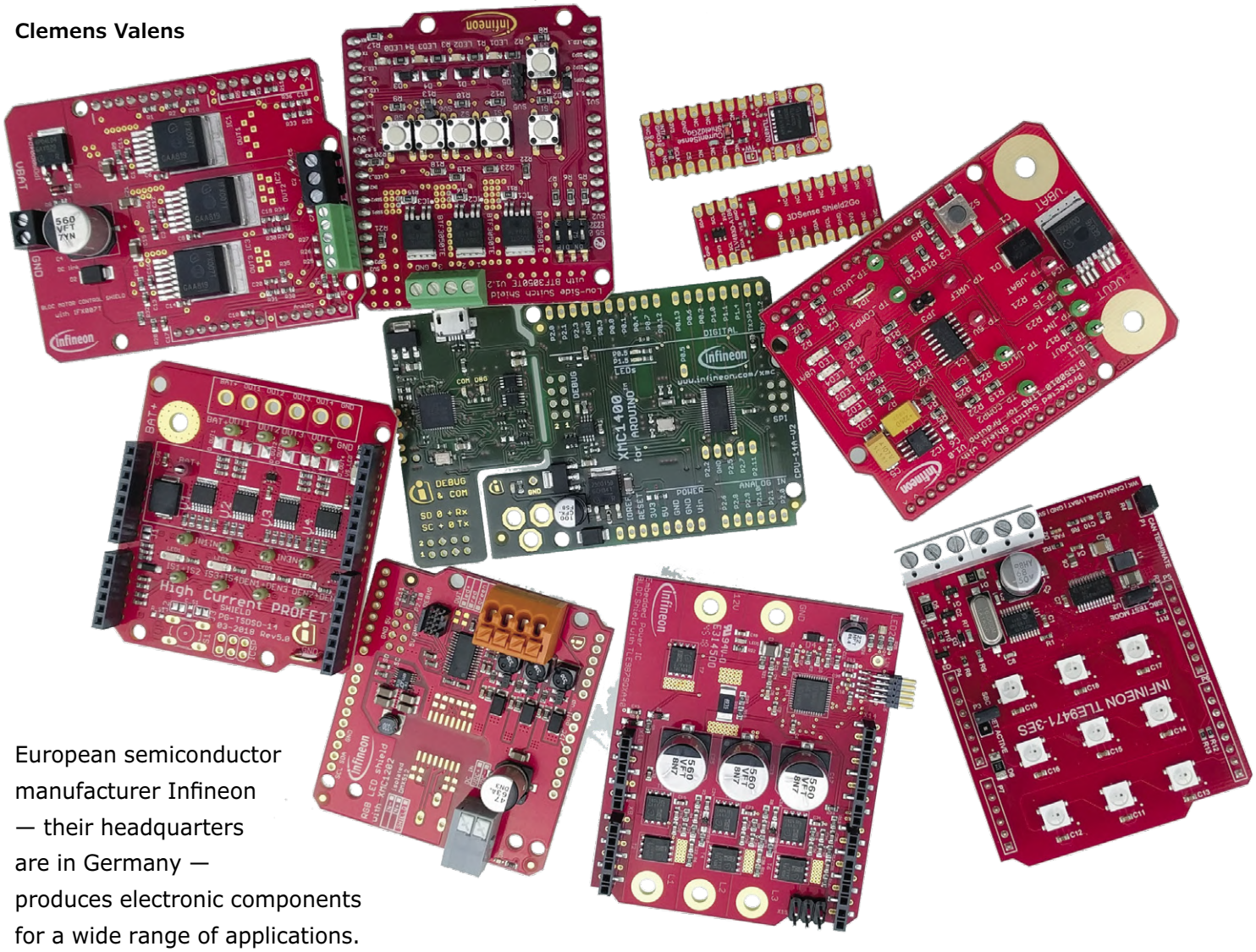
180571-F-01



@ Elektor Labs: www.elektormagazine.com/labs/1827

Having Fun with Development Boards from Infineon

Clemens Valens



European semiconductor manufacturer Infineon — their headquarters are in Germany — produces electronic components for a wide range of applications. Evaluation and development boards supporting these devices ease the life of design engineers. Now that Arduino has become the de facto standard for fast prototyping tools, Infineon too has created a line of Arduino Shields. Let's have a look at a few of those.

Infineon for makers

If my memory is accurate, it was during an Infineon presentation that I first saw 'Makers' as a target audience on a slide during a presentation at a professional electronics trade show. These were no idle words and since then they have been busy developing Arduino-compatible evaluation boards ('Shields') for their components. Since Infineon specializes in automotive and power electronics, and digital security, it will come as no surprise that these boards revolve around these application domains. Respecting common maker-scene conventions, the shields are supported by open-source Arduino libraries published at GitHub and the hardware designs are available as Eagle files. Besides Arduino-compatible products, more traditional tools exist as well. For the serious Infineon microcontroller application programmer there is the free, Eclipse-based DAVE with tons of example code.

Robust microcontrollers for industrial and automotive applications

Because microcontroller units (MCUs) are at the heart of most applications today, Infineon supplies them too. Their multicore Aurix family is intended for high-security and computation intensive automotive and other applications while the XMC line mostly targets industrial and lighting. Having 32-bit ARM cores, the "XMCs" are rather suitable for makers: the XMC1000 series have Cortex-M0 cores, XMC4000 devices have a Cortex-M4 core. Some specialized parts with a Cortex-M3 core exist also. Besides the standard peripherals — note that at Infineon a USART is called a USIC and a CCU hides a timer/counter (with lots of PWM options and capture/compare) — these MCUs offer advanced PWM functions for motor control and lighting applications. Some even feature a CORDIC engine for fast trigonometric calculations together with a 32-bit hardware divider. The XMC4300 and XMC4800 devices also integrate EtherCAT for real-time control automation over Ethernet.

Currently the XMC1000 family comprises four product lines:

- **XMC1100** – general-purpose;
- **XMC1200** – LED lighting and human-machine interface (HMI) applications;
- **XMC1300** – motor control and digital power conversion, but also features peripherals for LED lighting applications;
- **XMC1400** – the most powerful of the Cortex-M0-based devices, combines the capabilities of the XMC1200 and XMC1300 devices.

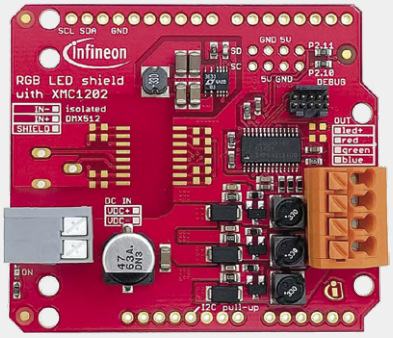
Control power with high- & low-side switches

A high-side switch connects a load to the positive side of a power supply whereas a low-side switch sits between the load and ground. High-side switching is preferable for switching heavy

RGB LED Lighting Shield with XMC1202

To be precise, this shield features an XMC1202-TO28X0016 microcontroller. It integrates a 9-channel 'Brightness and Colour Control Unit' (BCCU), a peripheral that enables flicker-free LED dimming and colour control through pulse density modulation (PDM). The shield can drive up to three LED channels with constant current. The buck converter providing this current can be simulated online. This shield is supported by almost two dozen software examples.

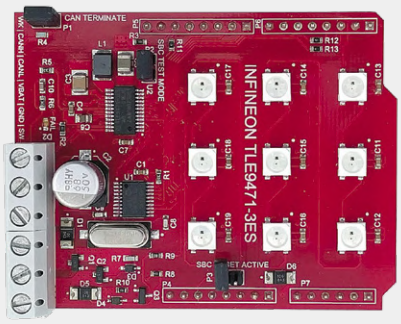
www.infineon.com/cms/en/product/evaluation-boards/kit_led_xmc1202_as_01/



DCDC System Basis Chip Shield with TLE9471-3ES

The TLE9471-3ES shield is built around a so-called DCDC System Basis Chip (SBC) and sports nine WS2812B addressable RGB LEDs, a CAN bus, and a high-side switch with flyback diode capable of switching (inductive) loads up to 2.5 A. The buck converter powering the LEDs, the CAN bus and the high-side driver are all integrated in the TLE9471 SBC. Communication between the SBC and the baseboard is over SPI. To program the SBC a graphical configuration tool is available that spits out the necessary code for use with for instance Arduino. Thanks to the wide variety of on-board peripherals, this shield is a good platform for simulating car electronics.

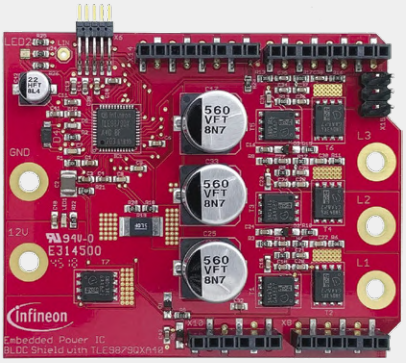
www.infineon.com/cms/en/product/evaluation-boards/sbc-shield_tle9471/

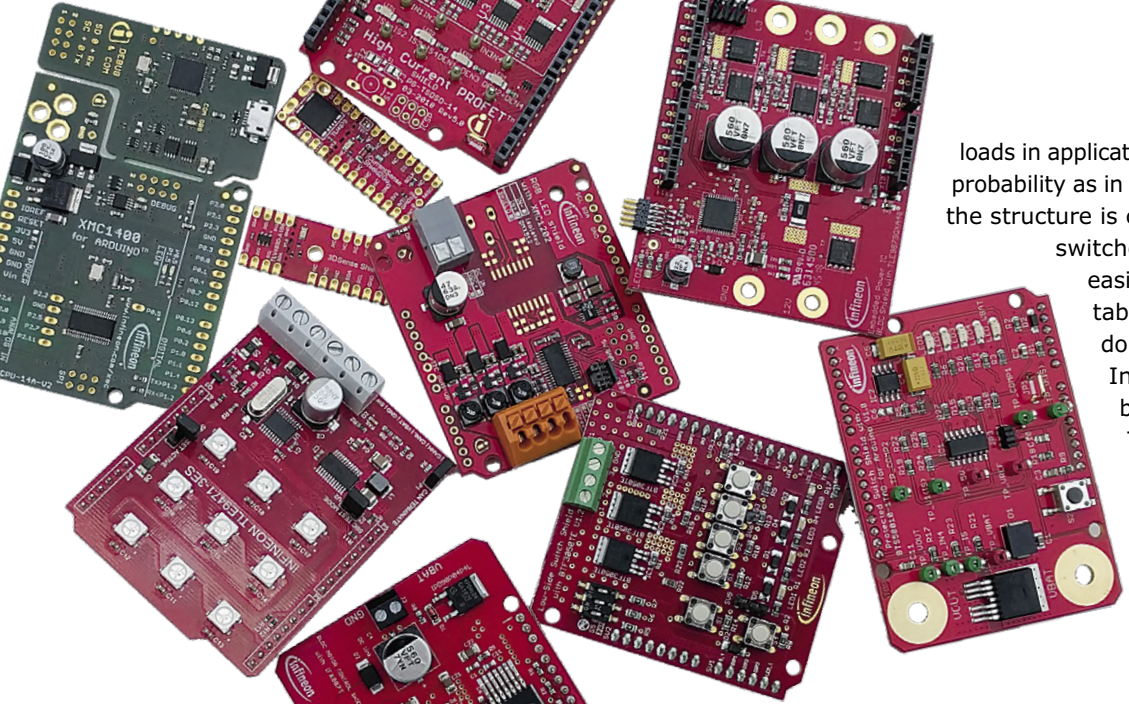


BLDC Shield with TLE9879QXA40

Member of the Embedded Power Microcontroller Boards family, this shield is equipped with a TLE9879QXA40, a three-phase motor driver with an integrated ARM Cortex-M3 core. The IC is assisted by six IPC90N04S5-3R6 power transistors each capable of passing 90 A (with proper cooling, the shield itself is specified for a maximum output current of 10 A). An RGB LED is present on the board too. The MCU comes preprogramed with three motor control algorithms: sensorless Field-Oriented Control (FOC); back EMF and Hall-sensor-based block commutation. An Arduino-compatible baseboard communicates with the TLE9879 over SPI and lets the user change the parameters of the motor control algorithms. Up to four of these shields can be stacked on one baseboard. Example software for Arduino and µVision can be downloaded from the shield's webpage.

www.infineon.com/cms/en/product/evaluation-boards/bldc_shield_tle9879/





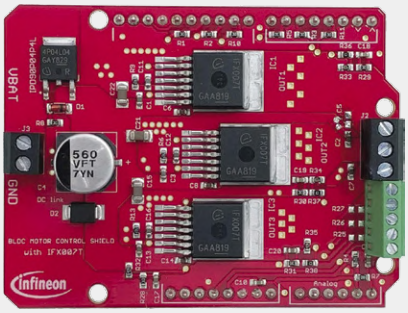
loads in applications with a high short-to-ground probability as in cars or machines where most of the structure is connected to ground. Low-side switches tend to be cheaper and are easier to control, making them suitable for switching light loads and doing PWM control.

Infineon makes a wide range of both high- and low-side switches. The high-side family is based on what they call “PROFET” technology, which stands for PROtected FET. These devices target automotive and industrial applications and can switch currents up to 40 A and volta-

(BL)DC Motor Driver IFX007T Shield

Power MOSFETs are very common in brushed and brushless (BL) DC motor drivers. Infineon has this domain covered with all sorts of parts but due to the limited space available, we will only highlight the (BL)DC motor driver IFX007T Shield. This is a kind of opposite of the TLE9879QXA40-based shield mentioned elsewhere in this article. Instead of an intelligent device running fancy algorithms all by itself, this is a discrete design based on three IFX007T high-current integrated half bridges. The board can control both brushless and brushed DC motors up to 300 W continuous load. The shield has several operating modes, to be activated with a soldering iron by reconfiguring some resistors; the baseboard, an Arduino Uno for instance, must provide the brains. This way of doing things allows connecting Hall sensors for sensor-based control, or, on the other hand, implement sensorless Back EMF or space vector modulation algorithms.

www.infineon.com/cms/en/product/evaluation-boards/bldc-shield_ifx007t/

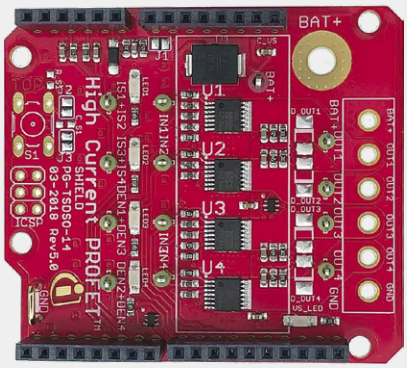


High-Side Switch Shield with BTS700x-1EPP

An interesting family of shields is formed by the four-channel BTS700*x*-1EPP high-side switch shields where ‘*x*’ is ‘2’, ‘4’, ‘6’ or ‘8’. The lower the number, the higher the current they can switch (resp. 20 A, 15 A, 12.5 A and 10 A). Only a check box on the packaging tells you which device is mounted on the board; the boards themselves are not marked. Targeted at high current automotive applications, the switches can be used for relay and fuse replacement, to control and protect outputs of a 12-volt supply, switch loads and measure the load current, and detect no-load conditions.

www.infineon.com/cms/en/product/evaluation-boards/shield_bts7002-1epp/

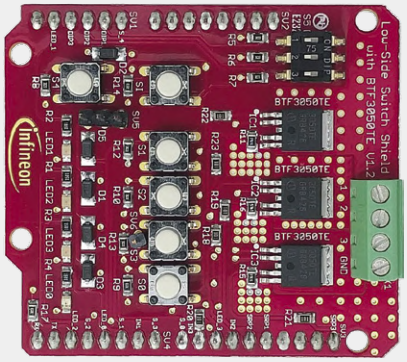
www.infineon.com/cms/en/product/evaluation-boards/shield_bts7004-1epp/



Low-Side Switch Shield with BTF3050TE

For experimenting with low-side switches the BTF3050TE-based shield with its three channels is a good candidate. Each channel can handle resistive, capacitive and inductive loads up to 3 A DC nominally and provide overvoltage, overcurrent, and overtemperature protection. PWM is supported as well (up to 14 kHz). The on-board pushbuttons and LEDs are practical for creating a basic user interface for your application. Example software for Arduino as well as DAVE is available.

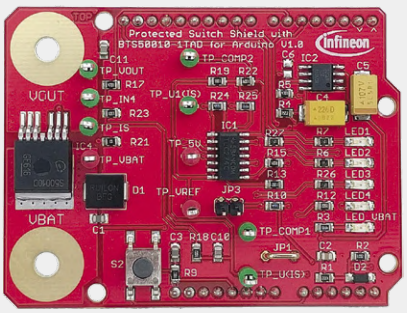
www.infineon.com/cms/en/product/evaluation-boards/shield_btf3050te/



12V Protected Switch Shield with BTS50010-1TAD

The BTS50010-1TAD high-side switch is an N-channel device with an on resistance of 1 mΩ and capable of delivering 40 A. It can be controlled directly from a microcontroller running at 3.3 V or 5 V. Built-in protections include loss-of-ground and loss-of-load detection, over- and undervoltage protection, overcurrent protection, reverse polarity, and thermal protection. An integrated output clamp makes the addition of an external flyback diode unnecessary. A sense pin allows for monitoring the load current and detecting faults.

www.infineon.com/cms/en/product/evaluation-boards/shield_bts50010-1tad/



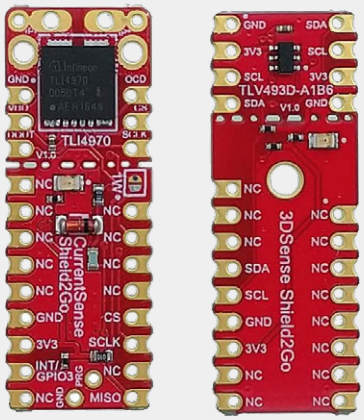
Shield2Go TLV493D and TLI4970

The TLV493D is a 3D magnetic sensor that measures linear motion and rotation in three dimensions. Applications include joysticks and other control elements, anti-tampering and any other application that requires accurate angular measurements. It also integrates a temperature sensor.

www.infineon.com/cms/en/product/evaluation-boards/s2go_3d-sense_tlv493d/

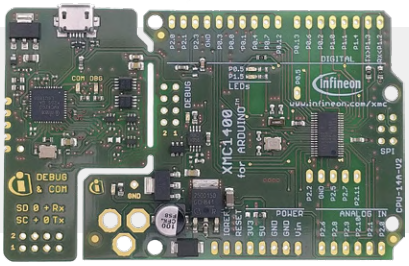
The TLI4970 is a coreless magnetic AC/DC current sensor capable of measuring currents up to ±50 A with a resolution of 12.5 mA (13 bits) and an SPI interface. Hand-soldering this device is difficult, making this Shield2Go a handy breakout board. Although the sensor may seem like a pretty simple device, it isn't as it integrates a DSP with ADC and digital filter, overcurrent detection, an EEPROM, a temperature sensor, and a mechanical stress sensor.

www.infineon.com/cms/en/product/evaluation-boards/s2go_cur-sense_tli4970/



XMC for Arduino

Arduino-compatible development boards exist for XMC1000 and XMC4000 devices. As examples we can mention the 'XMC1400 for Arduino' and the very similar 'Boot Kit XMC1100'. These boards are supported by the Infineon Boards Package for Arduino so the user can program them through the Arduino IDE.



The Infineon boards presented in this article are available in the Elektor Store!

ges up to 28 V. Internally they integrate an N-FET with all sorts of protection circuits and a charge pump, allowing a microcontroller to control them directly. Infineon's low-side switches use "HITFET" technology (Highly Integrated Temperature protected FET). Their main playground is automotive applications where they control lights, heaters, relays and small motors.

Evaluate sensors with Shield2Go

Infineon is active on multiple fast prototyping fronts. Shield2Go is their own approach of such a platform. It consists of a family of breadboard-friendly tiny breakout boards (39 × 15 mm) with a standardized pinout. Each board is equipped with a different sensor or a security IC; there is also a board with an XMC1100 microcontroller ('XMC 2Go'). The "chip-of-interest" is placed on one end of the board that can be broken off. Thus, the user can first try out the IC on a breadboard and, when ready, break off the chip to build it into a device without consuming too much space. Instead of connectors, the boards are furnished with clamping pin headers to use them without requiring any soldering. The 'My IoT Adapter' board bridges the gap to Arduino-compatible platforms like Infineon's own Boot Kits. With this adapter up to three Shield2Go boards can connect to a baseboard. Arduino libraries for the Shield2Go boards are available at GitHub. ⏮

190352-01

Web Links

[1] Infineon for makers: www.infineon.com/makers

[2] Infineon at GitHub: https://github.com/Infineon

Let There Be (Good) Light!

Quality criteria for room lighting with LED lamps

By **Uli Zappe** (Germany)





In the past, everything was better — or at least simpler: if you wanted to buy an incandescent or halogen bulb, all you had to do was think about its performance, dimensions and beam angle; the properties of the light emitted were mostly fixed by the lighting principle. Even brightness was so strictly coupled to the power consumption that consumers went by watts rather than by lumens. LEDs on the other hand can be tailored for a wide variety of properties. But what are the criteria for good LED light?

If you want to find quality criteria for LED lighting, you're not spared some colour physics. And there, unfortunately and surprisingly, you get to deal with quite a mess. The reason for this is simple: colour as an object does not exist at all in the physical world — only our brain turns light waves into colour. The same applies to sound waves and tone heights, but the conditions are wonderfully simple: double the frequency, the same tone is an octave higher. Within colour physics we can only dream of such simple connections.

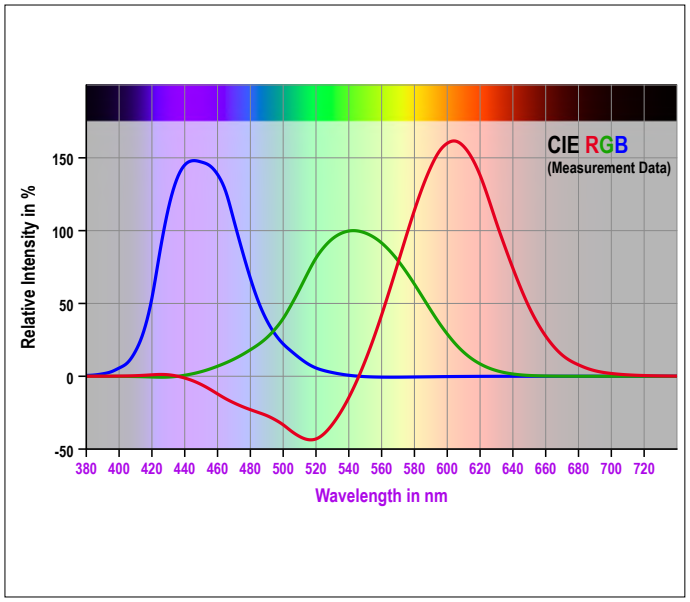


Figure 1: Result of the historical colour mixing experiments: Starting point for the Standard Observer of the 1931 CIE.

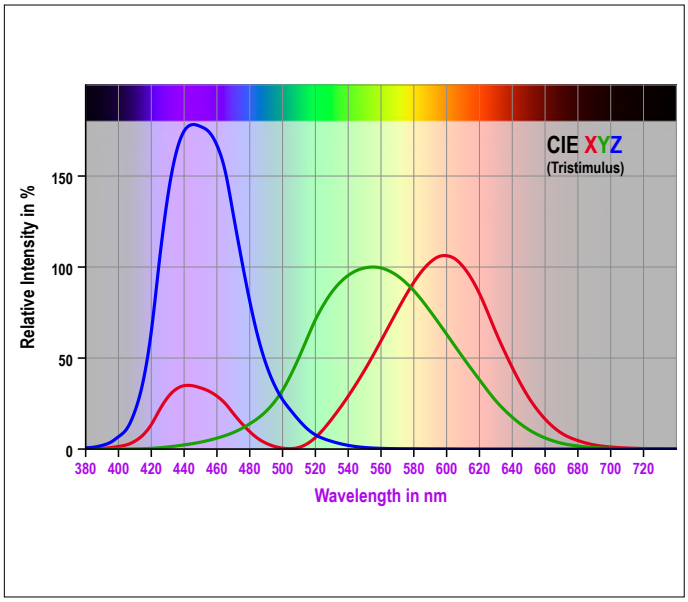


Figure 2. Tristimulus: transformed version of the Standard Observer's colour mixtures.

The Standard Observer

So how can we physically study something that only exists in the human psyche? The only way to do this is through psychological tests that put physical quantities and human colour perception in quantitative relation to each other. Such tests were carried out around 1930 on a measly 17 (!) test subjects and their results were averaged to the so-called standard observer of 1931, which the International Commission on Illumination CIE (*Commission Internationale de l'Éclairage*, [1]) declared the basis of colour physics. To this day, those 17 persons represent the colour vision of mankind physically more or less successfully!

Since it was already known at that time that any colour impressions could be created in humans by mixing three 'basic colours', the 17 test persons got a control panel pressed into their hands with which they could control monochrome light sources with wavelengths from 700 nm (red), 546 nm (green) and 436 nm (blue) ranging in brightness from 0% to 100%. Their task now was to reproduce, by appropriate light mixing, the light of several other monochrome light sources so well they were unable to perceive any difference. The statistically averaged result published by the CIE in 1931 looked the same as in **Figure 1**. By convention, the green control curve is scaled to 100%.

For obvious reasons, the three measurement curves were designated R, G and B (not to be confused with the RGB colour spaces commonly used for computers and monitors). For example, to simulate a monochrome 600-nm orange light source, 160% of R, 29% of G, and 0% of B were required.

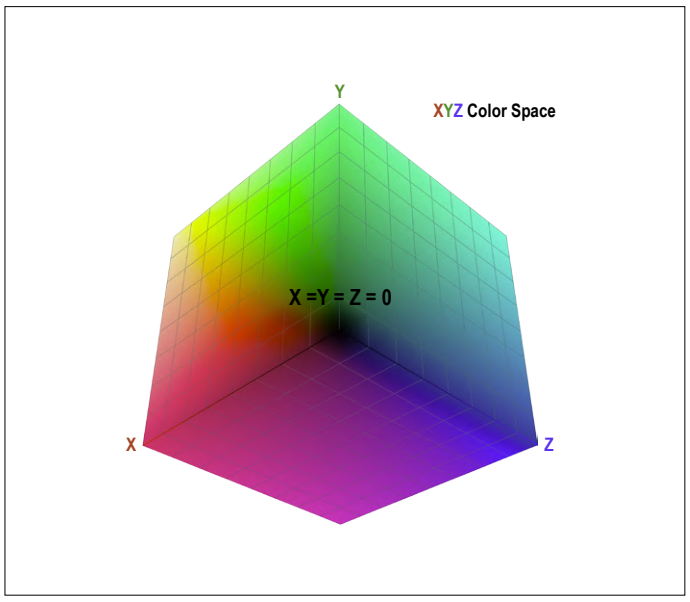


Figure 3: The XYZ colour dimension.

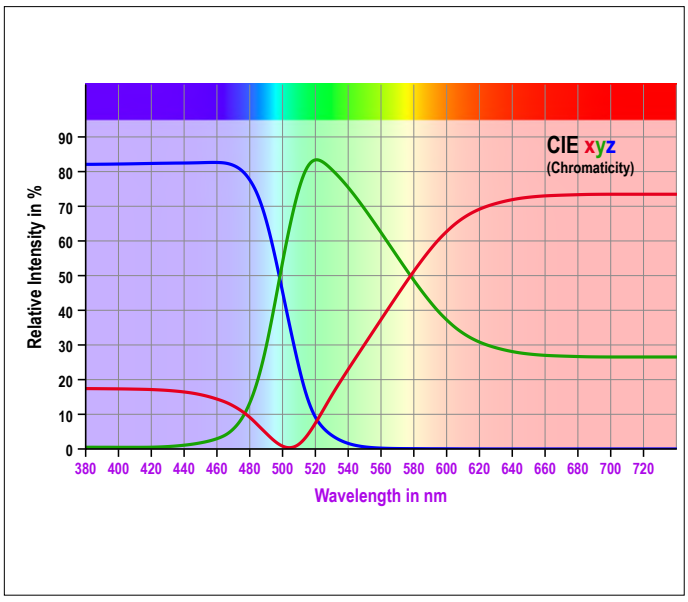


Figure 4: Chrominance curves. The sum of all three colour components is 1 for each wavelength.

Interestingly, it turned out that colours in the wavelength range 437–546 nm could not be composed perfectly from the mixed colours. This was only possible when the colours to be simulated were mixed with some of the red colour with 700 nm. In the measurement curve, this was expressed by negative values for R; the same amount of R was subtracted on both sides of the equation, so to speak. But what's an intensity supposed to be <0%? You're not the only one raisin your eyebrows here.
To solve the problem, a mathematical (linear) transformation was released on the measurement results, which shifted all values to positive but preserved the mixing ratios as information content of the measurements. While at it the researchers chose the transformation in such a way that the G curve at the same time correctly represents the brightness sensation of the human eye — very practical! But that was the end of the scientists' imagination, and they couldn't come up with new names that would make sense for the new curves: R, G and B had already been assigned. So, the generic variable names X, Y and Z (see **Figure 2**) used in mathematics were chosen without further ado.
Because these three measurement curves indicate how our viewing senses are stimulated to a colour impression by light waves, these XYZ curves are also called *tristimulus* curves. Many decades later, shortly before the turn of the millennium, modern technology was used to measure the spectral absorption behaviour of the three different types of cones responsible for human colour vision on the retina of the eye. It was found that these absorption curves were very closely related to the XYZ curves. A perfect fit.

Find colours

Physical quantities with three components can always be represented in a three-dimensional coordinate system; therefore, the quantification of colours is also referred to as colour space. You could sketch the XYZ colour space as in **Figure 3**. At first sight it looks pretty, but it's not very practical for finding colours since most are invisibly hidden inside this cube!
If it is only a matter of the colour chrominance, that is, whether we are dealing with turquoise or orange or violet, but not with the relative brightness or saturation, a trick is available: divide X, Y and Z of a colour by the sum of X + Y + Z. Thus, the relation between X, Y and Z remains unchanged but the sum of X, Y and Z is always 1 or 100%. And exactly this relation is the chrominance (ancient Greek *chroma* = colour). Here we ignore which specific values for brightness and saturation are involved so that the sum of the colour components results in 1 invariably.
When considering al this at that time, scientists were probably so exhausted by this mental feat (again), they could no longer think of any particularly original letters for the chrominance values. They were simply baptized x, y and z. **Figure 4** shows the resulting chrominance curves.
But what did we gain from all this? Since we know that the sum of x, y and z is always 1, we can be satisfied with a two-dimensional representation of x and y, because z always results from this as 1 – x – y (at the zero point of x and y, z is logically 1). This results in the chrominance diagram from **Figure 5**.
The greyed out half above the diagonal does not contain any meaningful values since there x + y > 1 would exist, which by definition was excluded. So stay half way down on the left, and you'll find — hooray! — all perceptible colours, even if at

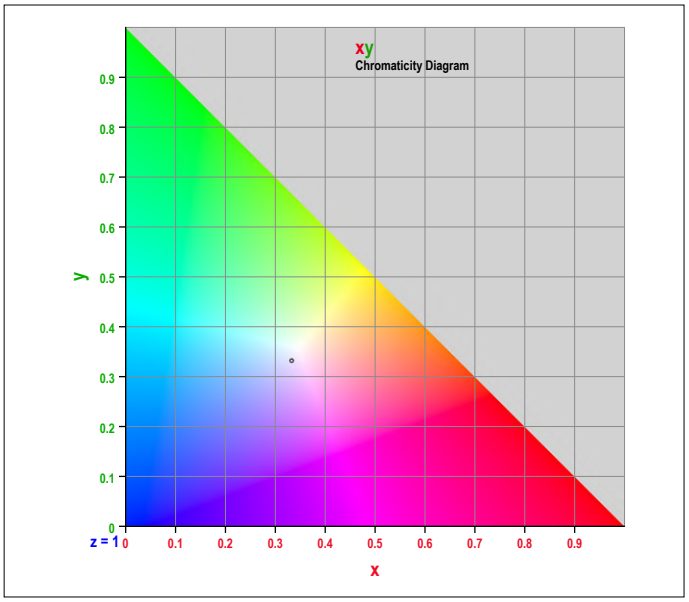


Figure 5: The chrominance diagram. z is equal to 1 – x – y.

certain saturation and brightness values only. More precisely, all perceptible colours.
Let's not rejoice to soon! In the chrominance diagram, not only all perceptible colours are 'found' now, but even some that do not exist. Note that the colours shown are only symbolic, because ElektorLabs magazine cannot print 'non-existent colours' right now. By now eyebrows will be raised, but the reason is: R, G and B were real existing monochrome colours used in the experiments to determine the standard observer. By the linear transformation to avoid negative curve values they got shifted to X, Y and Z, and these 'shifted colours' do not exist in physical reality— they are imaginary, purely mathematical entities.
The next question already arises: which colours in the beautiful

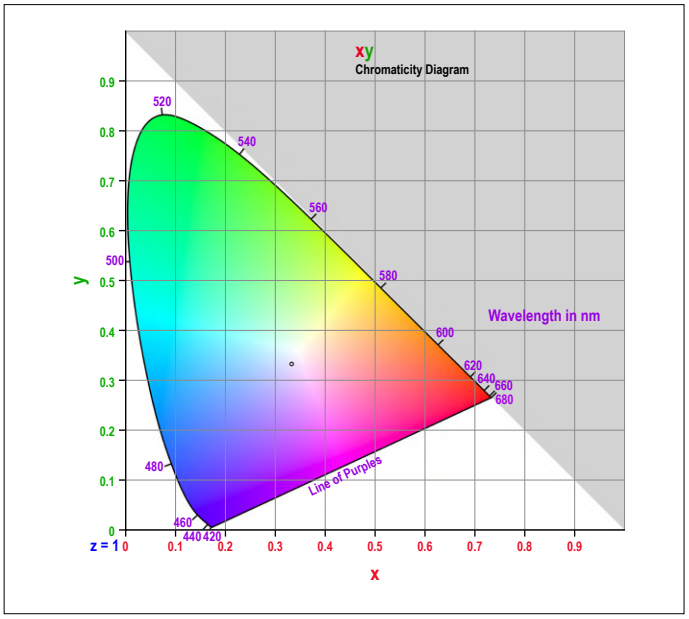


Figure 6: Chrominance diagram of actually existing colours.

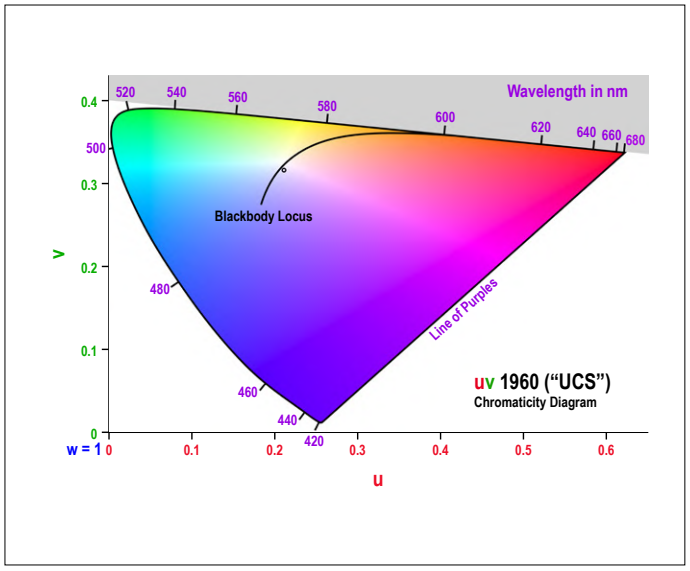


Figure 7: uv-1960 chrominance diagram. (Uniform Color Space)

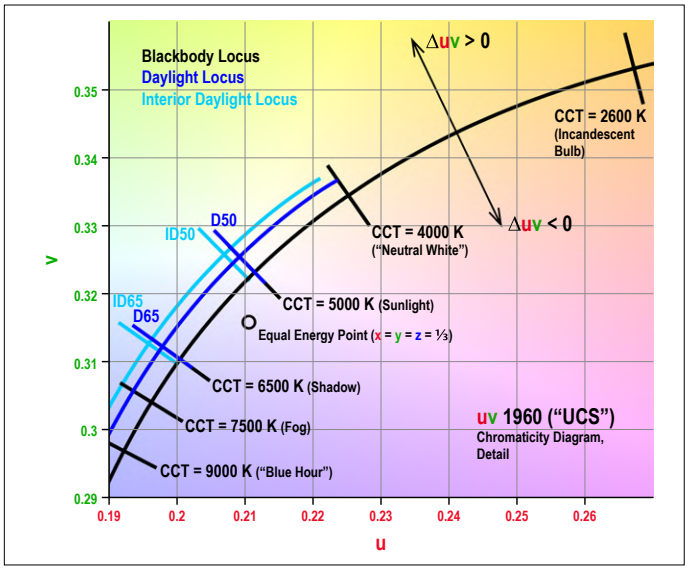


Figure 8: Correlated colour temperature. Excerpt from Figure 7 with examples of different correlated colour temperatures.

chrominance diagram of Figure 5 actually exist? Quite simple: for all monochrome colours, the *tristimulus* curves are used to determine which XYZ values correspond to the monochrome wavelengths, convert them into xy chrominance values, enter them into the chrominance diagram and — importantly and illuminating at the same time — connect all these points with each other. All existing colours then lie within the geometric shape drawn in this way, because either a certain colour is monochrome or it consists of a mixture of several monochrome colours and then logically must lie in this two-dimensional space between them, but not outside. This results in the chrominance diagram of real colours in **Figure 6**.

On the straight connecting line between the shortest and longest wavelength, which closes the geometric form, no further monochrome colours can occur since together they form the curved part of the geometric form. All colours at this straight connecting line thus exist — unlike the colours at the rest of the outline of the form — exclusively as a mixture and never as monochrome colours. Since the mixed colours along the connecting line are mainly blue and red and mostly purple, this connecting line is also called purple line.

The small point in the middle of the chrominance diagram is the so-called Equal Energy Point, i.e. the point of equal energy where x, y and z are the same size and therefore all have the value ⅓. If you look closely, you will notice that this point is not your expected neutral white, but slightly shifted into violet. What we perceive as white is therefore different from a colour that is neutral from a physical point of view. This is a good example of the contrast mentioned at the beginning between colour physics and acoustics, where everything is more harmonious.

Colours in uniform

Almost everything you need to evaluate light sources is to hand. In relation to the xy-chrominance diagram of Figure 6 the American physicist David Lewis MacAdam specifically criticized that equal distances by no means imply equal colour differences: while in the area of the centre of the image (x = y = 0.5) orange quickly turns yellow and then green. On the same stretch in the upper, green part of the diagram hardly anything changes; the hue remains iron green.

Therefore MacAdam again released a mathematical (linear) transformation on the chrominance data, which in this case can be illustrated conveniently by looking diagonally from the side on a xy chrominance diagram that looks accordingly distorted, as if the same distances now make up approximately the same colour changes. The largely 'uniform' (equidistant) UCS (Uniform Colour Space) was finished.

But since X, Y and Z as well as x, y and z were already used up, MacAdam in an eruption of fantasy named the coordinates of his new colour space U, V and W. The corresponding chrominance diagram is therefore the uv diagram (... and has nothing to do with UV light). The CIE adopted this colour space as its official recommendation in 1960 but sought further improvements in uniformity. This effort led to a updated 'colour space' in 1976. In order to make the chaos perfect, the CIE then renamed the coordinates to U, V and W. If you want to refer to the colour space of MacAdam, the additional specification '1960' is absolutely necessary. **Figure 7** shows the resulting chrominance diagram: uv-1960.

Due to the development of further, even more 'equidistant' colour spaces just mentioned, UCS 1960 is sometimes considered obsolete today. However, this is not the case for lighting technology, as many of the relevant measured variables are defined in UCS 1960.

MacAdam himself used his new colour space, among other things, to specify tolerances for colour deviations from light sources. The smaller the permitted deviation of a production batch of LEDs, the smaller the circle around the nominal uv colour dot. However, the near perfect circles in the uv-1960 chrominance diagram transform into ellipses in all other, perspectival different colour spaces. Therefore, today people generally speak of MacAdam ellipses [2] when it comes to the scattering tolerance of the light colour of light sources. The standardized MacAdam ellipses, also known as SDCM (Standard Deviation of Colour Matching), are available in graduated

tolerance classes from 1 (very narrow) to 7 and up.
With UCS 1960 understood and explained everything is finally ready to name the parameters for good light.

Black bodies

Apart from the glow worm principle, light in nature is practically always due to matter that glows. The exact colour of the emitted light depends on the temperature of the body. The higher the temperature, the bluer the light. It therefore makes sense to specify the light colour as the colour temperature in the unit K (Kelvin). For the sake of clarity, assume an idealized, black body that does not reflect or otherwise emit any light at all, but emits light exclusively due to its glow. For such a black body, the light colour is clearly determined by its temperature. The temperature-related possible light colours are all available at so called 'black bodies', which is also shown in the uv-1960 chrominance diagram of Figure 7.
The black-body curve makes it plausible why the colour temperature seems to require only a single numerical value to indicate the light colour, whereas the chrominance diagram requires two numerical values for this purpose: the implicit second value results from the condition that the chromaticity coordinates must lie on the black-body curve. Unfortunately, most real light sources deviate from this ideal condition. Therefore, the concept of correlated colour temperature has been introduced: All colours that lie on a line perpendicular to the black-body curve (in the uv-1960 chrominance diagram) have the same correlated colour temperature: the one at which this line intersects the black-body curve. You can take a closer look at this by zooming into the uv-1960 chrominance diagram around the Equal Energy Point. The resulting detail with some CCTs (Correlated Colour Temperatures) can be seen in **Figure 8**.
Since the different colours on these lines all have the same correlated colour temperature, for exact colour determination this value requires supplementing a second parameter in the form of the distance from the black-body curve in the uv-1960 chrominance diagram. This distance is called Δuv (if you can't find Δ on the keyboard, you can also write *Duv* ;-)
If this second value is missing, a correlated colour temperature is not unique.
Although the sun is a black body, the filtering of sunlight by the Earth's atmosphere ensures that the various possible daylight colours are located on a daylight curve adjacent to the black body curve. Correlated colour temperatures for daylight are often written as 'Dnn'. For example, D50 (Daylight 50) corresponds to daylight with the correlated colour temperature of 5000 K. Daylight indoors is shifted a little further by the filter effect of the window glass: ID50 is thus next to D50, since it is not interior daylight with the correlated colour temperature of 5000 K but interior daylight resulting from daylight with 5000 K.
It is obvious that our facial sense is optimally adapted to daylight through evolution. It can therefore very well compensate for colour shifts along the daylight curve towards blue or yellow; white paper thus appears white to us during the day under all lighting conditions. This adaptation process is called chromatic adaptation. Deviations to violet or greenish, on the other hand, do not occur in daylight. Therefore, chromatic adaptation fails here; people quickly perceive such light as artificial and unpleasant.
This is where it gets serious for lighting practice: the Δuv value of a light source should therefore not be too large. According to the CIE, it should not exceed ±0.0054. The length of the CCT lines in the uv-1960 chrominance diagram corresponds exactly to this value. Ideally, the light colour should be as close as possible to the daylight curve and better between the interior daylight curve and the black-body curve. Δuv should therefore never be under 0.

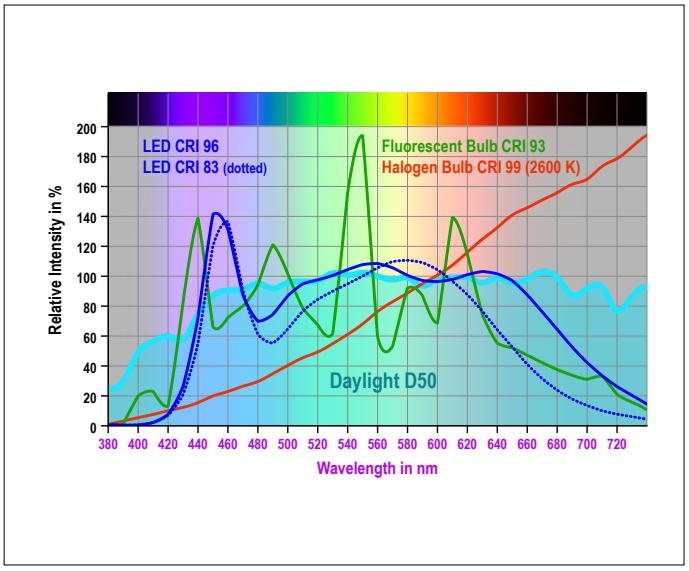


Figure 9: The measured spectral response of four different light sources.

Advertisement



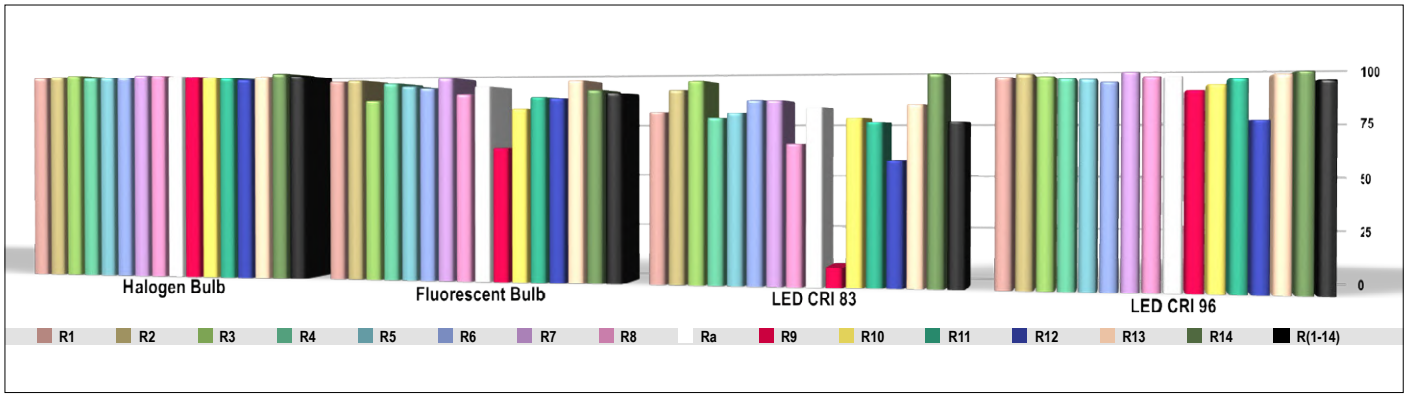
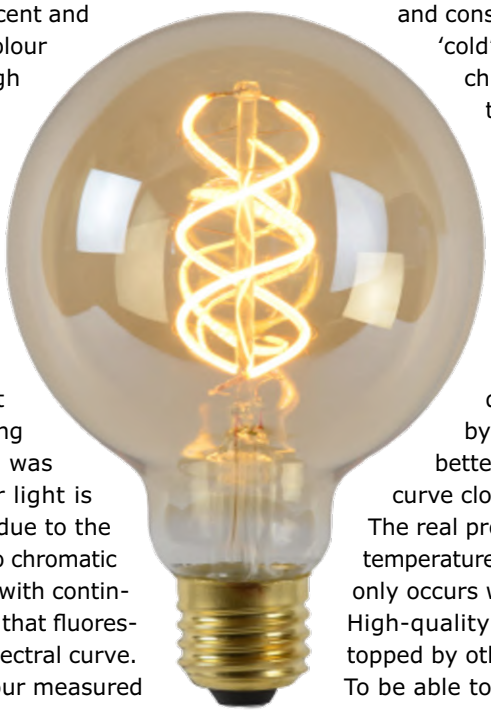


Figure 10: Measurements of the colour rendering index (CRI) for the four lamps of Figure 9.

Of steam locomotives and light bulbs

People sometimes have the strange habit of transfiguring obsolete and technologically and ecologically monstrous technologies as memories of the 'good old days'. That is why soot-spinning steam locomotives with a miserable efficiency (maximum 17.6 %) are considered 'romantic'. The same thing happens to the light bulb. Since higher temperatures of the filaments are not technically controllable, incandescent and halogen bulbs inevitably get stuck at a colour temperature of around 2600 K, although this is miles away from daylight and therefore unfavourable for the human eye. Today, with LEDs capable of reaching 5000 K without any problems, this lack is suddenly considered 'comfortable' and LEDs are provided with yellow tint without need.



The prejudice that 5000 K is "cold" as the light colour of the 'golden' afternoon sunlight is based on a mixup. The first technical possibility to produce illuminating substances with this colour temperature was offered by fluorescent lamps, and their light is indeed unpleasant. However, this is not due to the supposedly 'cold' light colour (which due to chromatic adaptation you would not notice anyway with continuous lighting with 5000 K) but to the fact that fluorescent lamps have a completely fissured spectral curve. **Figure 9** shows the spectral curves of four measured light sources (all have around 5000 K; except for the halogen bulb) and, as a reference, D50 daylight as the measure of all things. The latter has a largely balanced course over the entire spectral range. The fluorescent light bulb aka 'energy saving lamp' is a particularly high-quality copy at an outrageous €45, 'bred' for good colour rendition. Nevertheless, their spectrum is a jagged mountain ridge, which occurs nowhere in nature and consequently has an unnatural, repellent and 'cold' effect on the human eye. Even the best chromatic adaptation is powerless against this deviation from daylight. Although the halogen bulb has a completely smooth spectral curve, it shows its true face as an eco monster: most of the energy is wasted in the infrared range; it heats more than it glows. Due to their design, the two LEDs have an elevation in the blue range, because 'white' LEDs are actually blue LEDs whose blue light is distributed over the entire colour spectrum by a phosphor layer. However, at least the better of the two has a quite balanced spectral curve close to that of daylight.

The real problem with light sources with high colour temperatures is in fact a fissured spectral curve, which only occurs with fluorescent lamps and inferior LEDs. High-quality LEDs on the other hand can hardly be topped by other light sources.

To be able to indicate the negative influence on light

Table 1: Values and criteria for good light

	CRI R_a	CRI R_9	Δuv	GAI
Values for good light	≥ 90	≥ 80	≥ 0 ≤ 0.0054	≥ 80 ≤ 100
halogen bulb	99	99	0.0007	**44**
fluorescent bulb	93	**64**	**0.0080**	88
LED CRI 83	**83**	**9**	0.0036	**78**
LED CRI 96	96	90	0.0024	88

quality, the so-called colour rendering index CRI (Colour Rendering Index) was introduced. It is determined by comparing the light reflected from 14 defined rendering test colours (R_1–R_{14}) of the light sources under test with the reflection that results from illumination with daylight of the same correlated colour temperature. At colour temperatures under 5000 K the light of a black body is used as reference instead. A value of 100 means equal to daylight, i.e. best lighting quality. With increasing deviation, the CRI becomes smaller and smaller. In principle, it can even become negative, because it is not a percentage. **Figure 10** demonstrates how the 14 test colours 'appear' when illuminated with the four light sources from Figure 9.

The white bars in Figure 10 indicate the arithmetic average R_a, which corresponds to the CRI. The 'a' should imply 'all', but oddly enough R_a is only the average over R_1–R_8, all rather pastel colours. The saturated colours R_9–R_{14} and especially the saturated red R_9, which is particularly difficult for LEDs, are completely ignored. Therefore R_1–R_{14} (the black bars in Figure 10) should be used instead of R_a for the evaluation. At least you should consider R_9 in addition to R_a.

The light reflected by R_1–R_8 can be evaluated in a second way as another quality feature of light. The uv coordinates of the reflected light colours are entered into the uv-1960 chrominance diagram, the eight points are connected, and an octagon is obtained. The larger its surface area, the easier it is for the eye to distinguish between the finest gradations of the colours, because the light, so to speak, resolves them better (the pastel colours R_1–R_8 are predestined for precisely this test). The larger the area, the more saturated and 'luminous' the colours appear. This area is compared here with that resulting from illumination by a theoretical, idealized light source with a straight horizontal spectral line. The result is the GAI (Gamut Area Index). A GAI of 100 corresponds to the area illuminated by the idealized light source.

Real values

After all the theory, it's time to get real: **Table 1** condenses the precisely derived quality criteria for good light and contains all the measured variables discussed for the four light sources from Figure 9. Also pictured are the minimum values that should be adhered to for good light. When using many sources of illumination in one room, the MacAdam ellipse or SDCM colour dispersion tolerance should be better than Level 3. When buying lamps, you should therefore copy this table, fold it and put it in your wallet.

In summary, Schiller's phrase "Long speech short sense" is as fitting as it is rare: Good light is currently only possible with high-quality LEDs.

Although the halogen bulb has dream values at CRI and Δuv due to its horrendous yellow tinge (CCT from only 2600 K) it fails completely at GAI. The fluorescent bulb fails above all because of its pronounced green cast (ΔUV = 0.008) and the inferior LED fails in every respect basically.

If you think about good room lighting for offices, production halls or even for homes, you have to struggle with one main problem: unfortunately, there is no LED dealer or manufacturer far and wide who can give you all the values for his products that are indispensable for quality assessment.

But despair not with frustration: some information is occasionally found, and titbits are better than nothing. In addition, the publication of a project is being considered in which all these considerations on room lighting have been considered. You can then orient yourself on this...

190159-02

The Author

Uli Zappe first studied Physics and Mathematics and then Philosophy, Sociology and Psychology. For this reason, he likes to deal with phenomena in the fringe areas between technology and perception.

Web Links

[1] CIE: www.cie.co.at

[2] MacAdam ellipse set: https://de.wikipedia.org/wiki/MacAdam-Ellipse



Advertisement



HomeLab Helicopter

Wondrous Things Electronic Spotted From Above

By **Clemens Valens** (Elektor Labs)

FROM IDEA TO PRODUCT PART 2

In the previous instalment, we talked about the (secret) dream many of us have of bringing a cracker product to market. The all-important question remains: is this, and especially what comes with it, what you really want? From now on we will assume that you answered this question with a wholehearted *Yes!* We will also assume that the product is some sort of electronic device for which only an idea exists.

STAY OBJECTIVE, RETAIN USER PERSPECTIVE

A problem many designers face is lack of objectivity, sine they know their "product" inside out. At Elektor Labs we receive many great projects that are unusable simply because the designer was biased. To one designer, pressing S1 for 125 milliseconds followed by rapid triple pushing of S2 while waving a hand in an 8-shaped pattern in front of an infrared sensor to activate a certain mode (a gesture that somehow came into existence during the prototyping phase) is totally acceptable. With his gesturing and pushbutton pressing turned into a habit, the designer is likely to forget mentioning the 'control method' in the User Manual. Such a product is likely to fail commercially. At all times, strive to take the perspective of the potential user.

SEPARATE MUST-HAVE FROM NICE-TO-HAVE

A good product should satisfy certain needs of its users. Your own wishes are a good start but do not assume that everybody thinks like you. Ask around. Look for people you consider potential users and ask what they expect from the product. Do they feel it would be useful for them? If so, how? What should it do exactly, in what way? Compile a list of must-have functions — the shorter, the better. All other functions are nice to have and make a great list for future evolutions of the product.



Create a mock-up; it doesn't have to work.

CREATE A MOCK-UP

Future users will want to have an idea of what your product is going to look like. Start by making a few sketches of the final device. If you can't draw, get someone to do it for you. Of course, your taste is refined and you know exactly what is needed, yet getting some advice from a professional product designer won't hurt. Once you have good drawings, use them to create a 3D model of the product. Cardboard, wood or 3D printing, any material, crafting or technique will work here. Try to get as close as possible to the real thing. It doesn't have to work but it will give you and potential users an idea of how the real product can and will be used, and what can fit inside. It may also reveal things you hadn't thought about yet.

HOW MUCH CAN IT COST?

Besides looks and functionality, the price people are willing to pay for your product is another important factor for determining what's going to be possible and what not. The cost of your clever circuit inside the product only is a fraction of the end user price. Starting at the retail price and working back, you can estimate what the electronics part may cost. Subtract 30% from the retail price and spread the remainder over design, production, testing, packaging, shipping and handling, aftersales, and coffee. A budget of 10% of the retail price for the board costs probably is on the optimistic side. Mass production now shows promise since producing 10,000 units surely will be more advantageous than a batch of 100. Questions like "Who is going to sell them?", "Who are the buyers?" and "Who is financing all this?" were already answered in the business plan that you wrote. But where are you going to stock this? Is your bedroom or dorm big enough?



Cost versus features; a delicate balance.

WHAT'S NEXT?

Now that you know what your circuit may cost, what it should do and what it will look like, you can finally start working on its design. In case you already started, these new insights will require a critical review of every step you already went along the way. Maybe ditching the "future extension" connector is enough, or the handy but entirely optional status LED. Maybe a redesign or switching to another operating principle will do the trick. Or maybe it's just not feasible... More in the next instalment.

The Most Important Technological Development in Human History



OpenAI and Microsoft team up to extend Azure cloud computing services to large-scale A(G)I systems.

Artificial Intelligence is where all the money goes these days. Founded at the end of 2015, OpenAI is committed to advancing "*digital intelligence in the way that is most likely to benefit humanity as a whole*". The company started out with a capital of $1 billion, brought together by investors as Amazon, Elon Musk and his (pay) pal Peter Thiel.
Last July, Microsoft decided to add yet another cool billion to the capital. They teamed up with OpenAI to further extend their cloud computing services Azure to large-scale AI systems.

"The companies will focus on building a computational platform in Azure of unprecedented scale, which will train and run increasingly advanced AI models, include hardware technologies that build on Microsoft's supercomputing technology, and adhere to the two companies' shared principles on ethics and trust."

On the road to AGI

Current AI systems require training and handle only specific problems. However, using artificial intelligence for solving difficult real-world problems requires generalization and combining multiple technologies. What we need is Artificial General Intelligence, or AGI, collaborating "*with people to help solve currently intractable multidisciplinary problems, including global challenges such as climate change, more personalized healthcare and education*".
The CEO of OpenAI, Sam Altman said (in all modesty): *"The creation of AGI will be the most important technological development in human history, with the potential to shape the trajectory of humanity*".

Hummm... do we really want such an important thing to be in the hands of people who, up to now, mainly have been interested in what was good for themselves and making as much money as possible?

Must-have Homelab Tool



Tired of needing several serial-to-USB converters to adapt to 3.3 V and 5 V boards? Don't want to damage your computer's USB Port by a short circuit on the target board? Then this galvanically isolated USB to UART-TTL adapter is what you need. The µArt runs from 1.8 to 5.4 V, exposes RXD, TXD, DTR, RTS, and CTS signals (plus a few GPIO pins), and supports baud rates up to 3 Mbaud. Reverse polarity and over-current protections included.

www.elektor.com/the-muart-kit

The Keely Motor



The last known photo of the Keely motor taken by Thomas Burton Kinraide, a good friend of Keely and inventor of the Kinraide coil (similar to a Tesla coil but said to produce higher currents).

The article about the Rife Machine (*Sympathetic Resonance — "it's a kind of magic!"*, Elektor Magazine 3, 2019) ended with the mention of the Keely motor. I kind of promised to write about it if I received enough sympathetic responses. A few arrived indeed and here we go.

Now I was convinced to possess a book about John Keely, but I haven't been able to locate it (my spouse sorted our books by size; by colour was not practical, she felt). Luckily, we have Wikipedia.

John Ernst Worrell Keely, born in 1837, was an inventor who claimed to have discovered a new force generated from air and cold water and based on "vibratory sympathy". In layman terms, Keely explained it as follows: "*I take water and air, two mediums of different specific gravity, and produces from them by generation an effect under vibrations that liberates from the air and water an inter atomic ether.*" (Of course, it does.) He then spent the rest of his life convincing gullible — preferable rich — people and governments to give him money ("invest") so he could build a motor powered by this force. However, he never managed to give a convincing demonstration of the force, and sceptics suspected the involvement of compressed air. After his death in 1898, his workshop was dismantled and found to be full of hidden high-pressure tubes and other dubious contraptions that do not belong in walls, floors and ceilings.

Some consider Keely to be at the origin of the field of Sympathetic Vibratory Physics... For more information, please consult the SVP Wiki.

www.svpwiki.com

HOMELAB PROJECT

The JesFS Flash File System

BIG DATA 4 small controllers



Von **Jürgen Wickenhäuser**

Files are key elements to modern electronic devices. Many processors can use operating systems offering interfaces for different types of files, but especially in the embedded area and with low-power 'sub Linux' solutions the finds are rare. It's no surprise some embedded developers tend to include data in the code, which requires a firmware update for subsequent changes. The JesFS embedded file system solves this problem — it is specially designed for slim systems and ultra low-power applications.

In the cult film *Terminator 2 – Judgement Day*, robot T-800 (alias Arnold Schwarzenegger) has the line: "I have detailed files" to explain its (his?) success as a fighting machine. A visionary statement, because it hits the nail on the head: files are definitely among the key elements to all high-tech products. This also applies to embedded systems of all kinds. What was missing so far: a suitable file system. This gap will be closed here!

Files and code

Files are basically containers for grouping and transporting information of all kinds, as they are understood by software from the server down to the smallest microcontroller. In principle and just like in the land of milk and honey, you first have to struggle before you reach your destination. Fortunately, there are operating systems that offer interfaces for different types of files.

Especially in the embedded area and with ultra-low power 'sub Linux' solutions the air quickly becomes thin. It is not unusual for great projects to fail because functions and data are not properly separated. System parameters, for example, can often be installed quite easily somewhere in the code, but if changes are made later, a new firmware must be installed each time. However, this is often only possible directly on the device and on-site.

The author is professionally involved in scientific measurement and related instruments in the alpine region. A few months ago, he had a key experience: on an older device on a lonely mountain at 3000 m altitude settings had to be changed. Panting, the author stated that it is much easier to take care of such a device via the Internet than to dig holes in the snow at –20 °C (**Figure 1**). During the exhausting digging the Terminator film quote came to his mind and he decided to use a file system in the future.

There are already some embedded file systems, but all were sub optimal for its purposes. Author Jürgen Wickenhäuser therefore decided to develop one himself. The result is 'JesFS' — a kind of acronym for 'Jo's embedded File System'. It is specially designed for tiny systems and ultra low-power applications.

JesFS

The most important design criteria for JesFS were optimal reliability and small footprint. Equally important are data integrity and the option of (remote) firmware updates. Parallel to JesFS, a JesFS bootloader was developed, which can transfer AES-encrypted firmware files directly from the JesFS file system to the controller. The bootloader is completely independent from JesFS and is not discussed here.

Jürgen Wickenhäuser now uses JesFS exclusively for his daily work. It has proven to be an extremely important and reliable tool for him. And the beauty of it: he doesn't keep it to himself. JesFS is freely available under GPL v3 [1].

Reference Implementation(s)

JesFS can be compiled with Standard C without any problems. In the following, its functionality will be demonstrated on the evaluation board CC1310 from TI, which is equipped with a 32-bit ARM Cortex M3 “Radio on Chip” ticking at 868/915 MHz clock rate (**Figure 2**). For the CPUs of the CC13xx/26xx series there are not only very inexpensive evaluation boards called Launchpad [2], but also a very good, free and cloud-based IDE, the CCSTUDIO [3] from TI, which enables source code debugging. Especially worth mentioning: most CC13xx/26xx launchpads are already equipped with a serial Flash memory of 1 MB measuring only 2 x 3 mm, i.e. physically tiny!

In addition, with the free TI-RTOS, a real real-time operating system is available that’ also suitable for industrial use and for which there is copious support from TI.

The source code files for JesFS are not only compatible with it, but also with standard C compilers for Windows or Linux. If required, a suitable driver emulates a Flash memory in PC RAM.

Tiny Flash ICs

Embedded systems usually do not require too much storage space. Often a mere 1–16 MB is adequate. Fortunately, suitable Flash ICs have now become incredibly tiny: A 1-MB Flash chip now only requires a footprint of 1.24 x 1.29 mm (**Figure 3**). JesFS itself occupies very little memory space, allowing it to used perfectly from just 200 bytes of RAM.

The third aspect is economical: many larger file systems use RAM for their cache. This often makes them very fast but complicated. Switching to operating modes such as Sleep and PowerDown is slowed down. JesFS on the other hand was designed from the outset with minimum power consumption in mind and can be woken up and put to sleep in just a few microseconds. The beauty of this is that serial memories in Sleep mode often consume less than 1 µA of power, which is ideal for battery operation.

Exclusive features for real IoT

One of the basic ideas of JesFS is to be able to mirror the files of embedded systems easily and automatically by way of the Internet and onto a server. The server thus always has a copy of the files of the embedded system that is as up to date as possible. It can also handle scenarios in which the system acquires new data every minute, for example, but for energy-saving reasons only rarely (about every hour) compares it with the server via WLAN.

In addition, a server can use special flags, timestamps and CRC32 hashes to retrieve changed files very easily and even transfer new files and firmware updates to the remote system hallowed with JesFS if required. This topic and the special applications that occur result in almost infinite possibilities which cannot even be briefly described here. In any case, all features and necessary scripts have been thoroughly tested by the author and found to be stable.

First of all, here are the basics. Detailed information about the functionality of JesFS can be found in the reference documentation under [1].

Fully intentional: ‘Unclosed Files’

Most file systems only allow read access to files if they are not open for writing at the same time. Another problem: an unexpected reset may cause unclosed files to be incomplete



Figure 1. Luckily the sensor had low-power short range radio (LPRS), otherwise we would never have found it (image: Am Kandersteg, Switzerland; Jürgen Wickenhäuser).

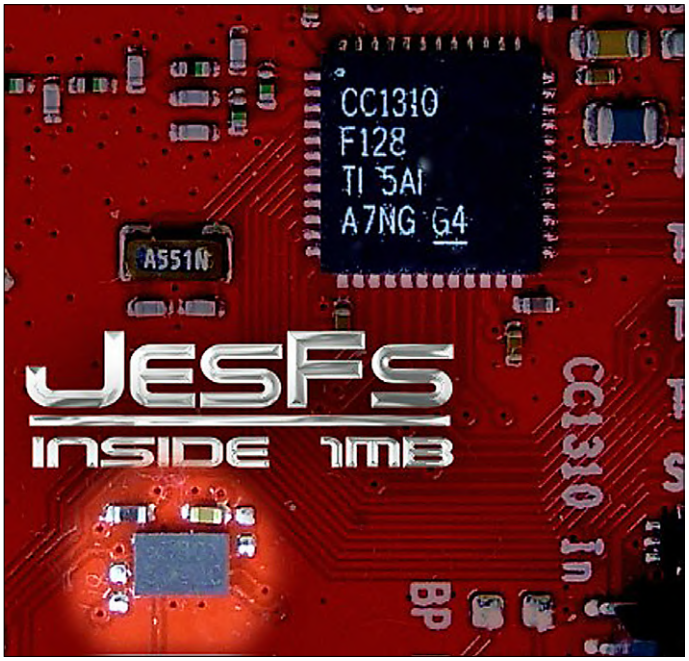


Figure 2. Evaluation board with JesFS (image: Jürgen Wickenhäuser).



Figure 3. WLCSP housings are even smaller (image: Macronix.com).

or, in the worst case, all the data on the storage medium to disappear into Data Nirvana.
Not so with JesFS — even files opened for writing can be read at any time. This is particularly advantageous for log files, for example, where embedded systems regularly store new data, while a server only retrieves the last or current changes at any time. Also, the problem of unclosed files is solved: An unexpected reset does not lead to data loss with JesFS!

A file system for the IoT

Limitations of JesFS

With JesFS all ballast was avoided. However, the access times are not optimized to the absolute minimum. As you will see later, there is also a small limitation with the Unclosed Files.
Since this is a simple file system for small embedded systems with limited resources, the length of the file names was limited to a maximum of 21 characters. For the sake of simplicity and low CPU load, no directories were implemented. So it's like a flat-file system.
For practical use on small systems, however, these restrictions should not entail any relevant restrictions. JesFS is an easily manageable and slim file system that can be used productively. The author himself was his most critical tester here.

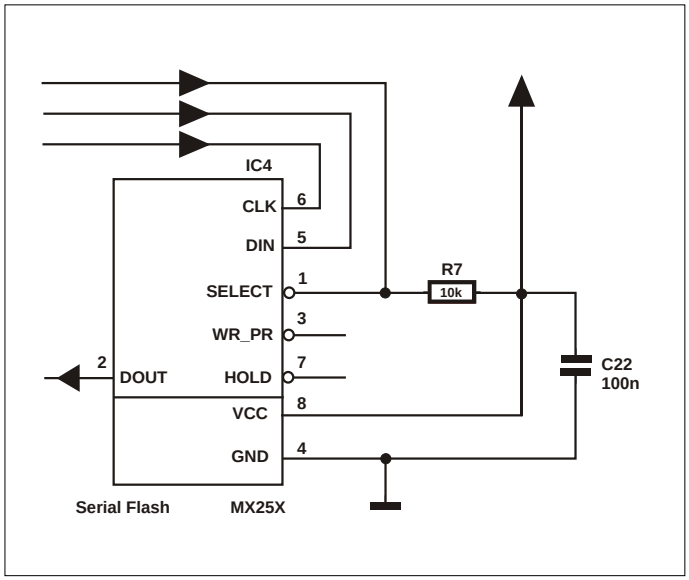


Figure 4. Standard assignment for 8-pin serial Flash chips.

Serial Flash Memory

Before the software details, first let's discuss a few basics about Flash memory. There is now an almost unmanageable variety of different Flash memory types. In the embedded sector, however, serial NOR Flash chips are mainly of interest.
In order to be able to adapt JesFS to your own hardware without problems, the following points have been explained in more detail, because they also concern low-level drivers.
In principle, almost all relevant Flash memories have the same structure. They often differ only in special features such as capacity, quiescent current, maximum clock rate or bits per clock.

Virtually all serial NOR Flash chips have six features in common.

- With an 'empty' memory, all bits are set to '1';
- When writing, bits are only set from '1' to '0';
- Conversely, bits are reset from '0' to '1' during deletion' and only there;
- 1 to 256 bytes can be written per write operation. This takes about 1 ms, consequently, as many bytes as possible have to be written at once.
- Data can only be deleted in blocks or sectors or completely (the chip as a whole). Sectors are usually 4 KB in size. Deleting a sector takes 20–200 ms. This should occur rarely though.
- Full sector clearing is finite (usually more than 100,000 times).

The access to a serial Flash chip is easy and is handled in four steps (see also the circuit of **Figure 4**):

- a few ns after "SELECT" has been moved from "1" to "0", the chip is active;
- writing commands and data via "CLK" (clock) and "DIN" (data);
- read data and status register via "CLK" (clock) and "DOUT" (data/status);
- a release by level change at "SELECT" from "0" to "1" deactivates the chip.

The maximum clock rate of a serial Flash chip is normally quite high (> 50 MHz). Most microcontrollers cannot even use this to full capacity with hardware drivers. The CC1310 controller of my reference platform with its 12 MHz still achieves a good data throughput of 1 MB/s.

Below is a short list of common memory chips for small systems. Many alternatives exist but they often require higher power-down currents:

- MX25R8035F (1 MB) to MX25R1635F (16 MB);
- W25Q80DL (1 MB) to W25Q128JV (16 MB);
- GD25Q16 (2 MB).

BlackBox-Demo – the flight recorder for home use

Many embedded devices work for years on end without problems. But at some point errors start to occur. In amazement we ask: “What went wrong?”
To answer the question, it would be useful to have enough ‘historical’ data to diagnose the fault. For example, did any internal parameters suddenly change, such as temperatures, voltages or humidity? Did anyone adjust anything? Have there been any signs of trouble before? These are exactly the same queries for which airplanes are equipped with the famous *Black Box.*

However, embedded systems often have little memory available for this purpose. with its ‘*Unclosed Files*’ JesFS offers a comfortable solution for a robust and maximum energy saving as well as very simple recording of diagnostic data. JesFS ensures that no ‘data corpses’ occur in the file system even in the event of faults and that further data can be written at any time (even after a reset or power failure, for example).
Two files are used for this purpose:

- the primary file *Data.pri is* written up to the specified *HISTORY* length;
- it is then renamed to the secondary name *Data.sec;*
- any existing secondary file is replaced when the current secondary file is created.

This ensures that a constant, minimum amount *HISTORY* of historical data, but never more than 2 x *HISTORY*. The correct order is obtained by first retrieving *Data.sec* and then *Data.pri.*
The complete and documented source codes for the BlackBox demo for various compilers (PC and Embedded) including the source code files for JesFS can be found under [1]. The listing printed here shows an excerpt from the code:

```
*********************************************
 * log_blackbox(char* logtext, uint16_t len)
 * This funktion logs one line to the the history
 *********************************************
int16_t log_blackbox(char* logtext, uint16_tlen){
   FS_DESC fs_desc, fs_desc_sec; // 2 JesFs file descriptors int16_t res;
   res=fs_start(FS_START_RESTART); // (fast) WAKE JesFs (might be sleeping)
    if(res) return res;
   // Flags (see docu): CREATE File if not exists and open in RAW mode,
    // (RAW needed because in RAW-Mode file is not truncated if existing)
    res=fs_open(&fs_desc,“Data.pri“,SF_OPEN_CREATE|SF_OPEN_RAW);
    if(res)return res;
   // Place (internal) file pointer to the end of the file to allow write
    fs_read(&fs_desc,NULL,0xFFFFFFFF); // (dummy) read as much as possible
   // write the new data (ASCII, from function arguments) to the file
    res=fs_write(&fs_desc,logtext,tlen);
    if(res) return res;
   // Show what was written
    uart_printf(„Pos:%u Log:%s“,fs_desc.file_len, ,logtext);

    // Now make a file shift if more data than defined in HISTORY
    if(fs_desc.file_len>= HISTORY){
      uart_printf(„Shift ’Data.pri’ -> ’Data.sec’\n“);
      // Optionally delete and (but CREATE in any case) backup file
       res=fs_open(&fs_desc_sec,“Data.sec“,SF_OPEN_CREATE);
       if(res)return res;                                }
      // rename (full) data file to secondary file
       res=fs_rename(&fs_desc,&fs_desc_sec);
       if(res) return res;
   }
   fs_deepsleep(); // Set Filesystem to
UltraLowPowerMode
   return 0; // OK
```

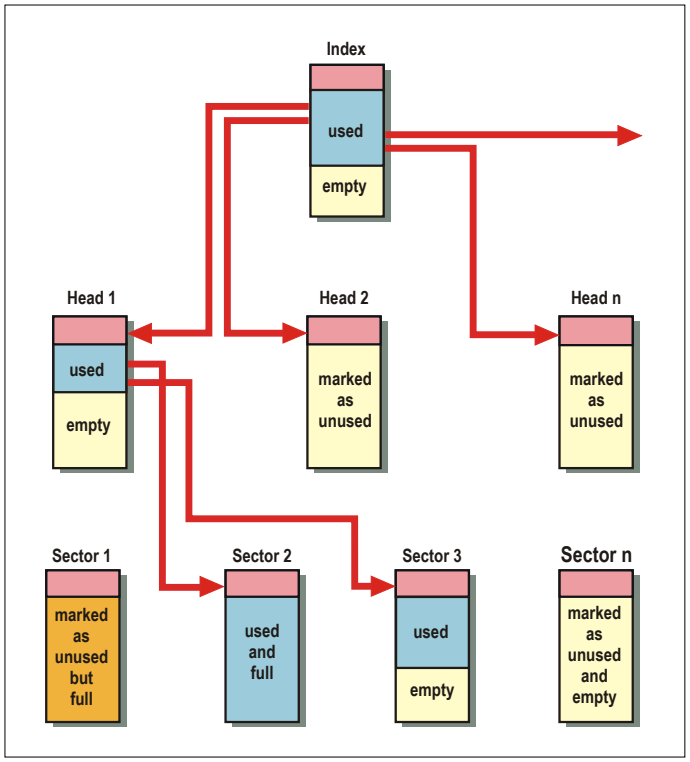


Figure 5. Index is always sector 0 pointing to the file heads.

Basics about the drivers

JesFS uses a high- and a low-level driver. The low-level driver is again divided into general commands for the serial Flash JesFS_ml.c and a lower part for the serial SPI hardware of the controller (here: JesFS_tirtos.c). The high-level driver is described in detail in the reference document [1].
The handling of Flash memory is quite simple. Most Flash chips employ special commands that are not used here. Very few commands are relevant though and fortunately they are identical for almost all available serial Flash ICs. For larger capacities (≥32 MB) the addresses need four instead of only three bytes. The source code file documents the commands more precisely. Here is an overview of the commands in hexadecimal notation:

- **9Fh**: reads a 3-byte ID (in hex) in which manufacturer, type and capacity (as hexadecimal power of two) are encoded. For the MX25R8035 chip installed in the CC1310 this ID is "C22814", where C2h = Macronix, 28h = Type and 14h = 20 = 220 = 1 MB.
- **B9h**: put the Flash chip to deep sleep.
- **ABh**: wakes up the Flash chip; after approx. 5–50 µs it is active.
- **03h**: reading. First the address is expected as a 3-byte value (or 4-byte for capacities from 32 MB), then bytes are delivered until CS# goes High.
- **06h**: must be sent before each write/delete operation.
- **02h**: write. First three (or four) address bytes and then another 1–256 data bytes, with 256-byte limit not to be exceeded.
- **20h**: delete sector. First three (or four) address bytes, then the relevant 4 KB sector is deleted.
- **05h**: tests whether the previous write/delete command has been completed.

Organization in Flash

The details and graphics of the implementation can be found in the documentation and the source code of the JesFS [1]. Only the structure is sketched here. JesFS divides the Flash memory into three different sector types. Each logic sector corresponds exactly to a physical 4 KB sector in the Flash chip. For this reason, a Flash chip with a capacity of 1 MB can store a maximum of 255 files.
Sector 0 has special meaning because it is used as a file index. Each index entry occupies 4 bytes. You can therefore see the index as an array of 1024 "unsigned long integers". For sector 0 the following must be observed:

- the index sector 0 is only deleted during formatting;
- the first three entries contain formatting information: Magic value, Flash ID (see command 9Fh) and the date of formatting (in Unix seconds since 1.1.1970). The value FFFFFFFFh is evaluated as invalid/unformatted;
- the remaining 1,021 entries are the start addresses of the file called heads. All these entries must be divisible by 2^{12} = 4,096 (the sector size). This is also checked by JesFS. Theoretically, a maximum of 1,021 files is possible.

For sectors 1 to *n* there are five possibilities:

1. Sector is empty (contains complete bytes of value FFh).
2. Sector is a *head* to an active file.
3. Sector is a *head* to a deleted file.
4. Sector is part of an active file.
5. Sector is the deleted part of an active file.

A *head* is a sector to which an entry in the index points. It is therefore important to give heads specific treatment since a value once written in the index is never deleted except during formatting. Each sector always starts with exactly three (or twelve for *heads*) values of the 'unsigned long' type:

- magic value of usage (*head*, file or empty);
- owner of this block (a reference to the *head* or, if the block itself is a *head*, the values FFFFFFFFh);
- pointer to the next block of this file or FFFFFFFFh if this was or is the last block of the chain.

Then either the file's data or, if it is a *head*:

- the file length in bytes or FFFFFFFFh, if not known or still open;
- optionally the CRC32 hash of the file (if it was written and closed in CRC32 mode);
- creation date of the file (in Unix seconds since 1.1.1970);
- file name (max. 21 characters, without final character 0h);
- open flags of the file (1 byte);
- 1 reserved empty byte.

A *head* always occupies 48 bytes and a normal data sector 12 bytes with management information at the start. The structure is graphically illustrated in **Figure 5.**
Deleted sectors are marked only. The Magic values used got selected in such a way that the status is actively deleted by additional overwriting with '0' values (which happens very quickly). Only when a sector is really needed and there are no

more free sectors, it is deleted by command 20h which takes a little more time.

Wear Leveling und Unclosed Files

With Flash, it is extremely important that all sectors are used equally often, as they can usually only be deleted about 100,000 times. So you could 'kill' a sector in a loop in a very short time. The avoidance strategy is called *wear leveling*. With JesFS this means that a file (due to the fixed entry in the index) may be deleted a maximum of 100,000 times. This only looks like a lot of reserve at first glance, but especially for embedded systems this limit is certainly not without problems in certain applications. On the other hand, it should be borne in mind that even if each sector is extinguished every hour, safe operation is achieved over eleven years. For most purposes that should loosely suffice.
In order to delete sectors as rarely as possible, it is also a good idea to leave active or used files open; deleted bytes always have the value FFh. If the file in question then never uses this value (e.g. by saving only ASCII data or replacing FFh with an escape sequence), the end of a file can also be determined by simple reading (up to one FFh character).
JesFS is equipped with a very fast routine that even finds the end of a 16-MB file in a few milliseconds.

CRC-32 hash and data integrity

Only very few files change frequently. Therefore JesFS can carry a CRC-32 when writing (and also when reading) a file, which is noted together with the length in the head in a 'file close'.
In this way, the integrity of a file can be checked at any time by simply reading it. The CRC-32 algorithm used is industry standard (ISO 3309 etc.) and is even supported by PHP (a common server script language).
At the start of JesFS (or optionally after each wake-up of the system) a comprehensive test is performed to check the basic integrity of JesFS. JesFS pays very careful attention to bugs. The demo source codes also contain a function that checks the complete JesFS for 'heart and kidneys' (i.e. all sectors and files).

Bootloader and firmware upgrades

Due to the very simple structure of JesFS, even a simple bootloader can read files with little effort and test their integrity. However, a bootloader is only bound to a specific CPU architecture or controller type.

Testing JesFS

The easiest way to test JesFS is on a launchpad of the CC13xx /26xx series. These boards are already available from approx. €30 at various retailers. In the documentation for JesFS on Github you can also find a step-by-step guide for a simple console application.
Alternatively, JesFS can be compiled on a PC running Windows or Linux, forcing the Flash memory to be be emulated in RAM. The generated images are then fully compatible with the real content of a Flash chip. Consequently, they can even be transferred or read back quite easily with a programmer, provided the data format of the microcontroller is also *little-endian*, which is the case with most ARM SoCs such as MSP430, MSP432 etc.

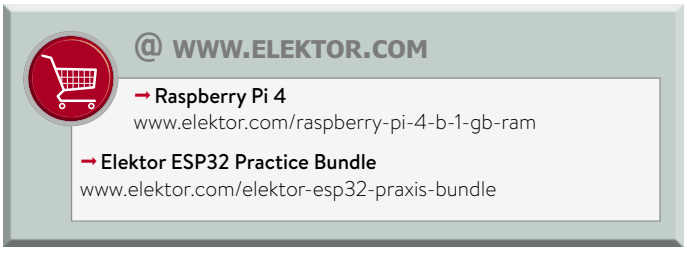
What's the next step?

Depending on reader interest, this report is only the beginning. Sure, a file system is no simple matter, but the author would be very happy if JesFS would develop into a useful general tool, because it has surprising potential.
Of course, JesFS is not only limited to controllers of the CC13xx /26XX series, but in principle suits almost all computing hardware from the 16-bit class upwards. Jürgen Wickenhäuser is currently working on the adaptation of JesFS for the MSP430 family. There are many other projects conceivable in combination with JesFS. Especially the upcoming IoT offers gigantic possibilities. If you are seriously interested, you can contact the author at his e-mail address joembedded@gmail.com

(190154-02)

The Author

Jürgen Wickenhäuser has been enthusiastic about electronics since his youth. After studying Electrical Engineering he turned his hobby into his profession in 1992 and has been working for several years as a developer of scientific environmental measuring instruments in southern Germany.



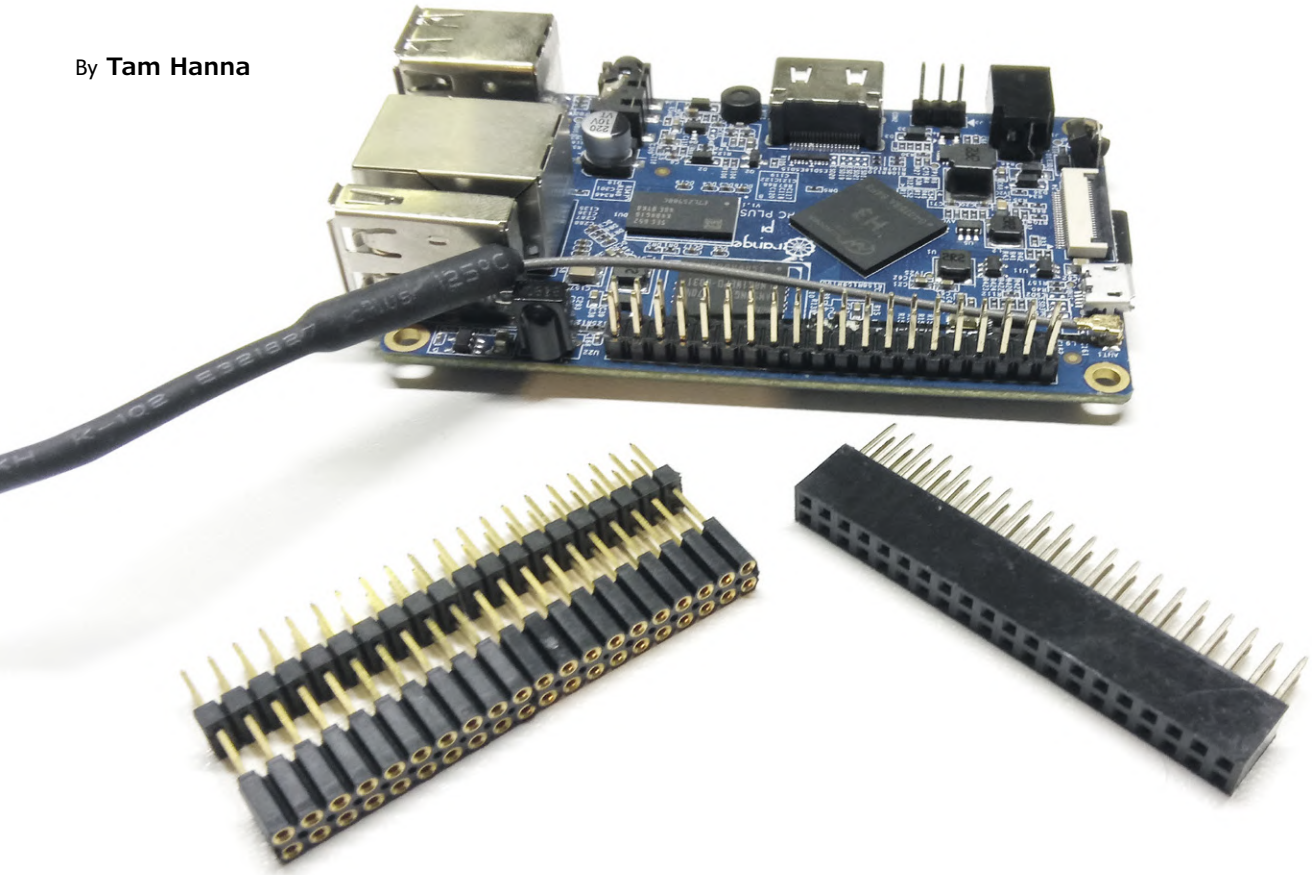
Web Links

[1] JesFS V1.1 auf Github: https://github.com/joembedded/JesFS
[2] TI-Launchpad: www.ti.com/tool/LAUNCHXL-CC1310
[3] CCSTUDIO: www.ti.com/tool/CCSTUDIO
[4] Macronix Flash: www.macronix.com/en-us/products/NOR-Flash/Serial-NOR-Flash

Raspberry Pi for Automation

Maker-System powers end-products

By **Tam Hanna**



Developing a single-board computer for a commercial automation application can be a tedious business. It's often cheaper, faster, and much more convenient (especially if only a small number of systems are required) to use one of the off-the-shelf development platforms popular amongst the Maker community rather than to go ahead and developing you own new SoC. How does the Raspberry Pi and other consumer SBCs perform when they are integrated into an end-product? Where are the pitfalls, what are the advantages and how do you find suitable accessories? The author gives an insight from his own experience designing commercial applications in the automotive and automation sector.

The more complex a system-on-a-chip (SoC) the development board uses, the higher the demands it puts on the developer. This ranges from the difficult layout of the board to the porting of the firmware. Anyone who has ever ported Yocto or even a full-fledged Linux distribution will already be aware of this.
Systems such as Raspberry Pi and clones of it like the Orange Pi from Shenzhen Xunlong have been a popular choice of platforms amongst the maker community for many years now. These process computers are small, relatively inexpensive and work (usually) without any problems. Linux runs - by default - also without problems and even wireless links such as WLAN and Co are now integrated on board.
Here we take a closer look at some of the practical aspects of using a development system in a commercial product as the author develops a system for a US company active in the automotive industry.

Which way to go

As an engineer, it is often tempting to pursue a 'technically innovative' solution just because it is technically innovative but with experience, we come to appriciate the value of the KISS principle. The decision to use or not to use an off-the-shelf process computer is a cost issue, which depends primarily on the predicted volume of sales.
The Raspberry Pi is not particularly price elastic, while the Orange Pi clones are, but due to the rising shipping costs, they work out cheaper when purchased in small or extremely large batches from the Far East. Since the whole thing generally

costs more than the sum of its parts, we have to determine the break-even point to determine how much a process computer costs compared to a completely in-house produced board. The formula for determining this is as follows:

*n * (Cost of Process Computer + Cost of Board) =*
*n * Cost of Board + Dev Cost*

Also note the not inconsiderable advantage of faster time to market when a ready-made board is used. Debugging your own design takes a lot of time due to the various board manufacturing steps, while a Raspberry Pi is ready to go within a few hours.

Which Raspberry Pi tastes best?

The selection of the desired process computer is anything but easy, because there are several versions of the Raspberry Pi available which are physically different. If a selected process computer becomes discontinued, a redesign of the system motherboard is usually unavoidable.

The second criterion is the processor used and/or the desired operating system. If you are new to Linux then an Orange Pi will not be the best choice. I have no problems with Armbian (the Debian and Ubuntu-based computer operating system for ARM development boards), and I am not nervous about recompiling a kernel if necessary. The Raspberry Pi really benefits from the support it enjoys from its community of users at the various online forums; if you are new to the system you can be sure someone out there has already overcome any difficulty you are experiencing and will be only too eager to point you in the right direction to resolve the issue.

There are four models in the Raspberry Pi range of processor boards assuming we ignore any bespoke versions which we will touch on later. At the bottom end of the range are the various Zero models which are mostly produced (along with most of the other UK-produced RPi boards) in the Sony production facility in Pencoed Wales. The RPi Zero measurers 65 x 30 mm and is based on the same Broadcom SoC (using the single-core 32-bit ARMv6Z processor) of the original RPi but clocked at a slightly more respectable 1 GHz.

The Foundation makes it clear that the Zero board is intended for hobbyists and the RPi Compute Module should be chosen for (high volume) commercial embedded applications. There is no guarantee that future incarnations of the Zero will be compatible with existing versions and the RPi foundation will currently guarantee its availability up until 2022 whereas the Compute Module will be available at least until 2026.

The main battleground is the classic Raspberry Pi B, which has been in production since 2012 with few physical and moderate technical changes. The model A+ was released a few years after the original and features user-friendly rounded corners and the now standard 40-way pin header HAT connector even though the board is smaller. The final one to mention, the RPi Compute Module shown in **Figure 1**, is relatively expensive.

The Orange Pi system boards from Shenzhen Xunlong are quite a bit smaller than the original RPi models; the Orange Pi Zero is only 48 x 46 mm but slightly thicker due to the height of the installed connector housings. They can run Android, Lubuntu, Debian and the Raspberry Pi Image but earlier variants did not provide an HDMI video output port.

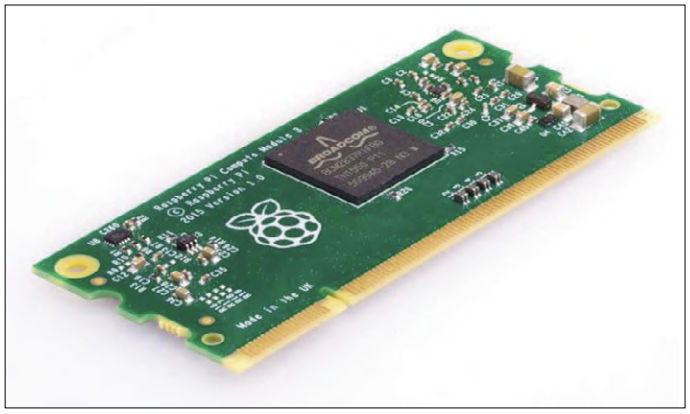


Figure 1. A process computer that looks like a PC memory card.

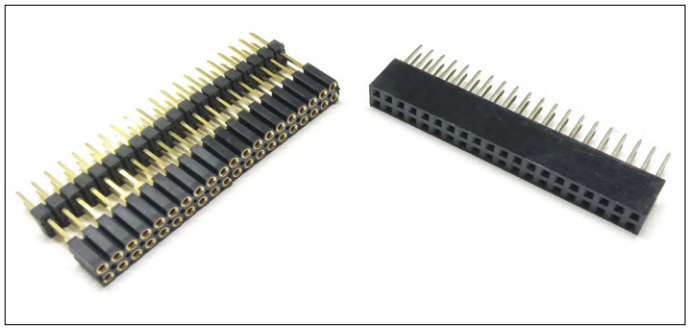


Figure 2. Various spacers provide the physical distance.

Physical compatibility

The author is often asked about the basic design considerations of automotive electronics, just think about issues such as taming the notoriously noisy vehicle DC supply which we covered in the March / April edition of Elektorlabs this year. As important as this subject is to build a reliable system there are also more practical considerations to bear in mind.

To integrate a Raspberry Pi or another process computer into a control system it's important to take care of some basic mechanical problems first. The RPi's 40-way pin header provides access to all the important signals and can be used to plug into a system board which provides all the system peripheral interfaces. The pins of the header strip (just like most of the other brands of development board) project only 10 mm above the board surface. The new Raspberry Pi 4 is, according to a quick empirical investigation by the author, 17.27 mm high at its thickest point (the various Orange Pis, which also have Ethernet and USB sockets, are of similar height).

This means that we cannot just mount an RPi onto another board using a standard height 40-way socket-header strip; the height of the PCB-mounted Ethernet and USB port connectors will prevent the 40-way connectors from mating. It's not advisable to offset the RPi position so that all its high components overhang the other board; that makes the whole assembly cumbersome and means that you only have two mounting holes available to secure the RPi to the system board.

The answer is to use an extended 40-way socket-header strip. There are several variants available and **Figure 2** shows two examples. The stand-off adapter on the right plugs into a standard 2x20-pin socket and adds extra height to the socket

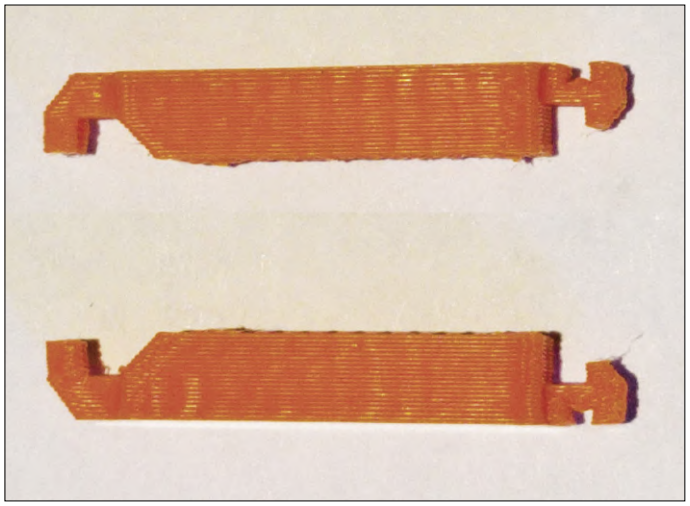


Figure 3. These somewhat crude spacers keep the two boards separated (image source: SpotMyBus).

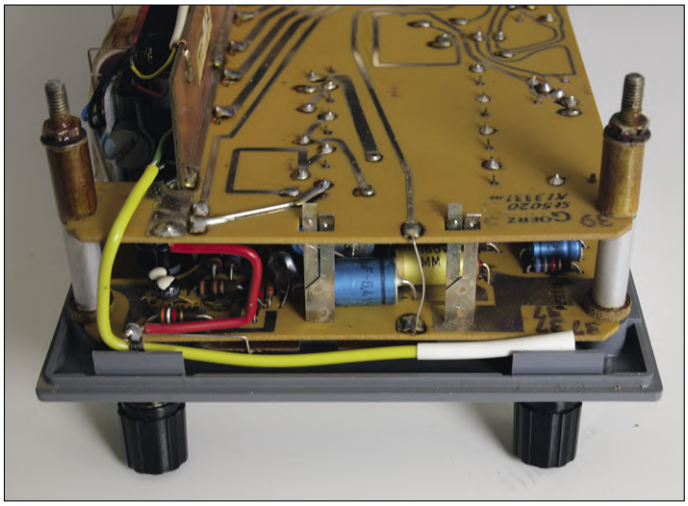


Figure 4. Stacked boards work for the Unigor 6e analogue multimeter.

header strip. The biggest advantage of this approach is that the adapter is widely available at low cost. To its left is a custom made version by Fischer, which can be soldered directly to the board, producing a more robust socket assembly suitable for use in harsh environments. This solution is slightly more expensive and less likely to be stocked by component suppliers, on the other hand, the author has heard of cases where system unreliability has been traced back to the use of the lower cost variant. An extension ribbon cable terminated with 40-way IDC sockets is also a popular solution to the problem. If the extension connector plugs into both boards securely any other mechanical fixing is less important. Use spacers of the appropriate length with the mounting screws to ensure the two boards are held apart the correct distance so that the extension plug is fully engaged (**Figure 3**).

Sometimes it's not possible to find the correct length spacers in which case you can use long bolts where the position of the nuts can be adjusted to provide the necessary spacing. An example of board fixtures is shown in **Figure 4**. Insulating plastic spacers can either be purchased or made with a 3D printer. It is bad practice to rely on an electrical connector to also provide mechanical fixing; in this case any vibration or shock loading to the assembly will be transferred between the boards via the connector.

The removable SD memory card can also be a source of problems. Some earlier models of the RPi and also some of the recent versions rely on a simple friction of the SD card in the slot to stop it from shaking free and falling out. Other models use a positive retention mechanism which requires a push to release a catch mechanism and another push to lock the card in place. Over-enthusiastic attempts to remove these cards without releasing the catch mechanism can result in a damaged/dislodged card reader unit.

The use of hot glue to secure the friction-fit SD cards in place is not recommended; over-temperature events can cause the glue to creep into the card reader innards and create bigger problems.

It should be noted that SD cards wear out eventually and should not be relied upon as a long term data storage solution when continually subject to read/write cycles. The author experienced regular SDcard failures with a Raspberry Pi 1B, some of the cards delaminated as a result of thermal stress. The problem is not as common today as long as you do not use the cheapest unbranded cards.

Shenzhen Xunlong, makers of the Orange Pi, can bypass this problem by supplying units with soldered flash memory but this limits the memory size specified by the manufacturer. Programming these devices can only take place during system operation which can be quite time consuming when a large batch of units need to be programmed.

Thermal and vibration-related issues

Systems built for use in motor vehicles must be designed to function over a relatively wide operational temperature range. This is only logical, because the vehicle should be able to cope with every environment it it is likely to encounter around the globe. The Raspberry Pi, is made from commercial grade components which are generally specified for use in a temperature range from zero to 85 ° C.

The discussion at [2] touches on Raspberry Pi and BeagleBone variants with a slightly wider temperature range - but you don't see many of them on the market. One test carried out in 2013 on an original RPi used liquid nitrogen to discover it remained operational down at minus 110 degree centigrade.

The author had no problem using his Orange Pi-PC-Plus based system onboard a bus in the US. The discussion at [2] touches on Raspberry Pi and BeagleBone variants with a slightly wider temperature range - but you don't see many of them on the market. One test carried out in 2013 on an original RPi used liquid nitrogen to discover it remained operational down at minus 110 degree centigrade.

Temperatures at the other end of the scale are more difficult to deal with. The battle between the various process computer manufacturers cause them to push performance limits and use the thermal budgets of their systems more and more - a normal Raspberry Pi 4 can reach more than 55 ° C under full load in the lab at an ambient temperature of 20 ° C. This obviously restricts the temperature range within which the system can operate reliably but there are techniques we can use to improve the situation.

The first measure to consider is reducing the processor clock speed – with products from Shenzhen Xunlong this is generally

Practical accessories for professional use

Memory Cards from SwissBit [10]
Memory card failures have always been a problem in the field of process control. In particular, if the application uses high typing traffic (logfiles), the use of professional cards here is the best option.
SwissBit may not provide the cheapest solutions but they have a good reputation for quality and reliability.
Most of the items are based on robust flash memory cells which feature authentification and encryption protection to deter hackers. The cards also include expansion in the area of the controllers to stay robust and reliable in the event of power outages and other nastiness.

A CAN HAT [11]
The CAN bus is not the simplest network to install - timing problems, voltage problems and the ever-present spikes ensure an extremely comprehensive specification.

The PiCAN provides a convenient way to connect the Raspberry Pi to CAN 2.0. The board provider uses CAN controllers from MicroChip, which should ensure good compatibility - DB9 and normal terminals are provided for network connection. Programming is achieved through the SocketCAN driver, which runs code in C and Python.

Raspberry Pi on DIN rails [12, 13]
Phoenix Contact has been known as suppliers of factory automation products for many years now. The company supplies the RPI-BC housing series for the Raspberry Pi, which can be easily mounted on a DIN rail.

Phoenix Contact implements the in-house HBUS standard, which makes it possible to connect several Raspberry Pis and/or other IO modules to build a complete system. In the housing there is also enough space for additional boards on which you can accommodate various extensions.

Strato Pi - UPS and RTC [14]
Uninterruptible power supplies are very important in industrial applications, especially in the control sector. Several companies offer UPS extensions for the Raspberry Pi, which use batteries or a large capacitor reservoir to provide standby power or allow a clean shutdown to take place.

Some versions of the Strato Pi come with a UPS as well as a battery-backed real-time clock, which ensures time-of-day accuracy independant of the Internet. Some models have galvanically isolated serial buses, allowing communication via RS232, RS485 and CAN.

PoE Hat [15]
In larger industrial applications, wiring is often a nuisance - every extra wire costs and installation costs. Power over Ethernet (PoE) has always been attractive in this area because you can delegate power to a (UPS-powered) hub.

The Raspberry Pi Foundation has been offering expansion boards for some time, providing a (compatible!) Raspberry Pi with PoE tapping capability. Please note, however, that the board runs warm.

easy, but underclocking the new Raspberry Pi models requires a bit more effort. Reducing the clock speed cuts power consumption and reduces chip temperature but also throttles back on available processing power.
The second method is to use standard heatsinks mounted on the processor chip. This seems like an obvious solution but can be problematic especially in environments subject to high vibrational stress - James K. Hollomon has already stated in his classic book *Surface-Mount Technology for PC Boards* [3] that the forces experienced at the component/board interface during vibration increase directly with the component weight. This is one of the reasons why SMD components (with the exception of connectors) can withstand higher vibration levels than their through-hole counterparts.
Surface mounted port connectors are not as tolerant to abuse as their through-hole equivalents; their mechanical strength is usually derived from the quality of the solder connections alone. According to engineers at the connector manufacturer WAGO, faults with SMD connectors can often be traced back to the use of cables that are too heavy.
Whilst most motherboards and industrial computers offer the option of securing a heatsink to the PCB with bolts, the Raspberry Pi requires a more elaborate bracket arrangement to fix the heatsink and keep the SoC temperature within spec and allow the system to operate over a wider environmental temperature range. There is also the option to fit a fan to draw air through the case.
In this context, it should be noted that housing manufacturers are not limited to Kawaii products [4] for normal users. Companies such as ItalTronic [5] offer various housing solutions that

Advertisement

BERNSTEIN
TOOLS FOR ELECTRONICS

Your SPECIALIST for ESD- and
PRECISION TOOLS



Visit us at PRODUCTRONICA 2019
hall A4 / booth 577
www.bernstein-werkzeuge.de

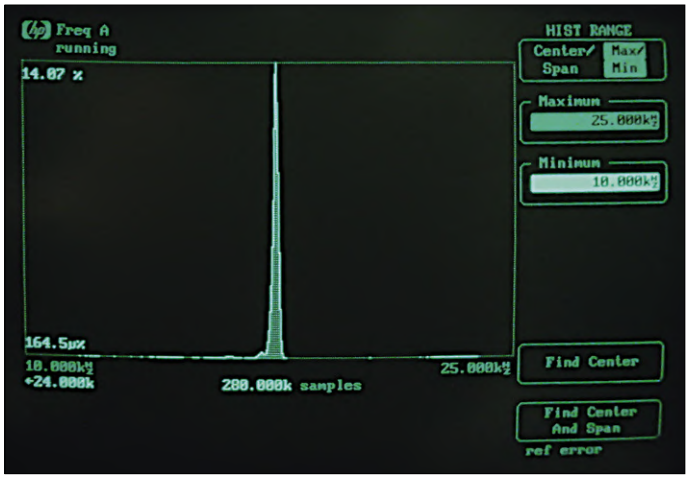


Figure 5. The weaknesses of the Raspberry Pi running Raspbian are clearly visible in the modulation domain analysis.

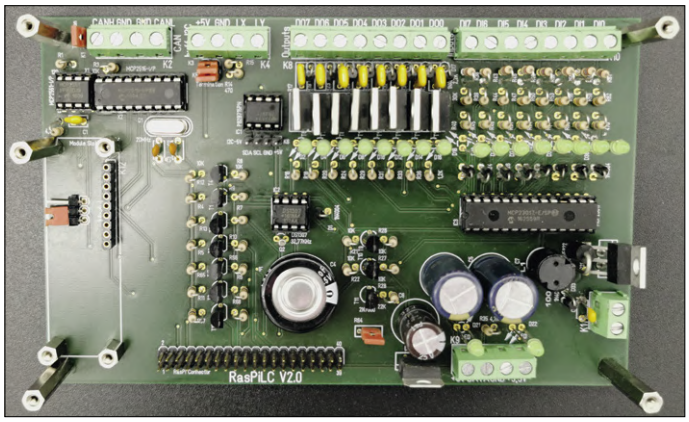


Figure 6. The RasPiLC is optimized for smart home applications.

can be mounted on DIN rails, for example. If you are serious about using the Raspberry Pi in a commercial product, you should take a closer look at these products.

Finally, it should be noted that the Raspberry Pi like almost all consumer SBCs use standard RAM technology; if you application demands ECC memory for whatever reason, this level of data security should be designed from scratch or provided via an ECC-equipped server.

Can we CAN? Yes we Can!

Whether Raspberry Pi or Orange Pi: The goal is always the commercial maker. It usually communicates with peripheral hardware via I2C, SPI or UART, and in some instances it may be possible to implement bit-banging using GPIO pins, but it's necessary to take into account the real-time response capability (**Figure 5**) of most operating systems.
Another problem is that the inputs to the process computer usually have a voltage range of 0 to 3.3V. Although the Raspberry Pi can tolerate 5 V input signal levels to some degree, it is definitely not recommended especially when the unit forms part of a commercial product. Just like many of the other single board platforms you cannot drive much power from the RPi GPIOs (16 mA per I/O, altogether a maximum of 51 mA for the chip). Any application requiring more will need an external driver stage with its own power source.
Off-the-shelf HATs are available to provide the necessary power interface and if the production run of your design is quite small then it can work out cheaper to use a readymade HAT rather than to develop your own from scratch.

In practice, one often encounters situations where additional bus protocols are needed. While one kernel module is available for OneWire (keyword: Dallas temperature sensors), more complex protocols are being used by CAN and others. For this purpose, it can sometimes make sense to implement a sepa-

Web Links

[1] Commercial use of the Raspberry Pi: www.raspberrypi.org/forums/viewtopic.php?f=63&t=176564
[2] Raspberry Pi for industrial applications: www.element14.com/community/thread/71335/l/raspberry-pi-for-industrial-uses?displayFullThread=true
[3] Holloman, James: Surface Mount Technology for PC Boards: www.amazon.com/Surface-Mount-Technology-PC-Boards/dp/0790610604
[4] Kawaii?: https://de.wikipedia.org/wiki/Kawaii
[5] Italtronic: https://deu.italtronic.com/
[6] RaspiLC: www.raspilc.de/
[7] Custom RPi solutions: www.element14.com/community/docs/DOC-76955/l/raspberry-pi-customization-service
[8] OpenSource-IPC https://revolution.kunbus.com/
[9] netPI: www.netiot.com/netpi/industrial-raspberry-pi-3/
[10] High-end, secure SD memory: www.swissbit.com/products/nand-flash-products/microsd-memory-cards/
[11] PiCAN: www.elektor.com/pican-2-can-bus-board-for-raspberry-pi
[12] DIN-rail enclosures for the RPi : https://bit.ly/2HnglLR
[13] DIN-rail enclosures for the RPi : https://bit.ly/31XJswY
[14] UPS, RTC and more: www.sferalabs.cc/strato-pi/
[15] Raspberry Pi PoE-HAT : www.elektor.com/poe-hat-for-raspberry-pi-3-b-plus

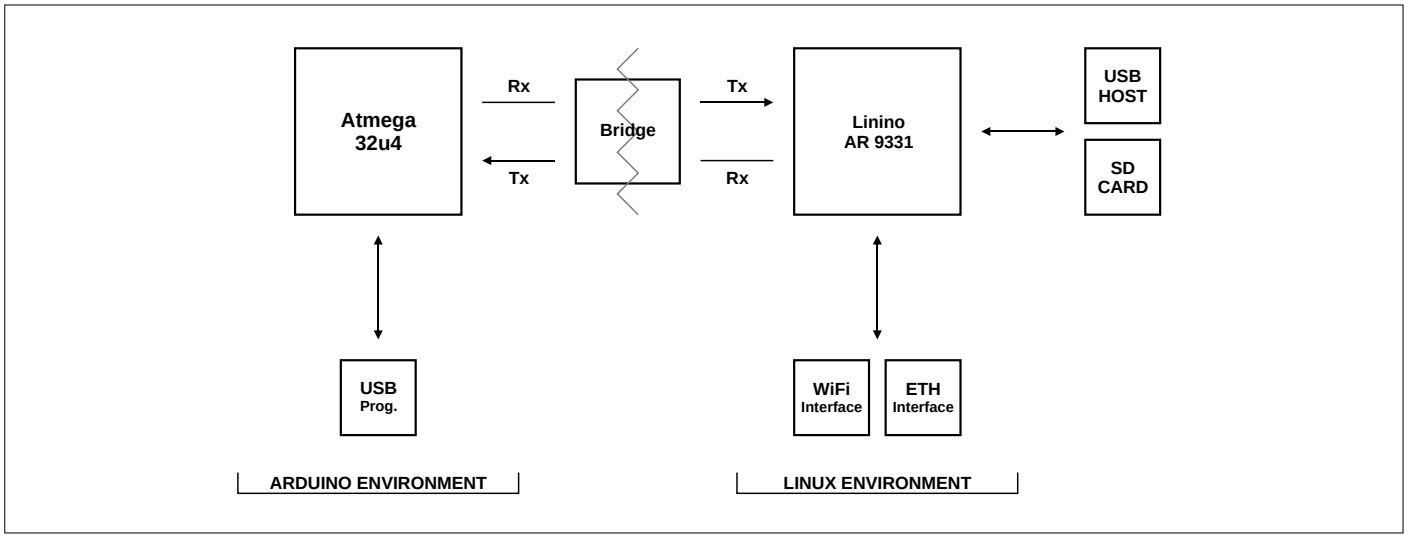


Figure 7. Any similarities to the Arduino Yun are purely coincidental...

rate hybrid process computer - a controller such as the ESP32 can link to the Raspberry PI via I2C or SPI and forward the incoming events.

Before you resort to this solution you should try to employ a spare USB port. For example, the author uses a low-cost USB GPS receiver to avoid the hassle of designing and testing a whole new module.

If this does not lead to the goal, one should look for suitable shields or HATs. Try entering "Raspberry Pi <Protocol>" into a search engine and you should come up with some helpful links. In the author's experience, a firm based in Schneeberg in Saxony provides some really useful hardware, offering RaspiLC [6] extensions for the smart home, such as the RasPiLC Advanced (**Figure 6**) which supports the CAN protocol.

Using this approach a developer can overcome problems resulting in the often poor real-time performance of a system running standard Linux. If a dedicated low-power controller takes care of event handling, the main processor can sometimes even be put into sleep mode. It can then be woken up when it's necessary to handle processor-intensive tasks. This approach extends battery life, an example can be seen in the diagram shown in **Figure 7** where the author has used both the PIC 16F1503 and STM32 series processors in an application.

The **box** lists websites offering some interesting add-ons for commercial use.

A Process-Computer Smorgasbord

A majority of the companies mentioned here advertise the 'open source' nature of their products so that all the production files are available but to modify and build the modules yourself with a Gerber file, involves a relatively heavy time investment. The Raspberry Pi Foundation - in collaboration with the manufacturing partner Element14 [7] - offers the possibility to supply customized RPis to meet your own particular application requirements. The minimum order size is 1000 units and price per unit will depend on quantity. The minimum order threshold rises to 5000 units if you want to use an older version of the SoC in the design.

One good example of an RPi Compute Module incorporated into a commercial product is the KunBus [8] range of 'ruggedised' DIN-rail mounted units which can be expanded using a variety of I/O modules and Fieldbus gateways for industrial control applications. Here the Compute Module runs a specially adapted Raspbian operating system, equipped with a real-time patch. The use of Raspbian ensures that most of the applications developed for the Raspberry Pi will also run on this system. Under [9] is another provider of a Raspberry Pi-derived process computer system.

Shenzhen Xunlong, makers of the Orange Pi range of boards, also offer a customized design facility of their products, subject to a minimum order quantity.

Conclusion

In my experience the single-board computers work without any problems both in metrology and in the automotive convenience-sector. So far, my clients have had no problems with systems supplied with either Raspberry Pi or Orange Pi-based process computers. The original idea behind the Raspberry Pi was as an educational tool and the Raspberry Pi Foundation is a registered charity. If you are planning to use it in an end-product the RPi Compute Module has a much better form factor for easier integration than any of the other versions. Security may also be an important factor in your design; the open-community support for the platform is a huge benefit for newcomers but together with its removable memory card makes any system based on them vunerable to hackers. ⏮

190331-02

@ WWW.ELEKTOR.COM

→ Raspberry Pi 4 B
www.elektor.com/raspberry-pi-4-b-4-gb-ram

→ HATs for the Raspberry Pi
www.elektor.com/catalogsearch/result/?q=raspberry%20pi%20hats

→ Literature for the Raspberry Pi
www.elektor.com/books/raspberry-pi-arduino-esp/raspberry-pi

Err-lectronics

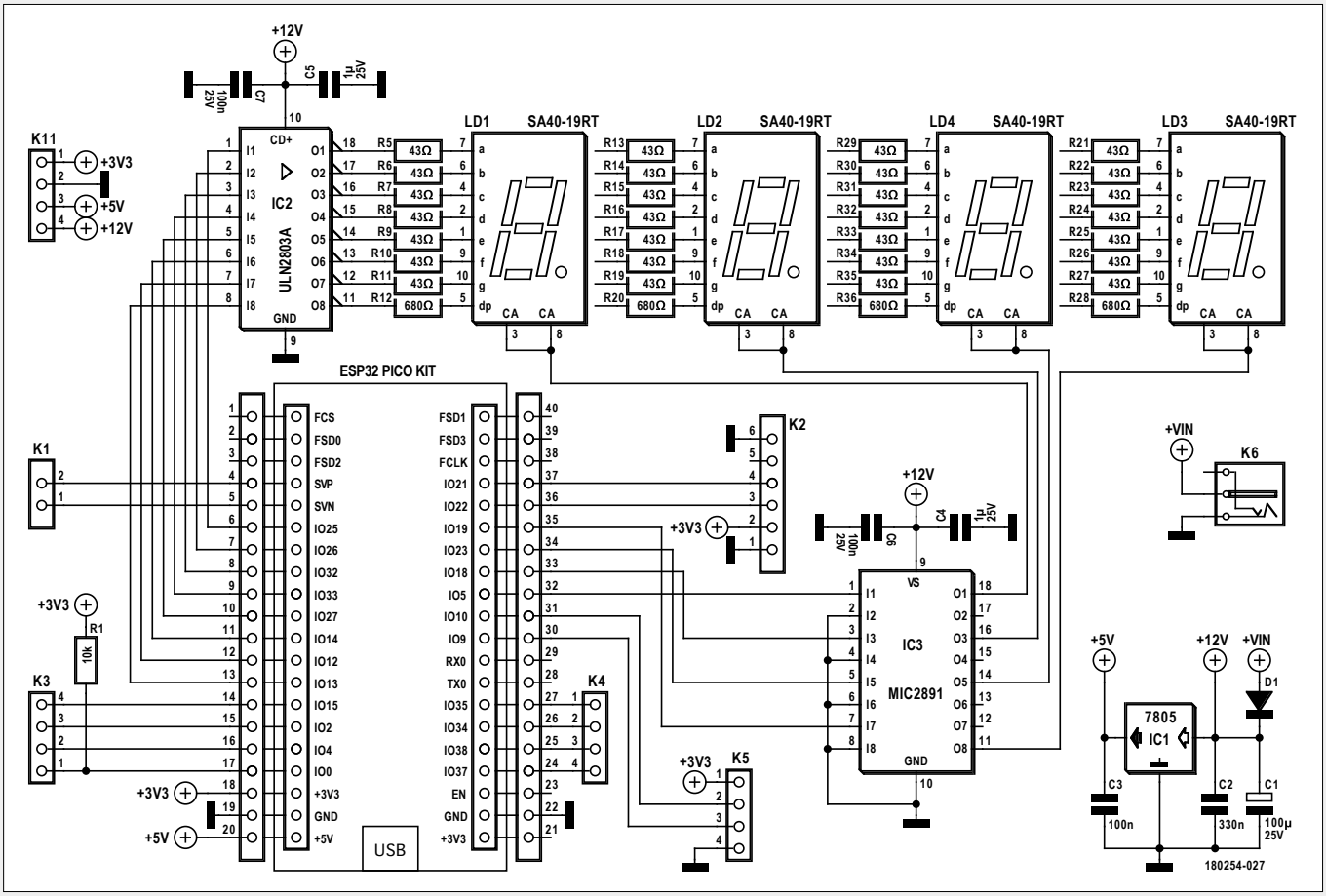
Corrections, Feedback and Updates to published articles

Monster LED clock with Wi-Fi and Temperature Display

ElektorLabs 4/2019 (May & June) p. 60 (180254)

UPDATE. Unfortunately an error has crept in the circuit diagram. The pin assignment of the 7-segment displays in the schematic as printed does not match that on the board. The corrected version of the schematic is shown here.





Monster LED clock with Wi-Fi and Temperature Display

ElektorLabs 4/2019, (May & June), p. 60 (180254)

MQTT Sensor Hub

ElektorLabs magazine 5/2019 (September & October), p. 6 (180254-B)



CORRECTION. R1 is erroneously shown as a 10 kΩ resistor in the circuit diagram and the parts list and should be corrected to read 1 kΩ. As long as a 7805 is used the 10-kΩ value should not cause problems. If however a DC/DC converter is used, the pin with R1 as the pullup must ensure a slightly higher current to prevent the ESP32 from jumping into boot mode unintentionally. Since the pin does not drive any other functions, a 1-kΩ pullup does not present problems.
Diode D1 should be a type MBRS540, not BAT85.

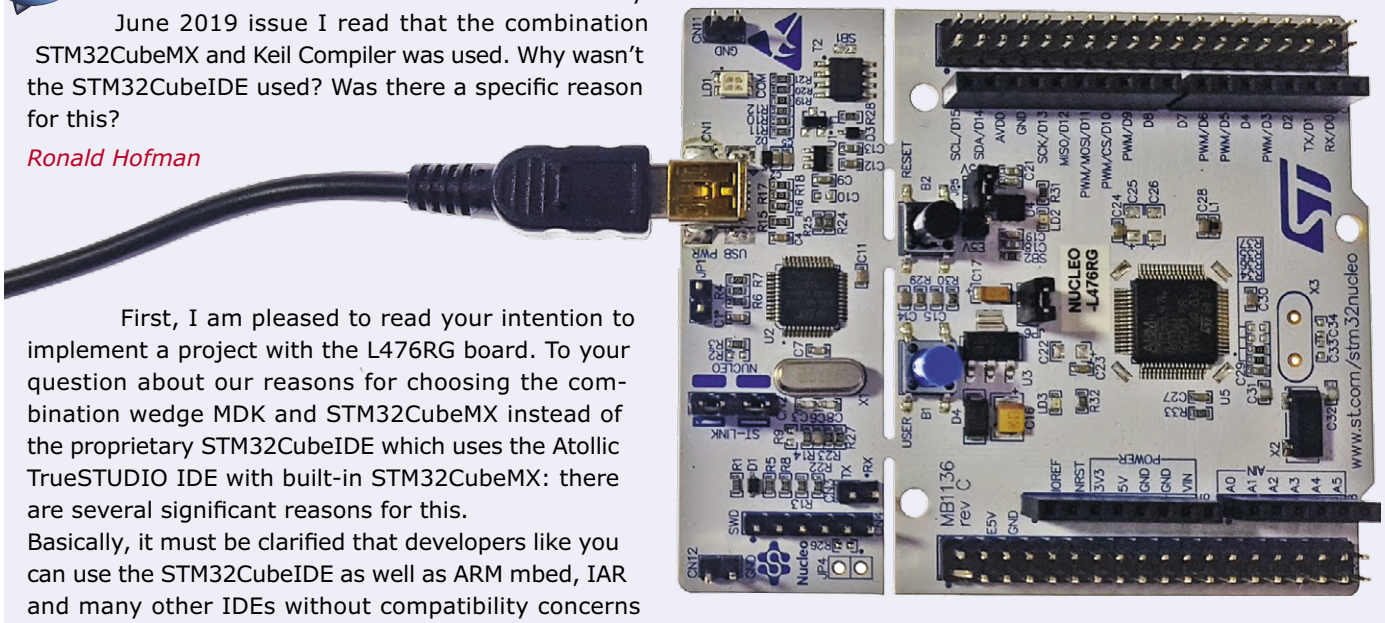


Microcontroller Kits for Intermediate Users

ElektorLabs 3/2019 (May & June), p. 38 (180604)

FEEDBACK. I started with the STM32 and bought an L476RG Nucleo board. At ST I downloaded the STM32CubeIDE. In the article in the May& June 2019 issue I read that the combination STM32CubeMX and Keil Compiler was used. Why wasn't the STM32CubeIDE used? Was there a specific reason for this?

Ronald Hofman



First, I am pleased to read your intention to implement a project with the L476RG board. To your question about our reasons for choosing the combination wedge MDK and STM32CubeMX instead of the proprietary STM32CubeIDE which uses the Atollic TrueSTUDIO IDE with built-in STM32CubeMX: there are several significant reasons for this.
Basically, it must be clarified that developers like you can use the STM32CubeIDE as well as ARM mbed, IAR and many other IDEs without compatibility concerns and with existing support as well as documentation of the manufacturer, without problems.
We chose the 'wedge' ecosystem because of its wide distribution and manufacturer independence, as this MDK is offered directly by ARM. On the other hand, the TrueSTUDIO IDE from Atollic integrated in the STM32CubeIDE, became manufacturer-dependent recently after its purchase by ST and will soon be available in the form of the STM32CubeIDE only. Experience has shown that most professional developers use the Keil MDK because it allows them to use the entire ARM range of different manufacturers in this category, and because the migration hurdles are minimal. In addition, there is the thought that some advanced users also use the universal debugger hardware from Keil. You will also find many documented ST applications on the net that use the Keil compiler.
Nevertheless, Keil should not appear to you to be the ultimate. If, for example, you want to implement your project with extension boards from ST or third-party manufacturers quickly and without high efficiency and security requirements, mbed and the corresponding code libraries are often your best choice.
Good luck with your project!

Viacheslav Gromov (co-author)

Improved Radiation Meter — Improved!

a hardware update for the alpha, beta and gamma counter

By **Ton Giesberts** (Elektor Labs)

Back in 2011 Elektor published an ingenious radiation meter that worked without a special, expensive and difficult to obtain Geiger-Müller tube. Author Burkhard Kainka, no stranger to faithful Elektor readers, showed beyond refute that a simple photodiode with some additional electronics was enough to show radiation. A few months after 'Fukushima', it was a hot topic.

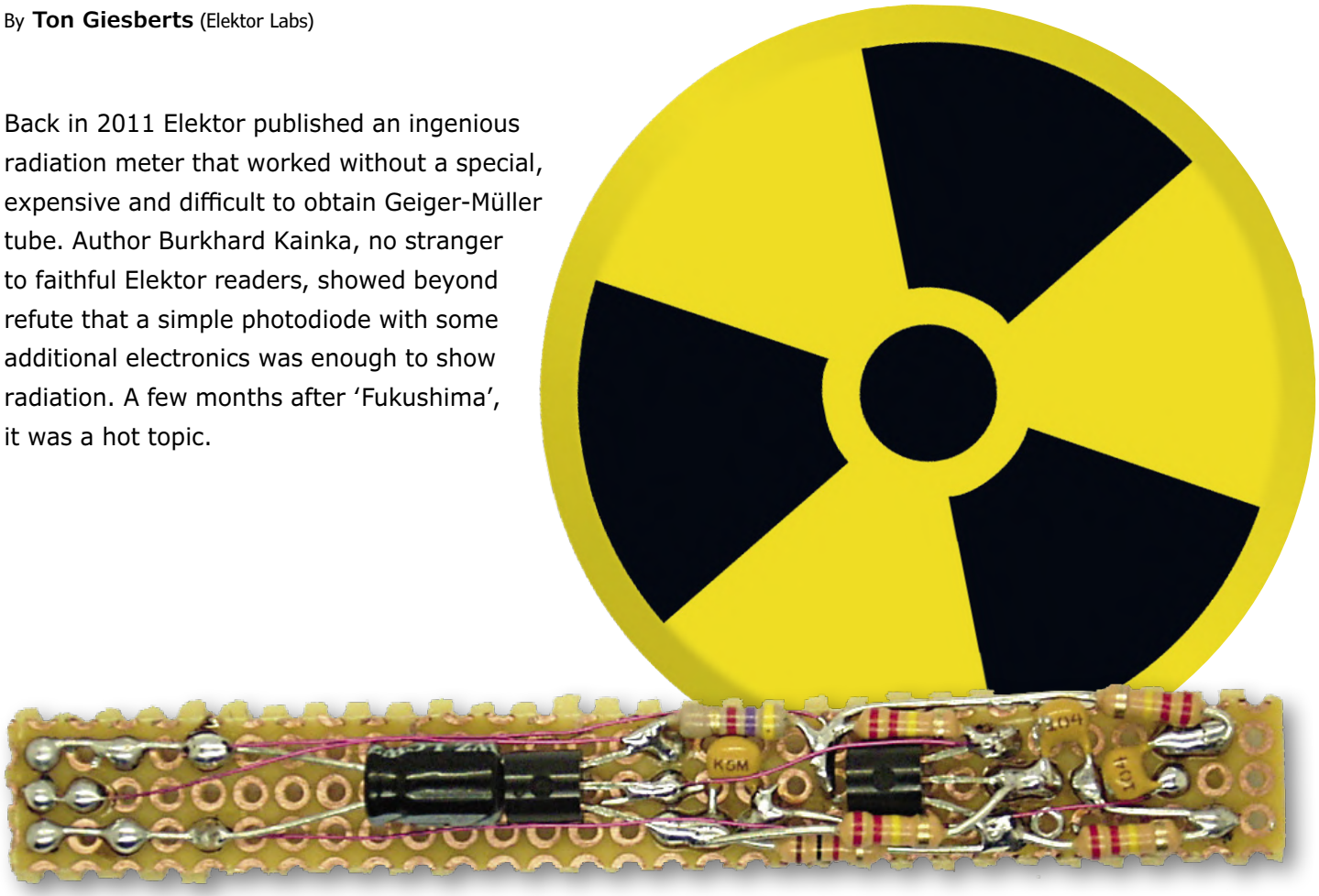


Figure 1: The original Radiation Meter from Elektor June 2011 was slapped up on a piece of perfboard yet worked well.

Quick Features

- Newly designed double-sided PCB
- Through-hole parts only
- Runs existing software
- Kit of parts available

Radioactive radiation (i.e. ionizing radiation) may reach us in three forms, which are also dangerous to different degrees. In the first place there are alpha rays — these are actually not rays but loose ^{4}helium atom nuclei (two protons plus two neutrons) released during the decay of radioactive materials. These particles are heavy and have a very limited range and ditto permeability; they can't even get through a piece of paper.

Then we have beta rays — actually no rays either, but loose electrons released when a neutron in an atomic nucleus changes into a proton when radioactive decay occurs. Beta rays have a higher

permeability and a greater range than alpha rays and are therefore more of a health risk. However, screening is perfectly possible with acrylic sheet (plexiglassTM) about 10 mm thick, or a not too thin piece of aluminium.
Finally, there is gamma radiation — the really dangerous 'hard' radiation that's extremely bad for your health and for which proper shielding is difficult to achieve (walls hardly stop this radiation, for which a thick layer of a dense and heavy material such as lead is required). Gamma radiation is naturally omnipresent: it is the cosmic radiation with which the earth (and everything on it) is continuously bombarded.

The original design ...

... is described in detail in the back article on the radiation detector [1] (Elektor June 2011), to which we refer for further information.
This original design was actually more of a proof of concept built with discrete parts on a piece of perfboard (**Figure 1**). That of course called for a follow-up which duly arrived in November of the same year [2].

... and the improved version ...

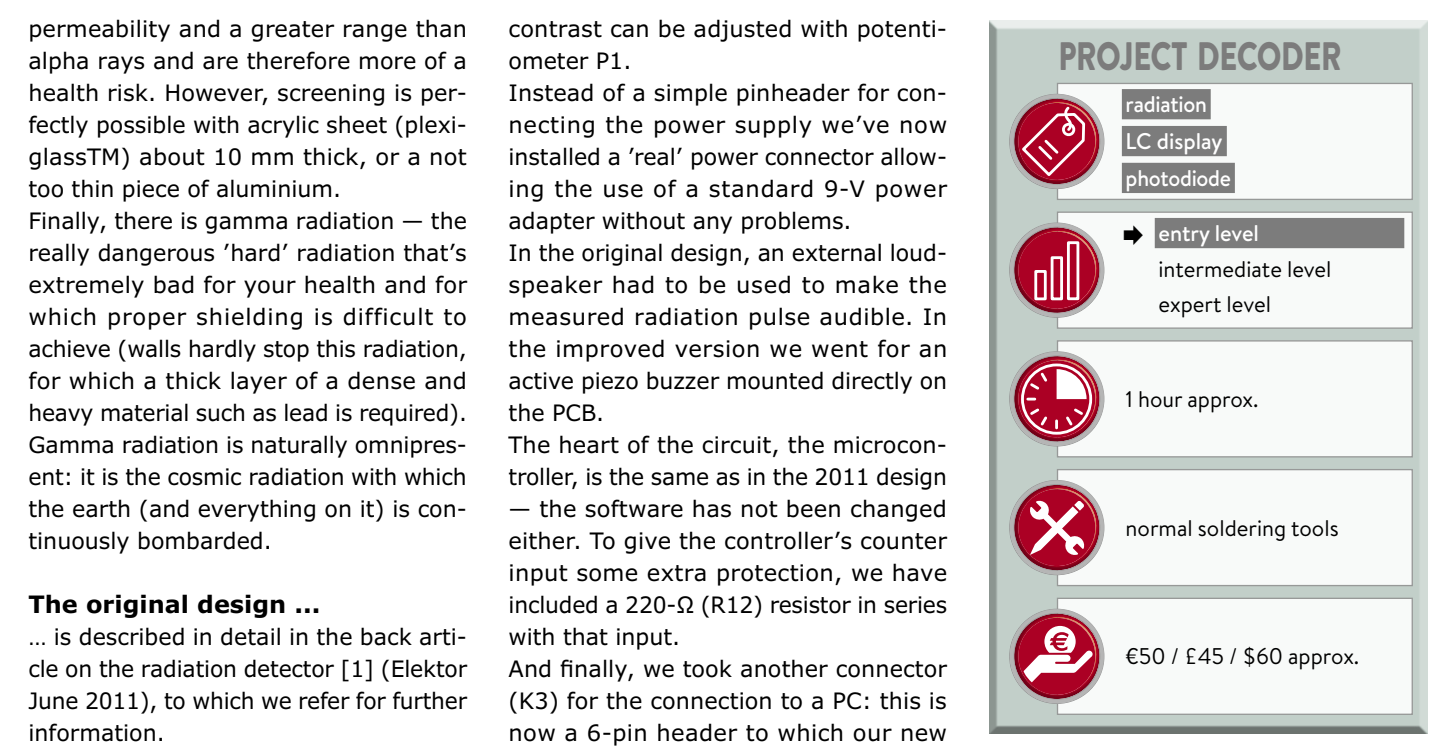
The extremely popular Improved Radiation Meter was built on a real PCB, had an enhanced preamplifier, and a microcontroller plus LC display. This resulted in a much more user-friendly radiation meter that also produced measurements from which a lot of interesting information could be derived. Again, we refer to the original article for more information. **Figure 2** gives an impression of this version.

What was left to improve?

Figure 3 shows the schematic of the improved version. Comparing it with the schematic in [2] you can only agree that very little has changed.
JFETs in a TO92 housing are difficult to obtain today, especially the good old BF245 we used in the original design. The BF256B is just as adequate though and more widely available, and now sits prominently in the new schematic.
In addition, we have now used another LC display: a green/black model with background lighting, which is available in the Elektor Store (see **@ www.elektor.com text box**). The backlight can be switched on and off with slide switch S2, depending on the conditions. The contrast can be adjusted with potentiometer P1.
Instead of a simple pinheader for connecting the power supply we've now installed a 'real' power connector allowing the use of a standard 9-V power adapter without any problems.
In the original design, an external loudspeaker had to be used to make the measured radiation pulse audible. In the improved version we went for an active piezo buzzer mounted directly on the PCB.
The heart of the circuit, the microcontroller, is the same as in the 2011 design — the software has not been changed either. To give the controller's counter input some extra protection, we have included a 220-Ω (R12) resistor in series with that input.
And finally, we took another connector (K3) for the connection to a PC: this is now a 6-pin header to which our new USB/RS232 converter (the FT231X BoB from Elektor 5/2019) can be hooked up directly. K5 is the standard Atmel 6-pin ISP connector for reprogramming the controller in-system if necessary.

Printed circuit board

And that brings us to the most important improvement: Elektor Labs have designed a completely new circuit board for the radiation meter (see **Figure 4**). Compared to the previous version, the identifiers for the various connectors are clearer. As with the previous version, the PCB has two sections: one with the controller and LCD, and one with the sensor



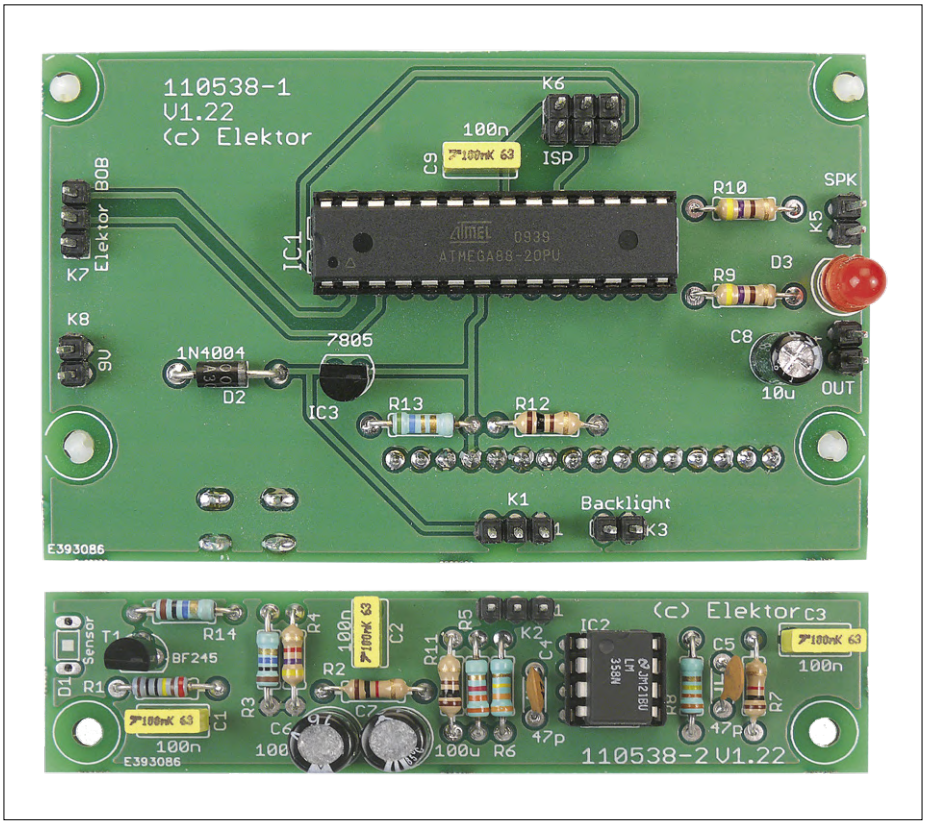


Figure 2: The improved version from Elektor November 2011 on a 'real' printed circuit board.

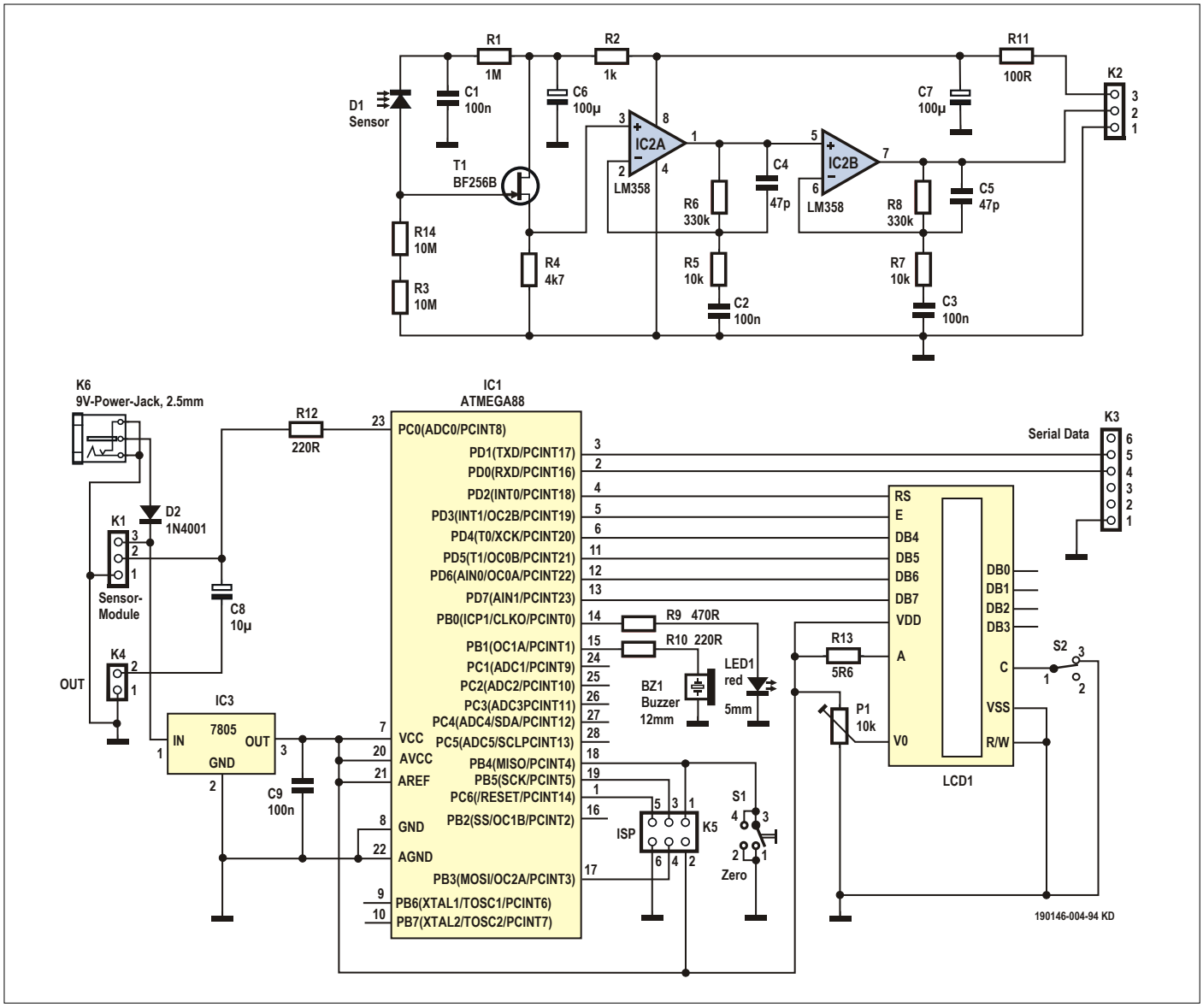


Figure 3: The schematic of the once-improved version differs only slightly from the original version.

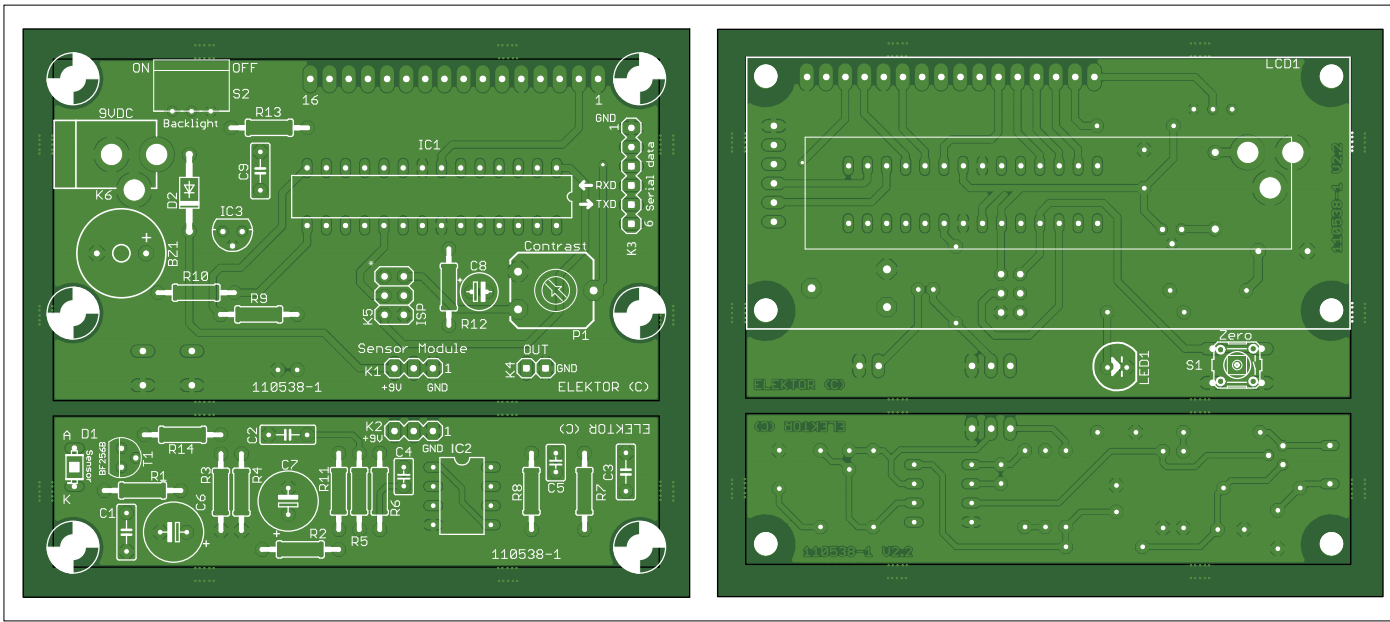


Figure 4: The new PCB has the same dimensions as the old one.

COMPONENT LIST

Resistors

All 5%, 0.25W, 250V
R1 = 1MΩ
R2 = 1kΩ
R3,R14 = 10MΩ
R4 = 4.7kΩ
R5,R7 = 10kΩ
R6,R8 = 330kΩ
R9 = 470Ω
R10,R12 = 220Ω
R11 = 100Ω
R13 = 5.6Ω
P1 = 10kΩ trimpot, horizontal

Capacitors

C1,C2,C3,C9 = 100nF, 50V, X7R, 0.2" pitch
C4,C5 = 47pF, 50V, C0G/NP0, 0.1" pitch
C6,C7 = 100µF, 50V, 3.5mm pitch, 8x11mm
C8 = 10µF, 50V, 2mm pitch, 5x11mm

Semiconductors

D1 = BPW34, straight
D2 = 1N4004
LED1 = 5mm diam.
T1 = BF256B
IC1 = ATMega88PA-PU (programmed, Elektor Store # 110538-41)
IC2 = LM358
IC3 = 78L05

Miscellaneous

Bz1 = piezo buzzer, 12mm, 1-3V_{pp}, 2.04kHz f_{res}, 6.5mm
IC1 = IC socket, DIP-28 narrow
IC2 = IC socket, DIP-8
K1.K2 = 1x3 pinheader, straight, 0.1" pitch
K3 = 1x6 pinheader, straight, 0.1" pitch
K4 = 1x2 pinheader, straight, 0.1" pitch
K5 = 2x3 pinheader, straight, 0.1" pitch
K6 = 12V, 3A DC power connector, 1.95mm pin, (Lumberg NEB 21R)
LCD1 = LCD module, 2x16, Elektor Store # 120061-74 (+ 16-pin pinheader, straight, 0.1" pitch)
Mounting material for LCD1
S1 = pushbutton, PCB mount, 6x6mm
S2 = slide switch, PCB mount, right angled, 0.1" pitch, (Würth Elektronik # 450302014072)
PCB no. 110538-1 v2.2

and preamplifier. Of course, this PCB is available in the Elektor Store, just like the ready-programmed microcontroller and a kit of parts including the PCB. The latter, pssst, for the lazier among you.

Construction

We can be brief about the construction: only through-hole (TH) parts are mounted on the PCB, so that shouldn't be a problem for anyone. Switch S1, LED1 and the LC display are mounted at the bottom of the controller board. As shown by the picture of our prototype in **Figure 5** we mounted the LC display with M3 bolts and spacers. But here, depending on your wishes and demands, you can let your imagination run free; it's not all that critical if it's solid.

All that remains for us to do is to wish you a lot of pleasure building and using the project — and not encountering high radiation levels in your immediate vicinity!

(190146-02)

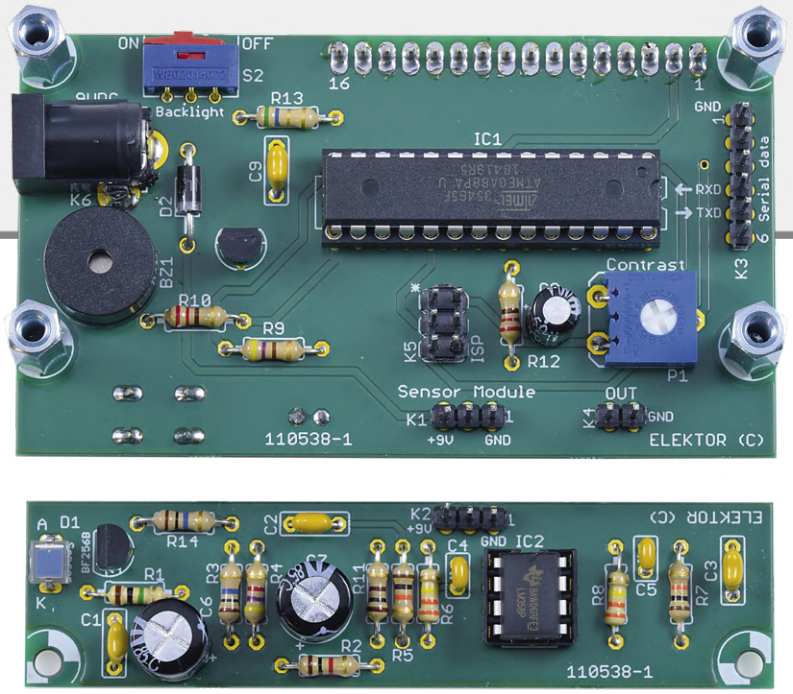


Figure 5: Even inexperienced soldering artists should have few problems with the construction.

@ WWW.ELEKTOR.COM

- **LCD green/black with backlight:** www.elektor.com/120061-74
- **FT231X BoB, bare PCB:** www.elektor.com/18878
- **FT231X BoB, assembled board:** www.elektor.com/18895
- **Radiation Meter, bare PCB v2.2:** www.elektor.com/110538-1
- **Radiation Meter, programmed microcontroller:** www.elektor.com/110538-41
- **Radiation Meter, kit of parts incl. PCB:** www.elektor.com/110538-71

Web Links

[1] Original Radiation Meter: www.elektormagazine.com/110372
[2] Improved Radiation Meter: www.elektormagazine.com/110538
[3] Project page for this article: www.elektormagazine.com/190146-02



VFD Tube Clock

Old wine in new (vacuum) bottles

By **Thomas Pototschnig** (Germany)

Vacuum fluorescent displays (VFDs) from the old tube times never fail to impress with their nostalgic charm and as a result they are often used for clock projects. The VFD Tube Clock presented in this article focuses on a minimalist look while offering other useful features.

Vacuum fluorescent displays (VFDs) from the old tube days never fail to impress with their nostalgic charm and as a result they are often used for clock projects. The VFD clock presented in this article focuses on a minimalist look while offering other useful features.

Vacuum fluorescent displays exert an almost magical attraction, prompting many developers to use them for clock projects. A particularly popular display is the IV-18 tube, which can be found among others in the Russian desktop calculator MKU1-1 [1] from 1989. The tube also appears under the type designations IV18, IW-18.

Our project also uses this type of tube, but we aimed for a minimalist design. If possible, only the tube itself should be visible and not the control electronics, which raises some design restrictions. On the one hand, the control electronics should be located completely behind the tube, which is only 18 mm in diameter. With a length of about 110 mm, the IV-18 is relatively long but a laterally protruding, newfangled USB connector would ruin its appearance. For this reason, the circuit board may only be about half as long as the tube.

As another consideration, VFD displays require a relatively high anode voltage of approximately 30 V requiring a DC/DC converter and, for each control signal, a driver that converts 5-V logic levels into voltages suitable for the VFD display. This of course takes up space on the circuit board. Even the usual SO28 VFD drivers are relatively (too) large and should not be used, apart from the fact that many special VFD ICs are obsolete anyway and/or expensive.

All these design specifications could be realized in the VFD tube clock, and even more: in addition to the time display there are other interesting and up to date functions.

The clock is set with an IR remote control whose codes are stored in the microcontroller of the clock. The USB port not only supplies the clock with power, it is also a USB-HID (human interface device) that can be addressed by a PC. In addition to reading and saving IR codes, it is also possible to output scrolling texts on the display. This offers many possibilities like status output of a server, temperature display of CPU cores, number of Youtube subscribers and so on ...

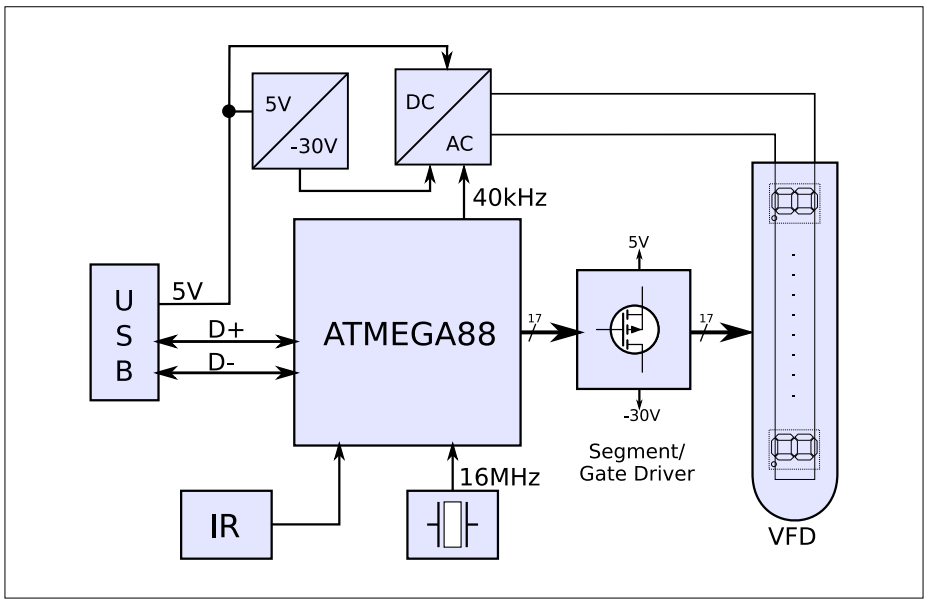
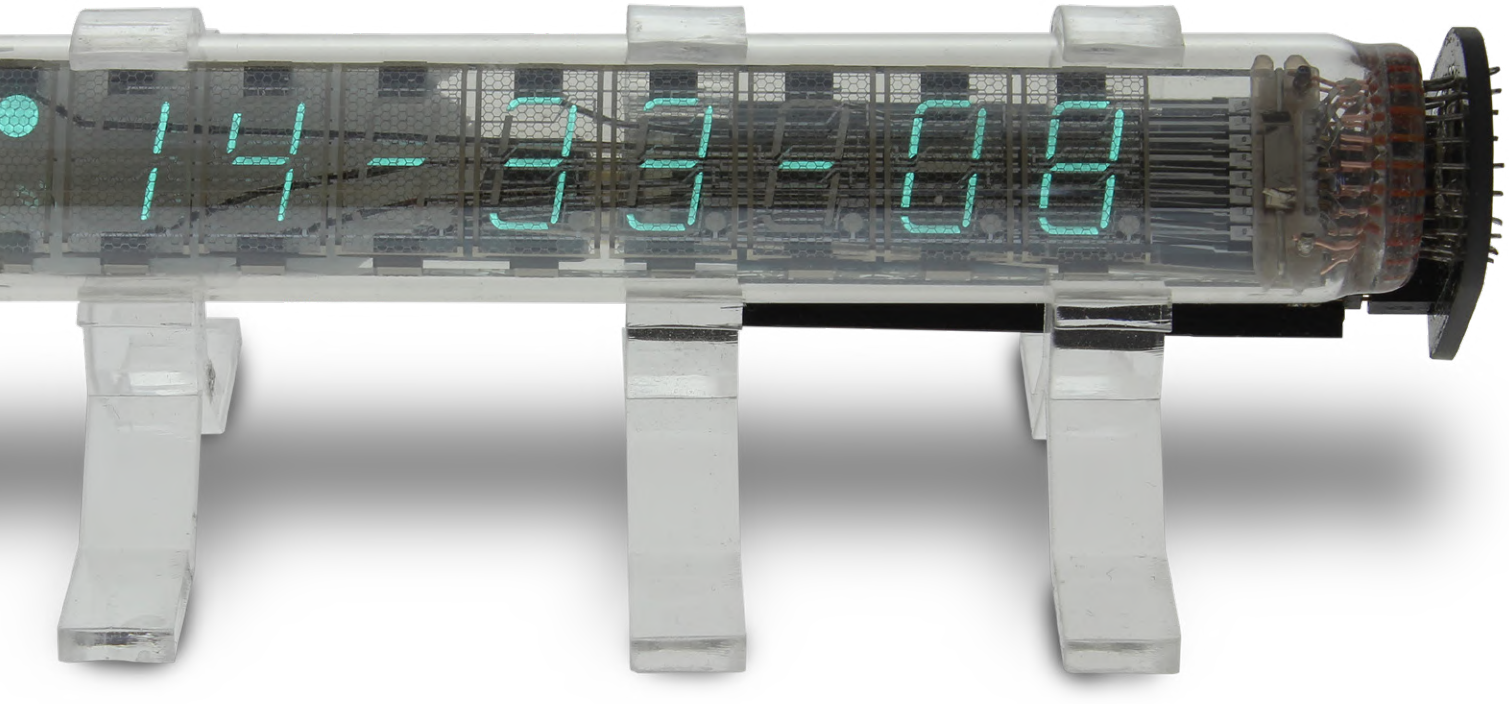


Figure 1: Block circuit of the VFD clock.

VFD control

Figure 1 shows the block diagram of the project. At the heart of the circuit is an ATmega88 microcontroller that controls all of the circuit's functions. The circuit is supplied with 5 V through USB, also permitting data to be exchanged with the microcontroller via this connection. A voltage inverter generates a voltage of around −30 V from the 5 V rail, which is required for the 17 MOSFET drivers for segments and grids as well as for gen-



erating the filament voltage. The circuit also includes an IR receiver to set the clock without data communication via a remote control.

The popular vacuum fluorescence display type IV-18 [2] is used in this project. Although these tubes haven't been produced for a long time, you can still find 'new-old-stock' specimens on many auction sites on the web, so you can easily buy unused tubes for a modest amount. Since such tubes may be 20-30 years old or more, it's a good idea to grab several as a precaution.

The function of a VFD tube is comparable to directly heated amplifier tubes. A heating wire (filament, cathode of the tube) emits electrons, which like in a triode, are attracted or repelled by positively or negatively charged grids and anodes (in this case the phosphor-coated segments). If the potential of a segment is more positive than that of the filament, electrons are attracted and hit the phosphor layer, which then lights up in the typical blue-green colour. If a segment is more negative, the electrons are repelled and the segment remains dark.

As shown in **Figure 2,** seven segments (eight with decimal points) are arranged in the form of common seven-segment displays and thus form a digit. In order to reduce the number of control lines required, VFD displays often use a time-division multiplex method (section 3.1.2 in [3]), which reduces the number of control lines — in the case of the IV-18 tube — from 66 to a happier 17.

For this purpose, VFD displays have a separately controllable grid for each position (digit) which sits between filament and segments and interrupts the electron flow in case of negative charge. This allows entire digits to be activated or deactivated in a flash, permitting time-division multiplexing (TDM). While all other digits are deactivated, one digit is activated at a time and the bit pattern matching the active digit is applied to the segments. If the switching frequency is high enough -— 500 Hz in this project — a flicker-free display is produced.

The segments and grids of a VFD display are controlled with voltages of about 30 V. The potential reference (potential of the filament) can be chosen almost arbitrarily, but the control electronics can be kept simple by a clever choice of the potential.

Figure 3 shows the selected potentials. There is an ON potential of +5 V and an OFF potential of –30 V, with which grids and segments can be switched on and

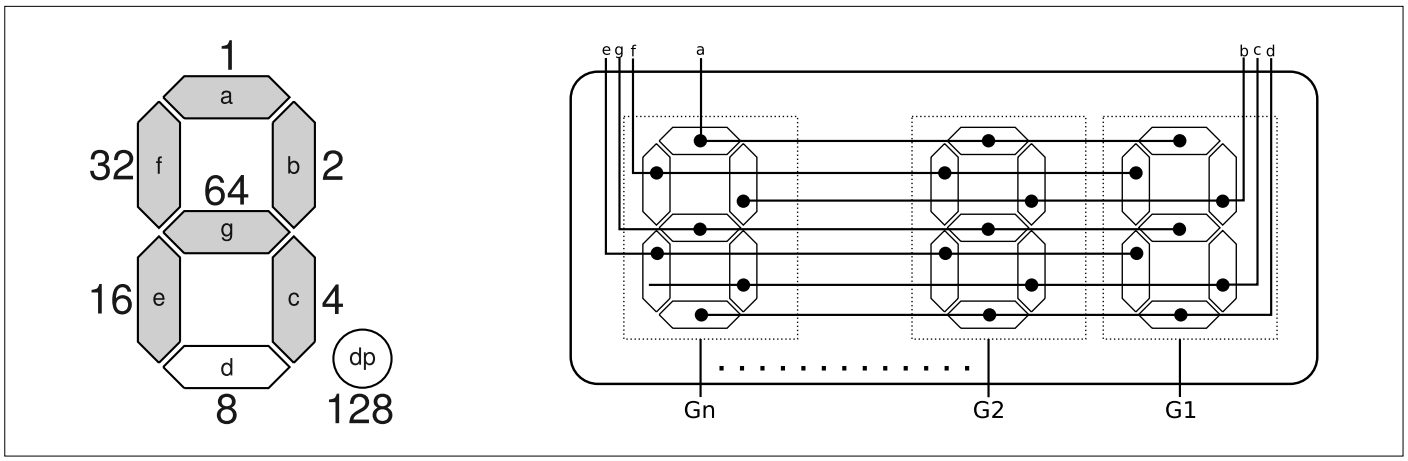


Figure 2: Control lines of the IV-18 (IW-18) tube, and structure of a digit.

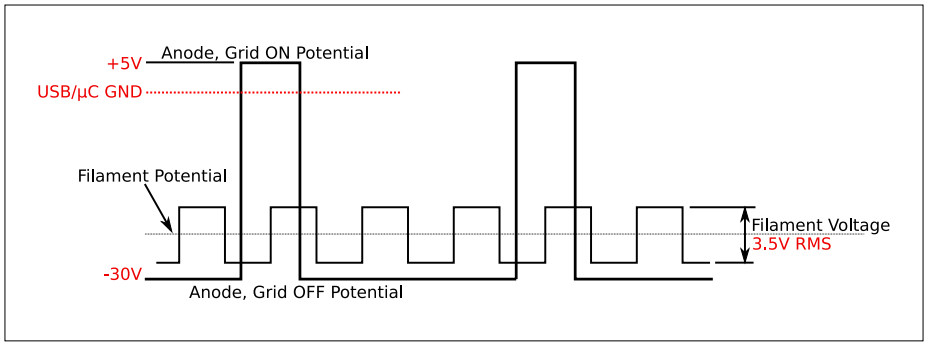


Figure 3: Potentials of the control signals.

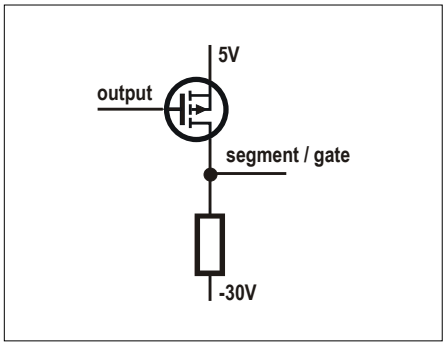


Figure 4: Simple MOSFET driver.

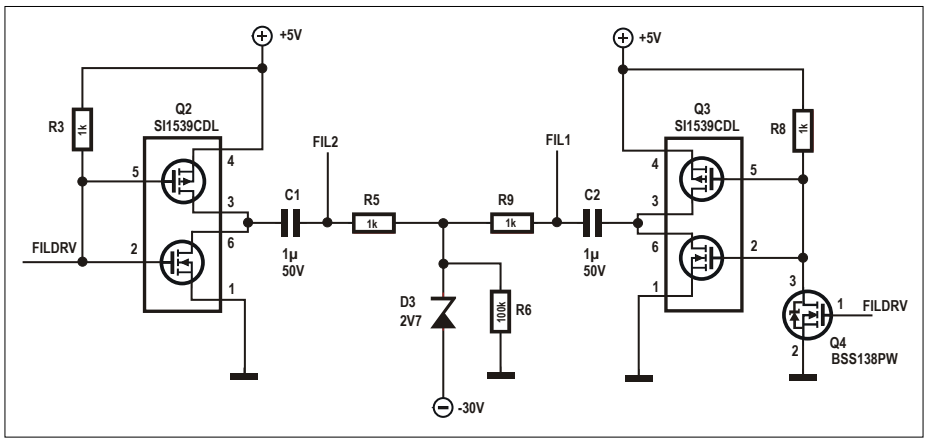


Figure 5: Two complementary MOSFET pairs, two half bridges.

off. The filament (cathode) 'floats' a few volts above the OFF potential to prevent ghosting during multiplexing. This choice of potentials makes the control of the grids and segments very simple, because low-cost logic-level P-channel MOSFETs such as the BSS84 can be used for the controls which the microcontroller can switch directly without any problems (**Figure 4**).

If a microcontroller output is switched to +5 V (U_{gs} = 0 V), the MOSFET blocks and -30 V is connected to its drain (and thus to a segment or grid). If the output is switched to 0 V (U_{gs} = −5 V), the MOSFET conducts and the segment or grid is at +5 V. The BSS84, a double MOSFET variant in a tiny SC70 package, has the voltage rating and is therefore perfectly suited for this application. A total of nine of these dual transistors is required — four for the segments and five for the grids.

The required negative voltage can be easily generated with a voltage inverter such as the LT1931, which is housed in a compact SOT23-5 package.

The last tricky part is the filament supply voltage. The inventors of the VFD displays, Noritake, write ([3], section 5.4) that alternating voltage should be used, because with direct voltage a decrease in brightness would be visible over the length of the tube. Many filament drivers rely on solutions with DC/AC converters and transformers (e.g. Royer converters), which unfortunately require a lot of space. However, Noritake in addition state that the filament can also be operated with a rectangular alternating voltage and recommend a frequency range of 10–200 kHz.

This can easily be achieved with a full bridge whose two half bridges consist of a complementary MOSFET pair (**Figure 5**). The two half bridges are driven in complementary fashion with a pulse-pause ratio of 50%, producing a potential difference between the drains of the two dual MOSFETs. Consequently an alternating voltage arises with an effective value of approximately 3.5 V, which is within the specification of the tube. The voltage is decoupled via capacitors C1 and C2 and pulled to −27.3 V via R5, R9, R6 and D3. SPICE simulations of the duals MOSFETs and later measurements of the assembled circuit showed that a dead-time setting of the two half bridges is not necessary. This reduces the control of the filaments to a single clock signal of 40 kHz, saving valuable outputs on the microcontroller.

In the complete circuit diagram (**Figure 6**) two further function groups are connected to the microcontroller, namely the USB interface and the IR remote control receiver.

USB on a shoestring

The Soft-USB stack called 'V-USB' can be used on any AVR microcontroller with at least 2 KB of Flash, 128 bytes RAM, and 12 MHz clock frequency [4]. If the microcontroller is operated at 5 V, only five components are required, as shown in **Figure 7.** This makes V-USB interesting for all kinds of applications. In order to keep the USB timing as exact as possible, the developers created optimized assembler routines for seven different clock frequencies [5]. For minimum code size, a 16-MHz crystal oscillator is recommended as used in this project. You should also consider that V-USB requires the highest-priority interrupt input, INT0, and interrupts must not be disabled in the software, otherwise the timing cannot be adhered to and communication errors can occur.

Communication with the microcontroller is by HID reports with a structure as shown in **Figure 8**. The IRD and IRC fields contain the last IR code received and a counter that counts the number of successfully received IR codes. A command can be passed in the CMD field to change the display mode, one-time teach IR codes, or set the time. The PARAMETERS field contains parameters for the individual commands, RAWDATA raw segment data (e.g. for text output

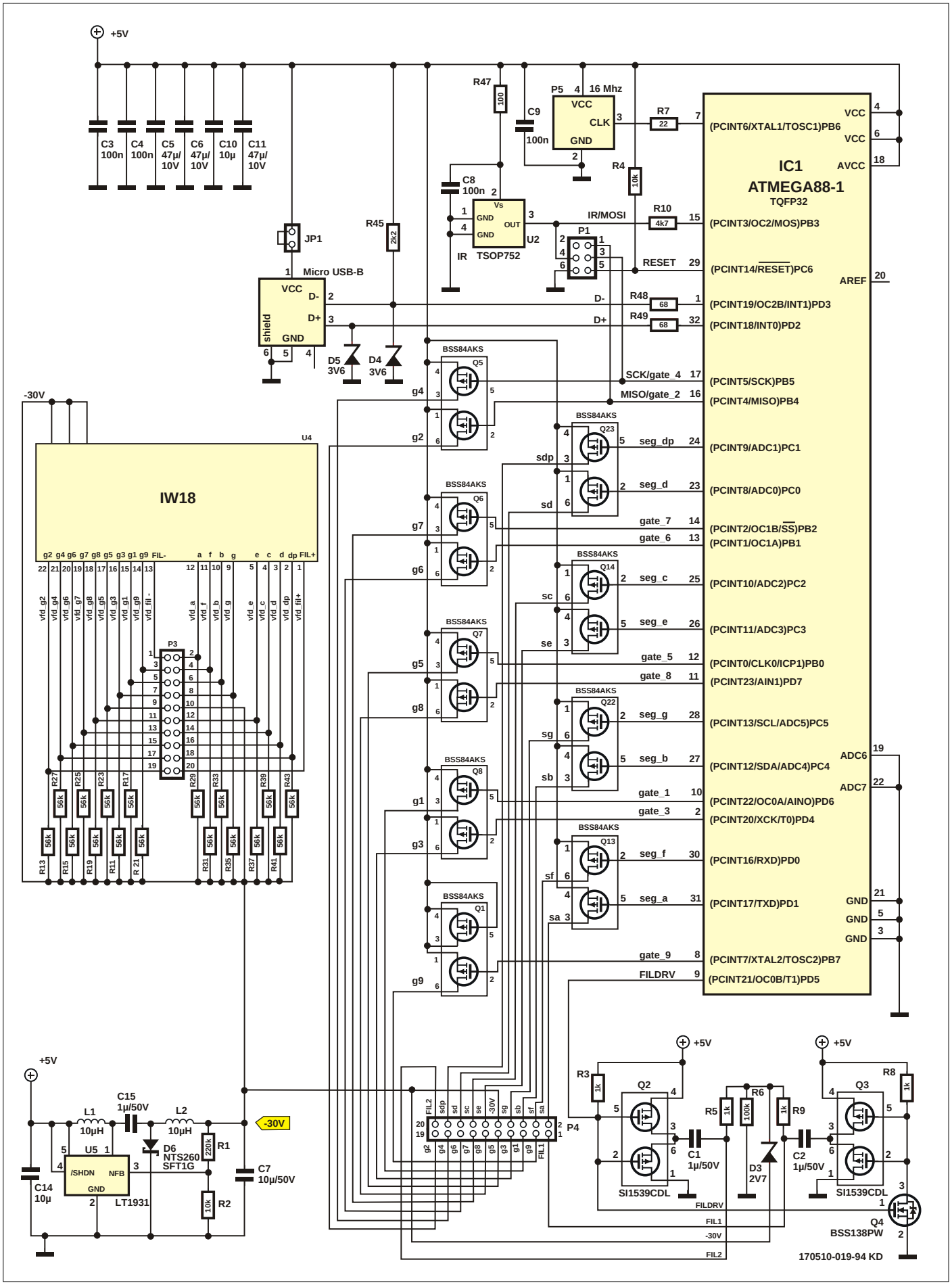


Figure 6: Complete circuit diagram of the VFD Clock.

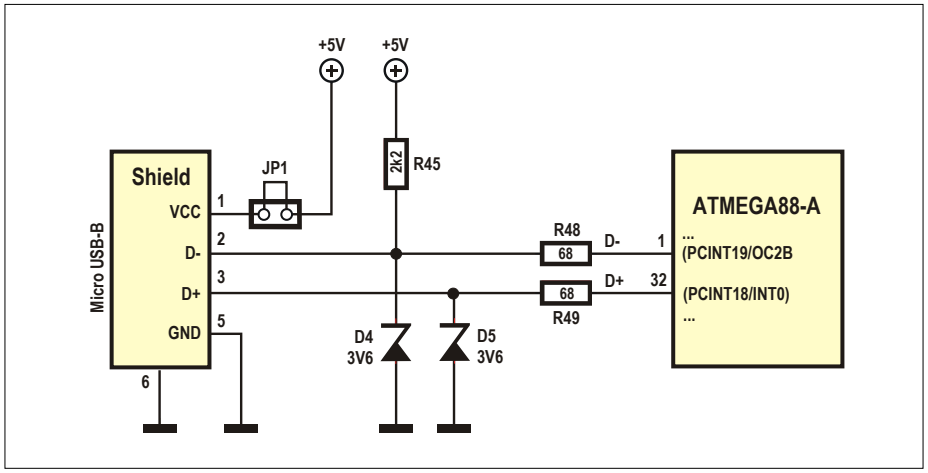


Figure 7: A USB port with five components.

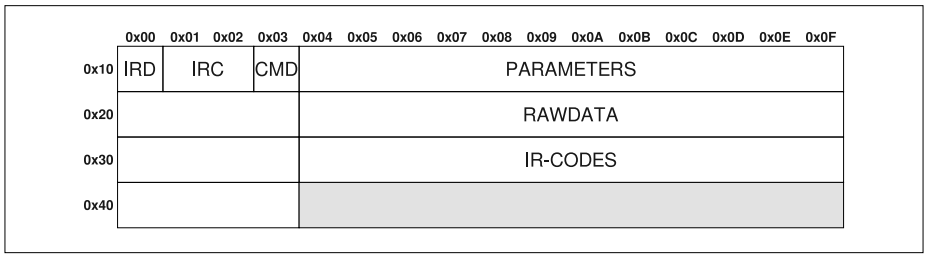


Figure 8: Control lines of the IV-18 (IW18; IW-18) tube.

on the tube), and IR-CODES, codes of the remote control. The commands are explained in the **text box.**

Infrared remote control

The project uses the IRMP library [6], which supports a variety of IR remote controls. In this project an IR receiver for remote controls with 38 kHz carrier frequency was used.

Before a remote control can be used, the control codes must be learned, for which a Python program exists in the GIT repository [10]:

```
$ sudo python setir.py
found: 0x16c0 0x05df VFD Clock
   microengineer.eu
Please press IR for key SET
read ir-code: 0xa0
Please press IR for key 1
read ir-code: 0x00
...
Please press IR for key 9
read ir-code: 0x08
Please press IR for key 0
read ir-code: 0x09
...
```

The clock can then be switched to the setting mode using the taught SET key and the actual time can be set using the numeric keys. The codes are not lost in the event of power loss as they are permanently stored in the microcontroller. A Youtube video [11] shows this procedure.

Superstructure

The project consists of two circuit boards plugged together at a 90° angle and secured to the tube (**Figure 9**). One board (**Figure 10**) contains the control electronics, the other only holds the VFD tube. The control board can also be assembled manually if desired, using solder paste and hot air. This requires a very steady hand holding a sharp pair of tweezers to assemble the transistors in their tiny SC70 housing. The project repository [10] also contains a file for a solder paste stencil among the Gerber files.

You have to be a little careful with the

Commands

The CMD field at address 0x03 specifies one of three commands:

CMD	Description	Parameter s (1 byte each)
0x01	Set time	(hh) (mm) (ss)
0x02	Switch display mode	(mode)
0x03	Set IR code	(index) (code)

x01: Set time

The command consists of three bytes — the values for hours (hh), minutes (mm), and seconds (ss).

0x02: Switch display mode

This command can be used to switch the display between time and raw segment data, which are transferred to the microcontroller in the RAWDATA data area. Each byte in RAWDATA corresponds to one digit on the VFD display (from left to right) and is binary coded (see Figure 2). For example, the letter A would consist of the segments a, b, c, e, f, g, corresponding to 119 or 0x77.

You can use this function to create tickers. An example program (written in Python) can be found in the project's GIT repository.

0x03: Set IR codes

Using command 0x03, IR codes from a remote control can be stored in the microcontroller EEPROM. An index key and the corresponding IR code are always transferred. The following indexes are supported:

0x00: SET key

0x01 to 0x0a: Keys "1" to "9" and "0".

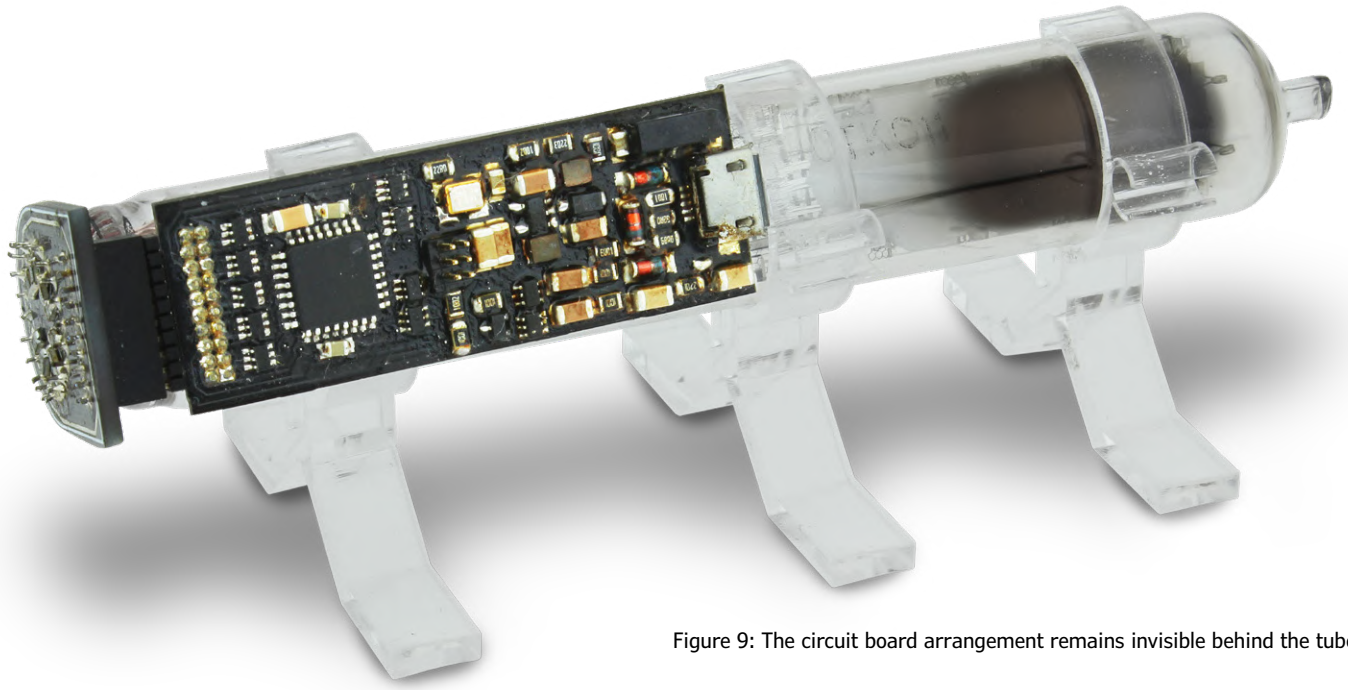


Figure 9: The circuit board arrangement remains invisible behind the tube.

IR sensor TSOP752 [7], which can be mounted lightly twisted. This can be installed with either orientation; here it is mounted horizontally to receive IR signals perpendicular to the board surface. Before the tube can be fitted on the mounting board (**Figure 11**), the eight resistors must be soldered in place. The picture shows pin 7, which is particularly easy to find on the tube since it is the middle of three connections not connected in the tube (see also [8]).

As a small extra there are mounting feet for the tube (**Figure 12**), which can be either laser-cut out of 8 mm thick acrylic (see DXF file), or 3D-printed (see STL file). The left holder differs from the right one only by a small clamp for the USB cable. In the holders, the tube can be tilted a few degrees up or down to perfectly adjust it to the viewer's viewing angle. Correctly adjusted, the control electronics vanish completely behind the tube and enable the uniquely minimalist look.

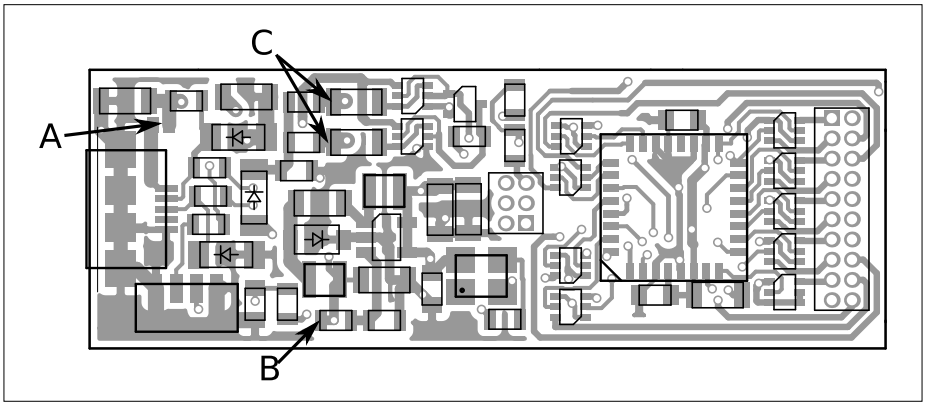


Figure 10: Layout of the control board.

Commissioning

After the boards have been assembled, a jumper must first be soldered closed (A in Figure 10). For the following measurements, the small circuit board with the tube should not yet be plugged in so that the filament cannot be destroyed in the event of a fault.

After the board has been powered via USB, the output voltage of the voltage inverter for the anode voltage must be about –30 V at point B. The anode voltage must be at the same level as the anode voltage.

After flashing the microcontroller, an alternating voltage with an effective value of approximately 3.5 V can (hopefully) be measured between the two C-points to power the tube filament.

If all voltages have been successfully tested, the holder board can be plugged onto the control board at a 90° angle. The unassembled side of the circuit board should point towards the tube.

Software

Linux is recommended as an operating system. In order to transfer the micro-

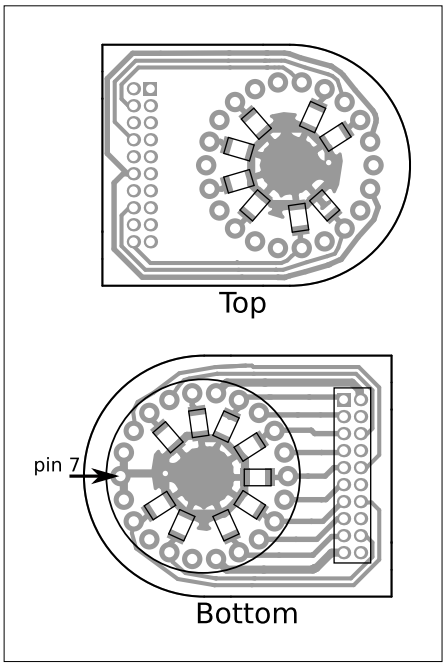


Figure 11: Layout of the tube holder board.

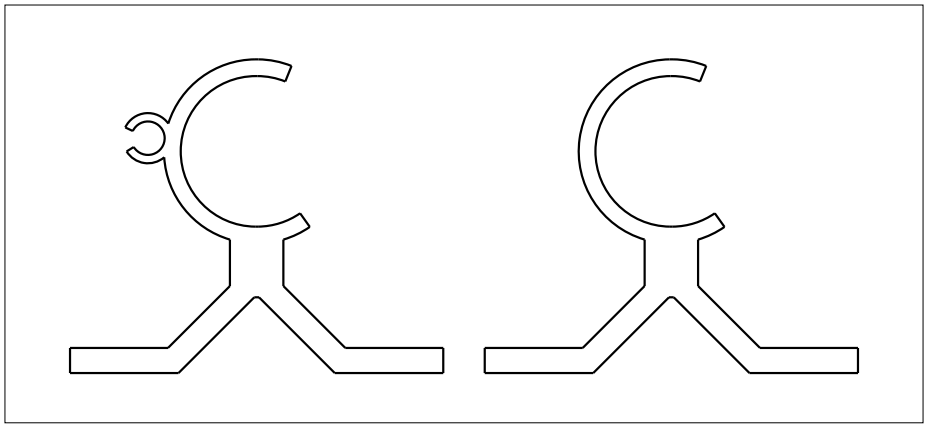


Figure 12: OMG: sockets!

controller software into the controller, some packages are needed for compiling and flashing (for example for Ubuntu). They are easy to install using:

```
sudo apt-get install gcc-avr make
   avr-libc avrdude
```

The software can then be compiled from the repository and flashed to the ATmega88:

```
git clone https://gitlab.com/
   vfd-clock/avr
cd avr
make hex
make fuses
make flash
```

The Makefile is configured to use a USBASP-V2 programming adapter which by itself uses V-USB :-).

@ WWW.ELEKTOR.COM

→ VFD Tube clock with ESP32 DevKit-C (no. 18455)
www.elektor.com/18455
(Note: This is not the VFD Tube Clock described in the article but an alternative project tested by Elektor Labs.)

→ VFD-Shield for Arduino (no.. 17244)
www.elektor.com/17244

→ Acrylic glass housing for VFD-Shield for Arduino (no. 17385)
www.elektor.com/17385

Although the pinout of the programming connector on the control board is 6-pin and arranged according to the standard, the pitch has been reduced from 0.1 inch to 0.05 inch mm for space reasons. An adapter board design is also located in the repository.

Linux

There are some example programs in the repository showing how to address the VFD clock through Python. But first you need to install pyusb:

```
sudo apt-get install
   python-setuptools
easy_install pyusb
```

Then the well-known 'Hello Elektor' sample program should work:

```
sudo python scroll.py “hello
   elektor“
```

An example video with the alphabet can be seen under [9].

Windows

The Python scripts also work with Windows (tested under Windows 10 and on a Windows 7 virtual machine). However, the installation instructions for the Windows version are somewhat more extensive, which is why only the documentation in the project repository [10] is referred to here. ⏮

(170510-02)

Web Links

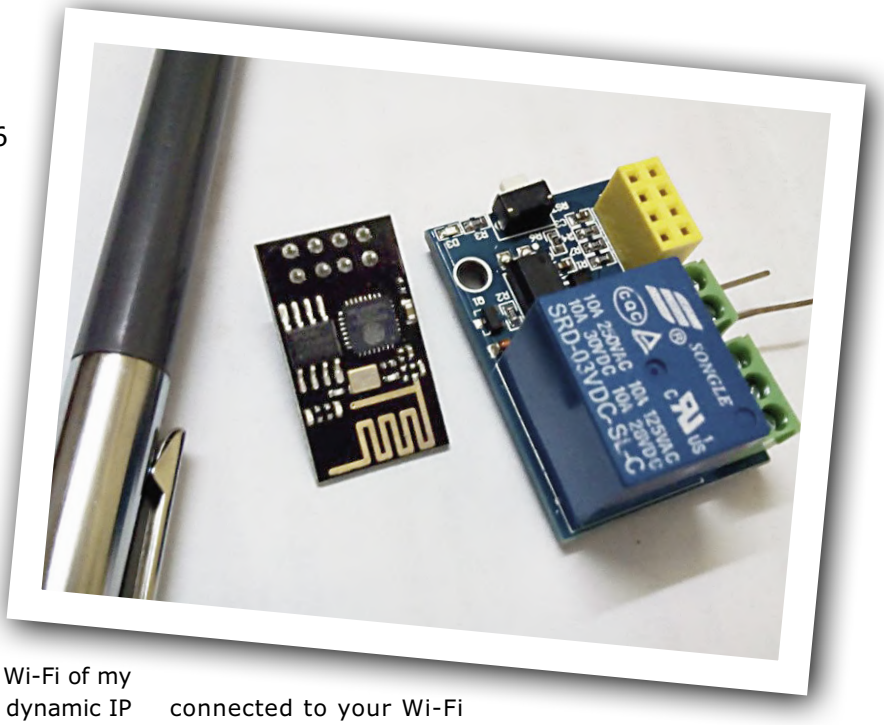
[1] Russian desktop calculator: www.leningrad.su/museum/show_calc.php?n=182
[2] IV-18 Vacuum fluorescent display: http://www.jogis-roehrenbude.de/Roehren-Geschichtliches/Nixie/IW18/IW18.pdf
[3] VFD control: www.noritake-elec.com/technology/general-technical-information/vfd-operation
[4] V_USB: www.obdev.at/products/vusb/index.html
[5] USB firmware for AVR: http://vusb.wikidot.com/hardware
[6] IRMP: www.mikrocontroller.net/articles/IRMP
[7] IR receiver datasheet: www.vishay.com/docs/82494/tsop752.pdf
[8] IW-18 brief data: www.jogis-roehrenbude.de/Roehren-Geschichtliches/Nixie/IW18/IW18.pdf
[9] Video on scrolling text: www.youtube.com/watch?v=fJh3KZqygx4
[10] Projekt Repository: https://gitlab.com/microengineer18/vfd-clock
[11] Video time setting: www.youtube.com/watch?v=s-MpQ9E_qpg

A 3-Dollar Tiny Wi-Fi Switch That Works

Creative dealings with the ESP8266

Bera Somnath

Out of curiosity I ordered a tiny ESP8266 relay switch board from aliexpress.com, for next to nothing. Blissfully unaware, then, of the struggle ahead to make the thing work as advertised. |I survived and conquered — here's the story.



The little gizmo pictured in **Figure 1** was stated by the vendor(s) to contain an inbuilt switching program, so all I had to do is "use some Android phone to control it". Sadly, despite several attempts using various Chinese and English-language Android apps, nothing ever happened! Sure, the device gets connected with the Wi-Fi of my PC or laptop as a server and it also allocates the dynamic IP address — but nothing happens after that. No server page etc. opens anywhere! Utterly frustrated, one day I decided to develop and install my own program into its dumb head!

Starting out

I reprogrammed the switch to connect to my Wi-Fi router with a fixed IP address and a fixed port like these: 192.168.1.99 and port 8083. Once connected the board behaved as a webserver at http://192.168.1.99:8083 and the 'switch' was visible in the browser. Pressing the On or Off button made it act accordingly. For simplicity, I added a refresh counter and a relay status to have a readout for the state of the relay (that being, on or off).

Strategy

Next, in my DSL router I set up 'port forwarding' of the 8083 port to the 8083 port of the "192.168.1.99" computer (i.e. the relay board) which gets connected to it. Once you know the external IP address of your router, you can connect to it from the other side of the universe! So easy! But the question is, how to know the external IP address of your router? Simple, you ask your son, daughter or your spouse to open the router's admin page, go to the Status tab and locate the IP address of the router — it's a common feature of most DSL router modems out there. When your laptop, desktop, or mobile phone gets connected to your Wi-Fi network, it gets an IP address as the identity of that device on the Wi-Fi network. Likewise, your DSL router gets an IP address when it gets connected to Internet, and that's the address you have to know.

But there is no dearth of problems in Bera's world of electronics! The external IP address of any router changes quite often — up to 10 times per is not uncommon. So really, you'd need a permanent valet near the modem to shout numbers 10 times a day … you'd wish you could train your pet to do this work there and then!

But where there is a problem there is solution! At least in the world of electronics this is so true. When you can formulate and then simplify a problem, you are actually pushing towards the solution. Fortunately, most run of the mill DSL modem haves an



Figure 1: Wi-Fi controller relay switch board as purchased from aliexpress. Hardware: Pass; cost: Pass; software: div-by-0.

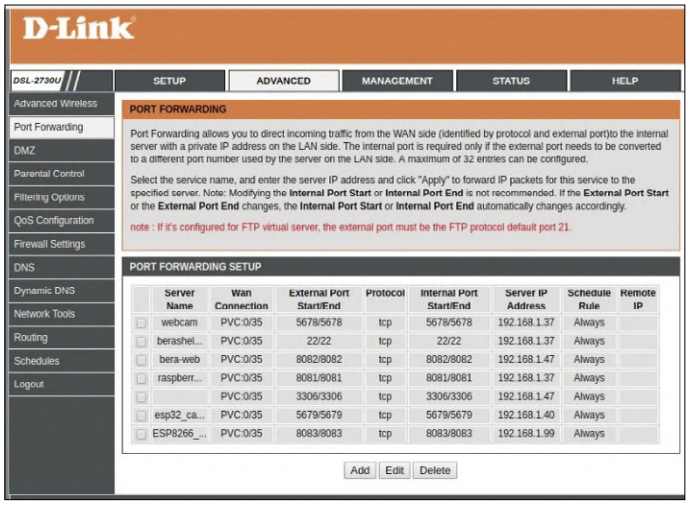


Figure 2a: Port forwarding on the D-Link modem.

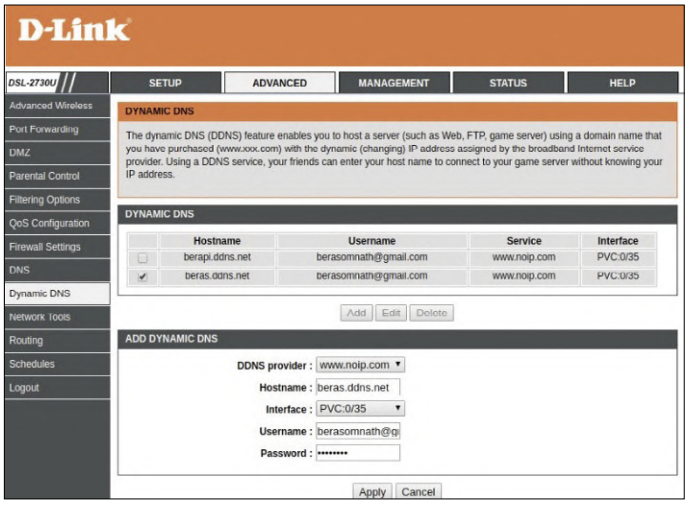


Figure 2b: Dynamic DNS-ing on the D-Link modem.

inbuilt DDNS (Dynamic DNS) tracking facility. You just supply the DNS host server's name, the DNS name and the remaining authorization details, and the Modem will track the changes of the external address on its own. You just need to remember the DNS name you have registered with the DNS host server's site and the port you are forwarding in the modem — in my case it's berapi.ddns.net:8083 from noip.com. See **Figures 2a and 2b** for my setups.

Architecture

The basic ESP8266 — also called ESP-01 — has 8 pins coming out of the chip board (**Figure 3**). But if you look carefully, the chip has total of 32 pins. That means for simplicity, on this board the other pins are not used by the manufacturer. The 5-volt relay is connected to GPIO0 pin through a switching transistor. When the GPIO0 drops Low, the relay is actuated and conversely, it's deactuated when GPIO0 goes High. The board has a common 3.3-V regulator chip installed and happily accepts 7–9 VDC on its Vcc pin.

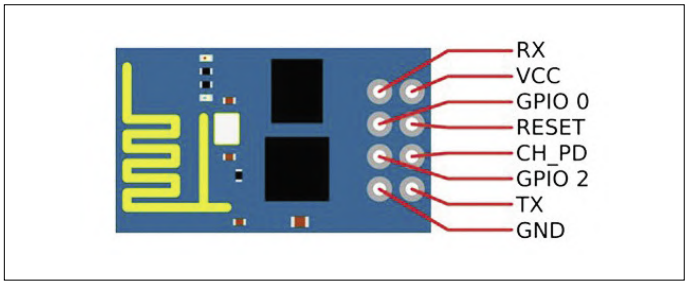


Figure 3: The basic ESP8266 (ESP-01) has 8 connections bonded out on pins.

Programmer

This is really the easiest part of the project. You just need a programmer to program this tiny little wonder! The programmer is just another way to transfer the Arduino sketch from the computer's USB port to the ESP8266's 'chip brains'. FTDI's CP2102 board is available at aliexpress for $1.02 only. In case

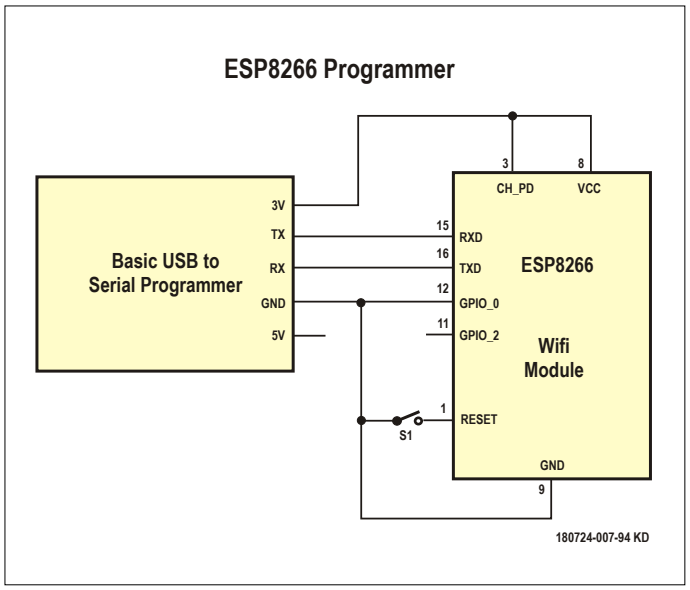


Figure 4. Sketch-uploading using the FTDI USB-to-Serial tool.

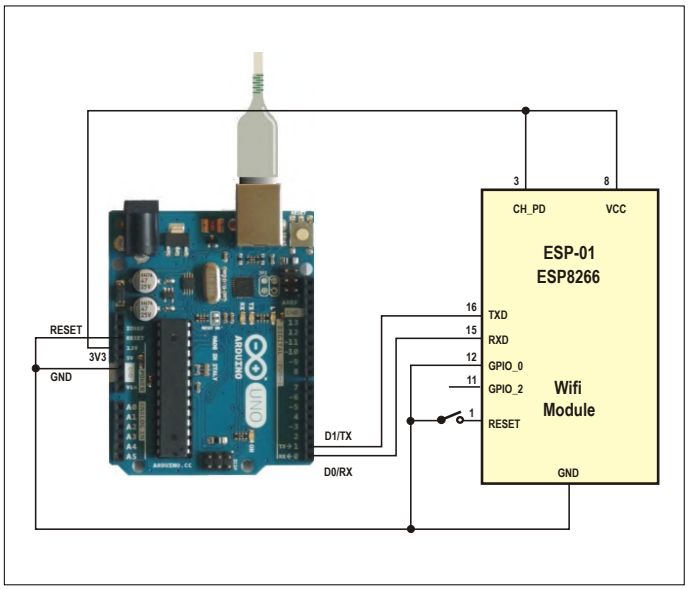


Figure 5: As an alternative to Figure 4: using an Arduino UNO board.

you have a working Arduino UNO board available around, you can also use it to upload your sketch to the little wonder. The connections for either type are shown in **Figures 4** and **5**. In fact, when you tie the Arduino's Reset pin to ground you are just making the Arduino UNO board act as an USB-to-serial uploader like the FTDI CP2120!

We have provided a Push-to-On switch in both cases, which needs to be depressed for about 10 seconds while uploading the sketch. After 10 seconds, release it and the sketch will upload in another 60 seconds. All the lights on the FTDI board will blink while the sketch is uploading.

Program output

The program sketch is available as a free download from the Elektor magazine website [1]. Replace the Wi-Fi ID and password with your Wi-Fi ID and password and then upload the sketch by pressing the 'Upload' command within the Arduino IDE (go to Sketch Upload).

Figure 6 shows the output of this little wonder, when the sketch got uploaded and the DDNS & port forwarding was done properly. By calling this url in any browser window be it on an Android or Apple phone, anywhere of the world, the relay will behave as per the button press.

A Bill of Materials, and prices

Prices from aliexpress.com and correct at the time of writing.
FTDI USB to Serial, CD2120: $1.02 (optional)
WiFi ESP8266 Relay: $1.85
Grand Total: <$3.00 (excl. 5-V power supply)

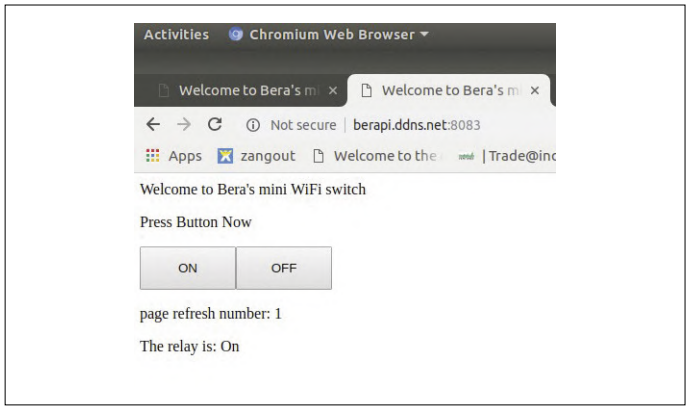


Figure 6. The author's mini relay switcher in action on a homespun web server.

Aftermath

The only thing left to now is to make the connection secure, otherwise anyone aware of the switch URL will be able to operate the relay, with dire consequences. I know you will start to explore that right now. Best of luck.

180724-01

Web Link

[1] Project software: www.elektormagazine.com/180724-01

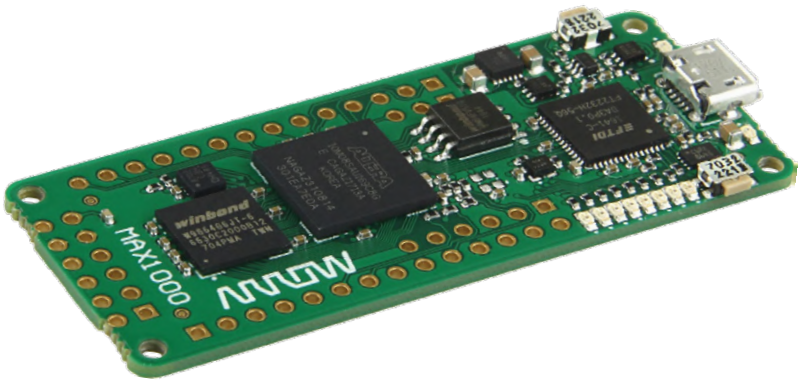
Advertisement



The SCCC Project (5)

Homebrew soft-core processor and C compiler

By **Martin Oßmann** and **Mathias Claussen**



In this the final installment of our series will be pushing the FPGA (almost) to its limit, receiving weather reports on 147.3 kHz, decoding them, and displaying them on a VGA monitor. As previously, our DIY CPU inside the FPGA will be at the heart of the experiment.

In experiment 8 we will demonstrate the full extent of what can be implemented within a MAX10 FPGA. The DDH47 weather service broadcaster transmits weather reports from Germany using frequency shift keying (FSK) on 147.3 kHz. The experiment will include a look at how to receive this information and display it on a VGA monitor. The signal from the antenna is connected to the 'Analog In' input (see **Figure 1**), which means that jumper JP1 should not be fitted.

Figure 2 shows a block diagram of the receiver. The yellow blocks are implemented in the FPGA, while the CPU is responsible for the functions in the green blocks. The receiver follows the classical SDR architecture: the input signal is mixed with the outputs of a local oscillator (or LO), creating in-phase (I) and quadrature (Q) components. These are passed through low-pass filters which serve to attenuate signals from other transmitters. The low-pass filters are implemented as CIC (cascaded integrator-comb) filters, which do not require the use of multipliers. From the low-passed in-phase and quadrature signals a CORDIC block computes the phase and amplitude of the received signal. Then there is a further low-pass filter and we obtain a clean phase signal. By taking differences between consecutive phase values we obtain the instantaneous frequency, which is the demodulated signal that we require. At this point the CPU takes over: it samples the instantaneous frequency at 500 samples/s, outputting the results using a 5-bit software UART. These characters are then converted from Baudot to ASCII and passed on to a VGA driver.

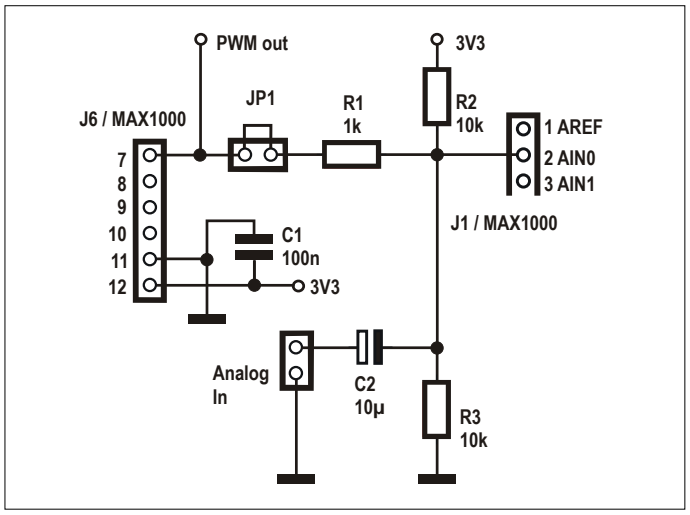


Figure 1. Signal outputs.

Now it is back the the FPGA to generate the VGA signal itself. The VGA monitor is connected as shown in **Figure 3**.

As usual, you can synthesize *experiment8.qpf* using Quartus, and compile *experiment8.c* and upload the code using Processing. The author used a mini-whip antenna, which gave trouble-free reception. While the project was being developed the author made extensive use of the dual DAC outputs in order to monitor the signals inside the FPGA 'live': see **Figure 4**. In the current version the frequency is output on channel 1 and the phase on channel 2. You can see the characteristic RTTY frequency shifts on channel 1 in the figure. The phase value rises or falls over the course of a single bit period at a rate which is proportional to the frequency offset; the phase jumps discontinuously when it wraps around by a complete cycle.

VGA output

The VGA output function is implemented as follows. First a divide-by-2 circuit generates the 25 MHz VGA clock from the 50 MHz FPGA clock. Strictly speaking we should generate 25.175 MHz, but this small deviation is not important. This 25 MHz clock is used to drive the horizontal counter `Htimer`. This counter has a period of `h_period`, which is a parameter that we can subsequently set from the CPU: normally we will set the period to 799 clocks. The reload rate of this counter is therefore 31250 Hz, which is a standard line rate for a VGA signal. The output of this counter is used to clock the vertical counter `Vtimer`, whose period is `v_period`. Again we can set `v_period` from the CPU, normally to 524. The resulting vertical frequency

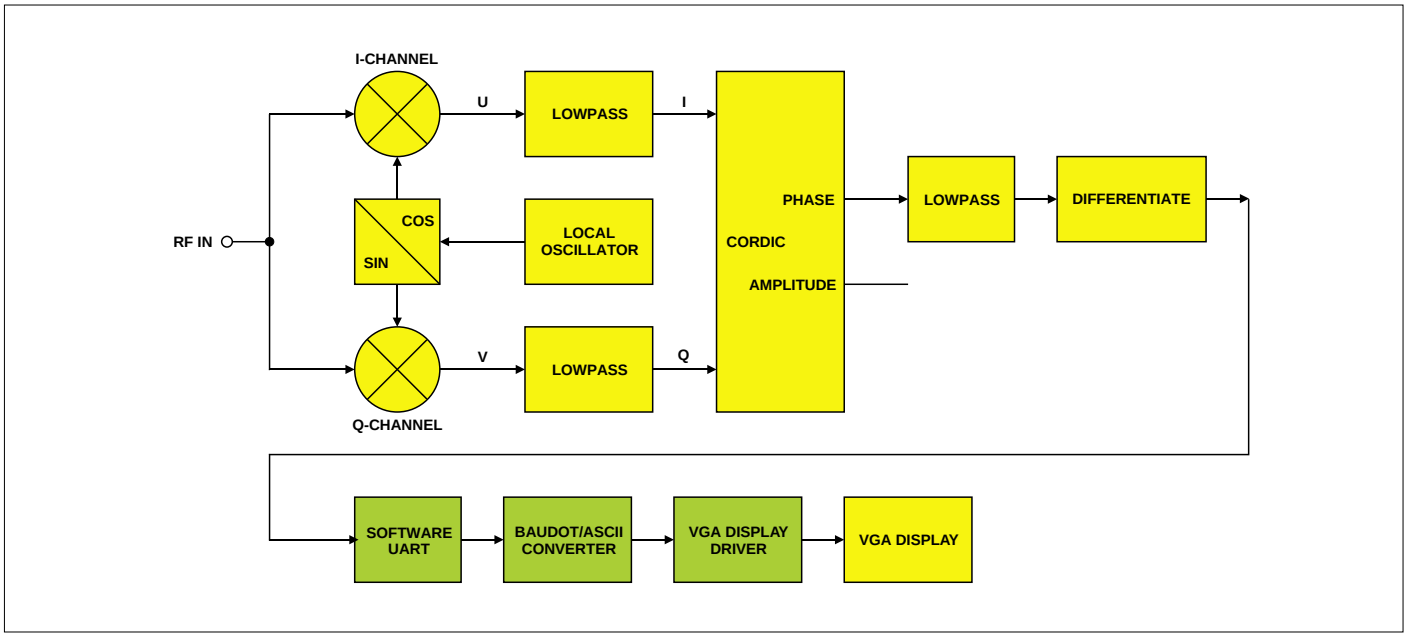


Figure 2. Block diagram of the SDR receiver.

is then 31250 Hz / 525 = 59.523 Hz.
The Verilog code below shows how the h_sync and v_sync synchronization signals are generated. The h_sync signal is active when the value of *Htimer* is between h_sync1 and h_sync2; similarly, the v_sync signal is active when the value of *Vtimer* is between v_sync1 and v_sync2.
The Valid signal is active when the current point in the scan is in the visible area of the screen: this extends from h_valid1 to h_valid2 in the horizontal direction and from v_valid1 to v_valid2 in the vertical direction.
These parameters therefore determine where the active area is positioned on the screen.

```
always @(posedge clock)
  begin
  vga_h_sync   <= ~((Htimer[10-1:0] >= h_sync1 ) &&
   (Htimer[10-1:0] <= h_sync2 ));
  vga_v_sync   <= ~((Vtimer[9:0]  >= v_sync1 ) &&
   (Vtimer[9:0]  <= v_sync2 ));
  Valid       <=  ( (Htimer >= h_valid1) && (Htimer
   <= h_valid2) && (Vtimer >= v_valid1 ) && (Vtimer
   <= v_valid2) );
  end
```

The timing relationships are illustrated in **Figure 5** and the parameters are set as follows.

```
  vgaSet_h_sync1(640) ;
  vgaSet_h_sync2(736) ;

  vgaSet_h_valid1(4) ;
  vgaSet_h_valid2(625) ;
  vgaSet_h_period(800) ;

  vgaSet_v_sync1(484) ;
```

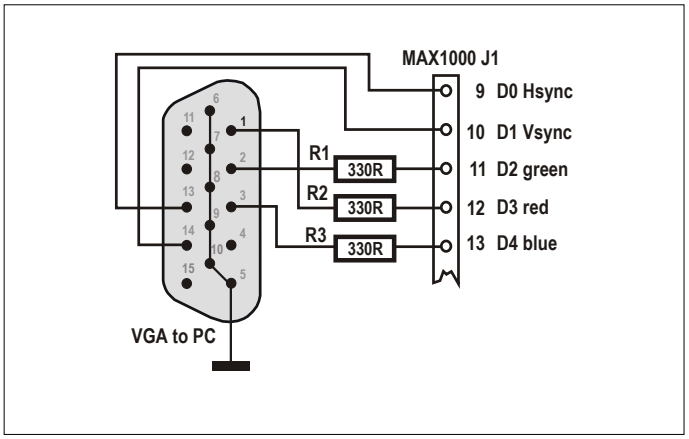


Figure 3. Connecting a VGA screen.

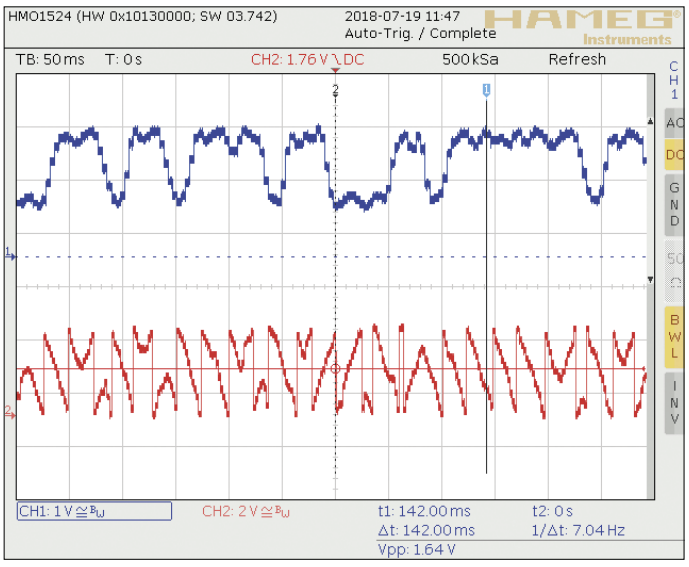


Figure 4. Phase and frequency plot of the RTTY signal.

```
  vgaSet_v_sync2(485) ;
  vgaSet_v_valid1(0) ;
  vgaSet_v_valid2(522) ;
  vgaSet_v_period(524) ;
```

The character generator `cgRom` is loaded from an external table as follows.

```
// character generator rom:

localparam cgRomPatternWidth = 12 ;
localparam CGromAdrWidth = 11 ;
reg  [ cgRomPatternWidth-1: 0] cgRom
   [0:(1<<CGromAdrWidth)-1] ;

initial begin
// Very important: set device option to: 'Single
   Uncompressed Image with Memory Initialization'
  $readmemh("CGrom128x12x16x1v01.txt",cgRom) ;
  end
```

The characters to be displayed are stored in a RAM called `vRam`, which is accessed as shown in the code below.

```
  always @ (posedge clock) begin
  vRamReadData <= vRam[vRamWriteAdr] ;
  if(vRamWriteEnable) begin
    vRam[vRamWriteAdr] <= vRamWriteData  ;
    end
  end
```

In the CPU we are now able to implement simple terminal output.

```
int vgaXpos,vgaYpos ;

moveUp(){
  int x,y,c ;
  for(y=1 ; y<28 ; y++){    // move up 27 rows
    for(x=0 ; x<51 ; x++){  // 51 chars per line
         c=vgaXYread(x,y+1) ;
         vgaXYwrite(x,y,c) ;
      }
    }
  }

vgaCrlf(){
  int x ;
  vgaXpos=0 ;
  vgaYpos++ ;
  if(vgaYpos>28) {
    moveUp() ;
       for(x=0 ; x<39+12 ; x++) { vgaXYwrite(x,28,'
   ') ; }
       vgaYpos=28 ;
       }
  }
```

```
vgaPutChar(int c){
  if(c==13){ vgaCrlf() ; return ; }
  if(c==10){ vgaXpos=0 ; return ; }
  vgaXYwrite(vgaXpos,vgaYpos,c ) ;
  vgaXpos++ ;
  if(vgaXpos>39+12) { vgaCrlf() ; }
  }
```

For the DDH47 application the demodulated received signal must be converted into Baudot-coded characters using a software UART. That is done by the following code, which runs on the CPU.

```
#define uartFullTime 10
#define uartHalfTime 6

checkForStartBit(int input){
  if ( input != 0) {
    uartTimer=uartHalfTime ;
    uartBitCount=0 ;
    uartShiftReg=0 ;
    }
  }

uartSample(int input){
  if (uartTimer==0) {
       checkForStartBit(input) ;
    }
   else {
    uartTimer++ ;
       }

  if (uartTimer>uartFullTime) {
    // a full bit time has elapsed again
    uartTimer = uartTimer-uartFullTime ;
    // shift input into shiftregister
    if ( input == 0) {
         uartShiftReg=uartShiftReg+(1 <<
   uartBitCount) ;
         }
    // count bits
    uartBitCount++ ;
    if (uartBitCount==6) {
      // we have enough bits
      uartShiftReg=uartShiftReg >>1  ; // eliminate
   start bit
      BaudotPrint(uartShiftReg) ;
      }
    if (uartBitCount>6) {
      // too many bits, restart
      uartBitCount=0 ;
      uartTimer=0 ;
      }
    }
  }
```

The conversion from Baudot code into ASCII is done as follows.

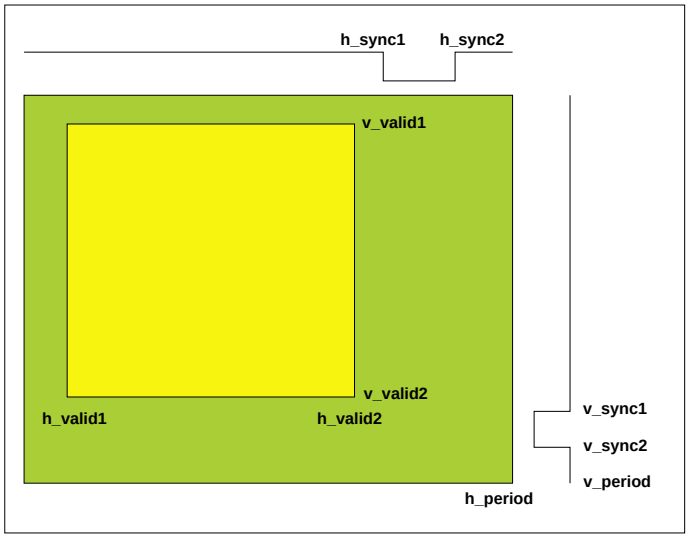


Figure 5. Timing relationships in the VGA signal.

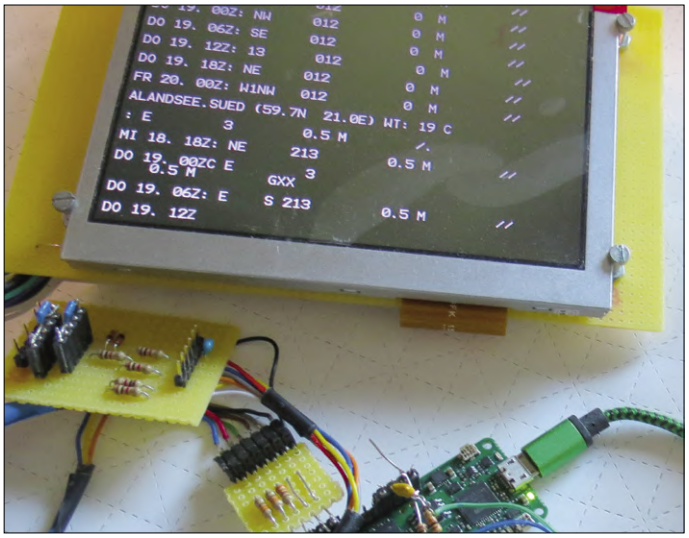


Figure 6. Weather information display.

```
BaudotPrint(int baudotChar){
   int AsciiChar ;
   AsciiChar= Btable[baudotChar+(baudotShift<<5) ] ;

   if ( AsciiChar==131) {
     baudotShift=0 ;
        AsciiChar=0 ;
        }
    else if ( AsciiChar==130 ) {
        baudotShift=1 ;
        AsciiChar=0 ;
        }
   if (AsciiChar!=0) {
     // display only printable characters
     putChar(AsciiChar) ;
        vgaPutChar(AsciiChar) ;
        }
  }  }
```

This experiment (see **Figure 6**) uses 70 % of the logic elements and 80 % of the RAM in the FPGA. So there is a little room to spare, but nevertheless we are making good use of the resources available. With a faster external ADC it would be possible to extend the SDR idea to work at higher carrier frequencies.

Experiment 8(b): VGA and prime numbers

Not all readers may be interested in receiving transmissions from DDH47, and so in the download at [2] there is a further mini-experiment, in which the CPU calculates prime numbers, with output over the VGA port. And along with that download the author has added a number of bonus demonstration projects which may provide further inspiration.

- *demo1* Example using the RTC and RXrdy interrupts.
- *demo2* How to connect an LCD panel to the MAX1000.
- *demo3* Measuring instruction timing with the help of the RTC.
- *demo4* Display of stack use in a recursive calculation.
- *demo5* Test snippets for the compiler and the CPU.
- *demo6* Examples showing how the compiler translates code.
- *demo7* Demonstration of the output options of the compiler.

And so we come to the end of our series of articles on the SCCC project: now it's over to you! You might want to add a CPU to your own FPGA projects, or perhaps expand the CPU with new peripheral modules. Or, for the really adventurous, you could even look at extending the compiler. ⏮

180394-E-02

@ WWW.ELEKTOR.COM

→ MAX1000 FPGA development board
www.elektor.com/max1000-iot-maker-board-8kle-8-mb-ram

→ E-Book: 'Microprocessor Design Using Verilog HDL'
www.elektor.com/microprocessor-design-using-verilog-hdl-e-book

Web Links

[1] The SCCC Project (1), ElektorLabs magazine 2/2019: www.elektormagazine.com/magazine/elektor-88/42444

[2] Project page accompanying this article: www.elektormagazine.com/180394-E-02

1-30 MHz Concept Receiver

AM/SSB/CW, with plug-in resonators, filters or crystals

By **Kostas Giannopoulos,** SV3ORA (Greece)

A 'concept circuit' is not a common thing among homebrewers. Most of them usually build classic circuits that satisfy their needs without further experimenting. Occasionally someone comes up with an unusual idea that is worth trying out. When this idea works, it becomes a building block in future homebrew gear. The radio receiver presented in this article is such a concept circuit. It allows for broadband single signal (USB) reception while avoiding the extra complexity posed by circuits with similar features. Sure, a modern receiver would employ a direct conversion phasing-type front-end circuit topology (as with SDR) but is sure to require a fair amount of money and time to complete. The alternative presented here can be built for little money or with parts already available in the junk box. Its features should not be underestimated though.



The reflex/regenerative receiver front end is simplicity itself, yet capable of remarkable performance, better than any simple HF band receiver I have built so far. The highly sensitive, single-frequency capable, all-HF-band receiver can be easily built in a few afternoons. It can receive any frequency on the HF bands (1-30 MHz) by just replacing the resonator for that frequency. No other components need replacing or rescaling apart from the resonator itself. The **head photo** shows the receiver along with a medication storage box to have the various resonators readily accessible to the 'operator'.

The receiver can work on **quartz crystals**, ceramic **resonators** with 2 legs or **ceramic filters** with 3 legs. The oscillator works with L/C combinations also (in place of the crystal) but this is something that has not been optimized yet. If ceramic filters are used, their centre pin can be grounded, or left unconnected. Note that crystals can achieve single-signal (i.e. single-frequency; single-channel; spot-frequency) reception within all bands but depending on the xtal properties the sensitivity drops compared to using ceramic resonators. Single-frequency reception using ceramic resonators was tested up to about 17 m successfully. With crystals, it works fine right up to the 10 m band. Note that some ceramic resonators may refuse to oscillate but most of them will, so don't be disappointed if the receiver does not work straight off with a particular ceramic resonator — just try another to see if it kicks in. With quartz crystals there is no such problem, but if you find the resulting sensitivity a bit low you can try decreasing the 10-k fixed resistor at the JFET source.

The receiver front-end

To come up with a circuit with such a low component count yet good performance requires thinking and going "the alternative way" all the time. **Figure 1** shows te result. Both stages of the receiver are used in more than one way. The 2SC9018 transistor acts as an untuned wideband RF preamplifier with RF gain control and an audio preamplifier at the same time. The J108 FET forms a self-oscillating regenerative detector.

RF signals arrive from the antenna at the 2SC9018 RF preamplifier where they are amplified and passed to the 4.7-pF and 100-nF capacitors. The small portion of the RF signal that passes through the 4.7-pF capacitor passes through the crystal to the self-oscillating regenerative detector J108. A small capacitance like 4.7 pF will not affect the tuning range much. Also, it isolates the oscillator from the RF preamplifier as well as lightly loads the oscillator tank circuit. This self-oscillating mixer 'blends' its self-generated signal with the incoming RF input signal and produces audio and some other frequencies. The strength of the regeneration is adjusted by the 100-k potentiometer. The regeneration potentiometer also acts as a frequency fine tuning control. Especially if ceramic resonators are used, zero-beating on an SSB signal might be tricky with just the frequency adjust potentiometer. To

keep cost down, a multiturn potentiometer is avoided. Fine tuning is performed by adjusting the regeneration potentiometer. The amount of frequency pulling caused by the regeneration potentiometer is small but adequate for zero beating.

The output signal (RF and detected AF) produced by of this stage passes through a 10-k isolation resistor, a 100-nF DC blocking capacitor and a 3.3-mH RF filter choke to the 2SC9018 transistor once again (when reflex action is switched on), where it gets amplified again. However, only the AF signal is amplified, RF being blocked by the choke at the base of the transistor. This choke also prevents RF from the antenna from passing to the regenerative detector. The amplified AF signal is then passed to the 4.7-pF and 100-nF capacitors and the other 3.3-mH choke. The 4.7-pF capacitor and the crystal after it, do not allow audio to pass and so return to the J108, hence most of the amplified audio is passed to the output of the receiver by way of the 100-nF capacitor and the series choke. The 100-nF capacitor also passes amplified RF signals from the antenna and the 2SC9018. Together with the choke in series with it, it keeps RF from passing to the audio output of the receiver. If you connect high-impedance headphones to the receiver or if you build the active AF filter shown further on in this article, you may remove this extra choke since the headphones cannot respond to RF anyway and the filter will suppress the RF. The value of both chokes is not critical, and you can use any values (even different values for both) provided they are in the low mH range. I used two cheap moulded 10-mH chokes I had available. Using a separate stage as an RF (and AF) preamplifier ensures isolation of the antenna from the self-oscillating detector. There is no LO (local oscillator) leakage to the antenna and no detector detuning either.

The output of the 'receiver' can be connected directly to high-impedance headphones. The volume is more than adequate when a big antenna is used and when reflex action is switched on. Alternatively, it can be connected to the microphone input of the 'radio shack' PC or laptop. There are numerous programs that allow further manipulation of the audio signals, as well as filtering and decoding to meaningful data, so the receiver does not need to be any more complex than that. However, you will need a good processor and sound card if you want to minimize latency due to the CPU-demanding DSP processing.

Practicalities and pitfalls

The circuit in Figure 1 was built many times over and it always worked. **Figure 2** show the working prototype. Different variations were built using tuning capacitors, varicaps, multiturn pots and or even LDRs! It always worked as expected. Note however that when the reflexion is switched on and under the presence of strong out of band (broadcast) signals, the B-E junction of the 2SC9018 will work as an AM detector! Effectively it is a diode across the antenna input, and it couples directly to the AF output without any filtering at all. This is the price you pay when introducing regenerative action in such a simple circuit. Bear in mind that the circuit has no RF front end tuned circuits so it is as wideband as can be. The good news is that the detected AM signals do not appear only when an external untuned antenna receives very strong broadcast stations and when the RF gain is set to maximum. On tuned or small antennas, the problem should not be noticeable. These detected AM signals usually appear at quite low levels at the AF output in comparison to the wanted regenerated signals. They are mostly noticeable when they exhibit high signal levels and when the regenerative detector is either not oscillating or when no ham band signals are tuned into. In practice they pose no significant problem in weak ham signal reception.

When AF filtering is used after the receiver (be it hardware or FFT) the AF bandwidth of these 6-kHz or so AM signals drops to 2.5 kHz or less, depending on your filter bandwidth. Using narrowband audio filtering (i.e. digital modes) these AM signals disappear due to the narrow bandwidth and associated low level. Both for wideband and narrowband, the RF gain control can help in reducing these interfering AM signals (but also the wanted ones).

Single-frequency reception

Single-frequency (a.k.a. spot-frequency; single-channel) reception is a feature seen on complex receivers only. Despite its simplicity, our receiver is also capable of single-frequency HF reception. The

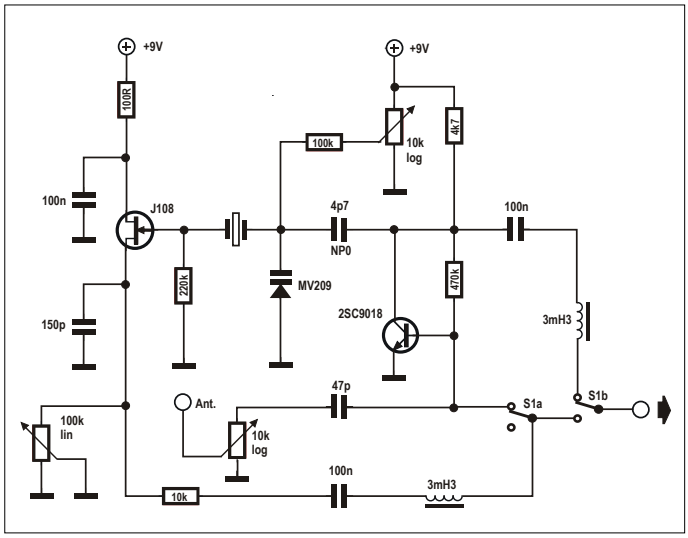


Figure 1: The receiver front-end is suitable for direct connection to the PC's microphone input, or high-impedance headphones.

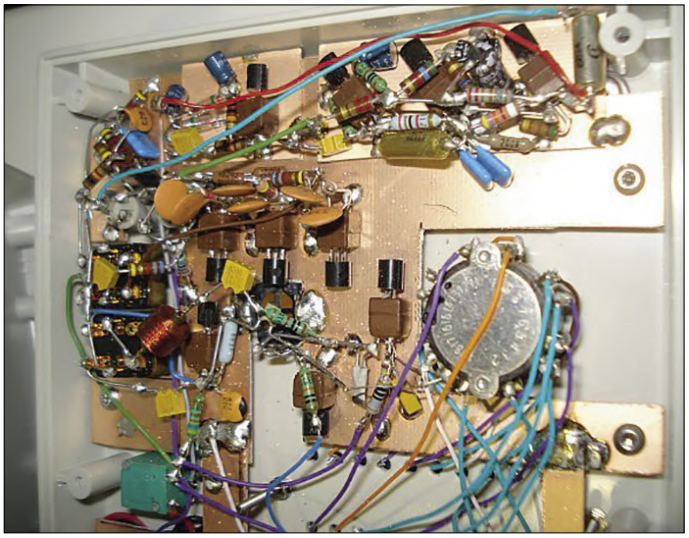


Figure 2: Practical construction of the RF front-end (two transistors near the selector switch) using 'dead bug' or 'flying lead' construction on a sheet of unetched circuit board.

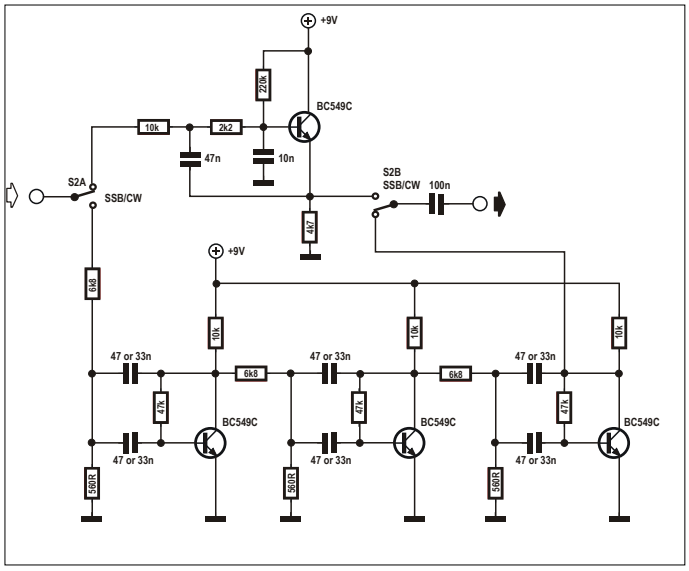


Figure 3: An optional audio BPF/LPF for CW/SSB, suitable for direct connection between the front-end AF output and the PC microphone input.

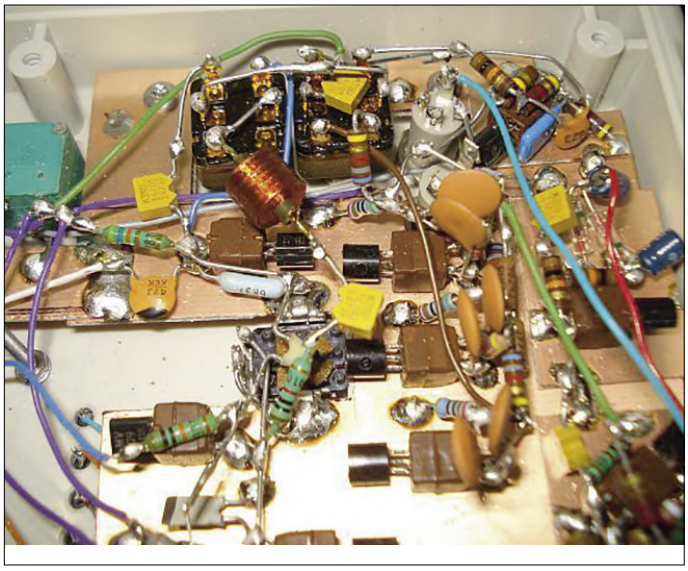


Figure 4: The three BJTs that compose the audio filter of the receiver.

regeneration setting is so accurate that it can be adjusted to be effective on one sideband only. To illustrate this better, assume a DSB signal is being received, i.e. both sidebands tone modulated. By careful adjustment of the regeneration control, a tone will be received on the USB and just a hiss on the LSB. The hiss will be like listening to an SSB modulated signal on an AM receiver. Not only that, but the received hiss on the LSB will be appreciably attenuated as a result of the lack of regeneration effectiveness on that sideband.

This feature has been tested using crystals and ceramic resonators, and single-frequency reception was achieved on all the HF bands using crystals, and up to about 17 m using ceramic resonators. There were no ceramic resonators above 17 m to test. This feature does not relate to crystal filtering of the incoming RF signal. It was only when the regeneration was set carefully near the point of oscillation (at each frequency setting) that the opposite sideband (LSB) was rejected. Actually, it is not rejected but attenuated, and the detector is not effective in 'beating' this sideband. This was verified by setting the regeneration a bit higher whereupon the opposite sideband got 'beaten' (i.e. its lower frequencies). When the regeneration was set even higher, the whole opposite sideband was beaten (i.e. DSB reception). The behaviour was the same on all bands. On the higher bands the setting was "touchier" than expected, but still perfectly usable. The impressive thing is that this can be achieved without the need for multiturn potentiometers, so no special parts are required. Now this is single-frequency reception on all HF bands, out of our ultra-simple detector! How much better can it be at this level of simplicity?

Reflex / regenerative modes

Up to this point, the receiver is described as being a combined reflex and regenerative circuit. With the use of a single switch (SW1) the reflex action can be switched off instantly, turning the receiver into a pure regenerative circuit. To notice the differences, it is nice to switch instantly between the two circuit configurations. Hence the switch was permanently installed there. In the non-reflex configuration, the output audio volume is lower but there is no significant problem arising from the B-E junction rectification anymore, even when the RF gain is set to maximum. That's because any rectified audio is filtered by the low value of the 4.7-pF coupling capacitor and does not pass to the regenerative detector or the AF output of the receiver easily. When tested with ceramic resonators, switching the reflexion on or off, causes some instant frequency pulling, apart from a change in the audio volume. Also, when the reflexion is switched off, changes in the RF gain cause minor frequency pulling. Fortunately, the RF gain is not something that the listener would want to change too often. When the reflexion is switched on, there is no frequency pulling at all when altering the RF gain.

Tuning: capacitor or varactor

The effect of a variable (or 'tuning') capacitor in the frequency pulling of the crystals in a circuit like this is well known and it has been demonstrated in many articles. Indeed, a pure variable capacitor (not a varactor) that was initially used to set the frequency, had the greatest pulling effect when it set at the lower capacitance values, and its minimum capacitance value is important. This makes tuning more difficult at these settings, also because of stray capacitance that now has a greater effect in frequency pulling. If you wish to avoid decade capacitors and lots of switches, two things can be done to restrict the problem. One is to use a variable capacitor with a reduction gear and an offset shaft position. The first allows fine tuning and the second, reduces frequency (dial) nonlinearity by positioning the shaft and the plates in a way that produces a linear relationship between shaft angle and resonant frequency, which can be used to compensate for any non-linearity of the frequency dial of the receiver. The other way is to use a varactor diode which has a capacitance/voltage curve that compensates for the non-linearity, together with a logarithmic potentiometer to control it.

Fine tuning of the varactor may be done by a multiturn trimpot (if a logarithmic one could be found), or an ordinary logarithmic potentiometer combined with a large diameter knob. Just to mention that some of these cheap plastic tuning capacitors salvaged from old radios may present quite low minimum capacitances (under 10 pF) and relatively high maximum values (150-250 pF or so). These values result in a favourably large capacitance range, plus they are not detuned by control voltage instability like varactors do.

However, apart from improving the tuning linearity, the use of the varactor has also other important advantages. It avoids stray capacitance caused by hand effects, obviating the need for RF shielding. It also avoids microphonics, it has a small size and comes very cheap. A good quality, humidity and dust-free potentiometer must be used for controlling it reliably. Even cheap potentiometers can do a pretty good job though in comparison to a cheap variable capacitor. Remember, minimum capacitance is important for stronger frequency pulling, so a varactor with its low minimum capacitance and relatively high maximum is preferred, but don't go for more than 150 pF of maximum capacitance or so, as higher values will yield a small effect on frequency pulling. Most expensive 500-pF varactors have only 20 pF or so of minimum capacitance and the pulling range above a certain point is small. Varactors may not be found in the junk box. In such cases, an alternative is the EB junction of a BJT or a common 1N4001 to 1N4007 series diode, reverse biased of course. Their pulling range is smaller, but they can be easily found. LEDs should not be used, unless they are shielded from ambient light.

Varactor control

In the SV3ORA receiver, the cheap MV209 varactor is used together with a 10-k logarithmic potentiometer to control it. This combination gives excellent results. The tuning linearity is remarkably good at all frequencies on crystals and ceramic resonators. The tuning range for the crystals tested, is about 2-3 kHz or so above 40 m, less at 80 m and almost none at 160 m. The tuning range of a 40 m ceramic resonator was about 60 kHz and at 80 m about reached 30 kHz or so. The more expensive type 1SV149 (500-pF) varactor, despite its much higher capacitance range, improved the tuning range of the 40 m ceramic resonator by only 10 kHz or so. However, the tuning linearity suffered at the middle settings of the potentiometer. Thus, it was decided to use the MV209 varactor with the 10-k logarithmic potentiometer.

Correct polarity of this potentiometer is important. Do not reverse its connections as this will lead to a much worse tuning linearity than that of the varactor alone. The correct polarity can be found by placing the receiver (fitted with a crystal) near a commercial SSB receiver. Then set the logarithmic potentiometer at middle position and zero-beat the commercial receiver to the receiver's local oscillator. Then turn the potentiometer shaft all the way to the right and then all the way to the left. If the pot is correctly connected, the tone that heard from the commercial receiver at both extremes of the potentiometer should be roughly of the same frequency. If not, reverse the connections of the ground and the VCC wires to it. Another way to find out the correct polarity of the potentiometer is to watch the local oscillator leakage signal of the receiver on a commercial SSB receiver. With the logarithmic potentiometer correctly connected, the frequency decrease when you turn the potentiometer clockwise.

Correct polarity of the 10-k (log) RF gain potentiometer is also important. Connect it so that the RF gain varies in equal steps along the wiper movement. If it varies most of the RF gain when the wiper is turned slightly, its outer leg connections must be reversed.

Optional audio BPF/LPF filter for CW/SSB

To reject nearby signals on CW, a suitable audio BPF can be added, see **Figure 3** for the circuit diagram. One could argue that the PC sound card in combination with suitable software makes an audio filter instead. However, apart from the need for a computer to be carried around, there is another drawback in this: latency. The software needs a lot of processing power to do the job in a short time. In older or lower end computers such processing power is not available and time elapses from the instant the audio signal enters the sound card to the instant the processed sound exits to the headphones. In some cases, the delay amounts to several seconds. Tuning the receiver to signals is a nightmare if the latency is high. A pure analogue hardware filter has zero latency.

Quite a lot of CW audio filters have been tested, LC types, regenerative types and so on. LC filter types required quite a few hard to find inductors for the filter to be effective. The regenerative filters tried in practice required quite a few critical adjustments to operate effectively. Moreover, none proved to be stable enough. Upon signal strength variations, they all oscillated violently. The CW filter that gave the best performance and stability and did not require any adjustments is the multiple feedback band pass filter presented here. It is composed of three identical cascaded filters. The six CW filter capacitors define the centre frequency of this filter. If 47-nF capacitors are used, the centre frequency will be close to 700 Hz. Using 33-nF capacitors gives an approximate centre frequency close to 1 kHz.

SW2 selects between CW and SSB filters. The SSB filter does not have to be more complex than a single stage, and this filter starts to roll-off from about 2 kHz. The CW filter with less than 250 Hz bandwidth yields excellent selectivity for its simplicity without any tendency to oscillation. To get an idea of its selectivity, it is suitable for selecting just one out of the two tones of an RTTY (telex) signal. Strong CW signals very close to the passband can still be heard in the background but are intensely attenuated and do not affect the wanted signal reception, even if its strength is lower than these. This proved to be a desirable feature when scanning the band for signals, allowing you to find out what is going on nearby without having to retune the receiver frequency. By comparison, the narrowband filter of many commercial rigs requires very slow tuning so as not to accidentally skip signals when scanning, which has proved to be inconvenient when scanning large portions of the band.

Figure 4 shows the practical realization of the BPF/LPF.

Optional AF amplifier with automatic limiter

The optional AF amplifier presented in **Figure 5** has been tested in conjunction with the receiver with great success. It provides a gain of about 100 dB or more, which can be thought at first as too much for this receiver. However, because of the considerable variations in the input signal level, there are many occasions where

high gain is needed, and this amplifier provides it. For example, some of the tested crystals had very high losses and the reflex switch had to be switched on, adding another audio preamplifier into the AF chain. Ideal signal reception is accomplished with combinations of the settings of the reflex switch, the RF gain, the regeneration, and the audio volume controls. There are times where it is preferable to switch the audio gain to high and limit the RF gain. At other times, the opposite is preferred, depended on the signal strength and the noise. But that is the beauty of this receiver, the operator has full control over these parameters, even the way detection is accomplished.
The AF amplifier has been initially thought to be the easiest part of the receiver, but surprisingly this was not as straightforward as it seems. There are various reasons for this. First, the receiver front-end audio output volume varies significantly, depending on the signal strength, the way the regeneration is set, and whether you use crystals or ceramic resonators. Then there were switch "clicks" from the reflex and the filter switches, as well as noise from the regeneration potentiometer, when regeneration exceeds the limits due to DC on the wiper. Additionally, the high-gain amplifier was found to oscillate violently at high input signal levels. Normally, if a speaker amplifier is to be used, the listener should not worry about these things.
However, when headphones are used, the situation will be completely different. All those noises from the potentiometers and the switches, when boosted in this high-gain amplifier, can be deafening while trying to dig out weak signals. The amplifier volume must be set on high gain when trying to receive these faint signals. This causes the amplifier to oscillate and produce deafening sounds again. Another thing that matters is noise. On HF, noise from the AF amplifier does not matter much. In this very quiet receiver, noise from the AF amplifier (mostly a hiss) is observed at high volume, because noise is amplified as well as the wanted signal. Another point is that, for measuring signal strength effectively in this direct conversion receiver, an AF AGC in the circuit should be avoided. Summarizing, the internal amplifier noise and the very loud noises must be cut off, but not the relatively strong signals.
In the AF amplifier shown, the first transistor 'behind' the input potentiometer is used as an automatic limiter. Audio signals are taken from the output of the amplifier and drive the germanium diodes (any Ge diode can be used, because of the low forward voltage drop), which rec-

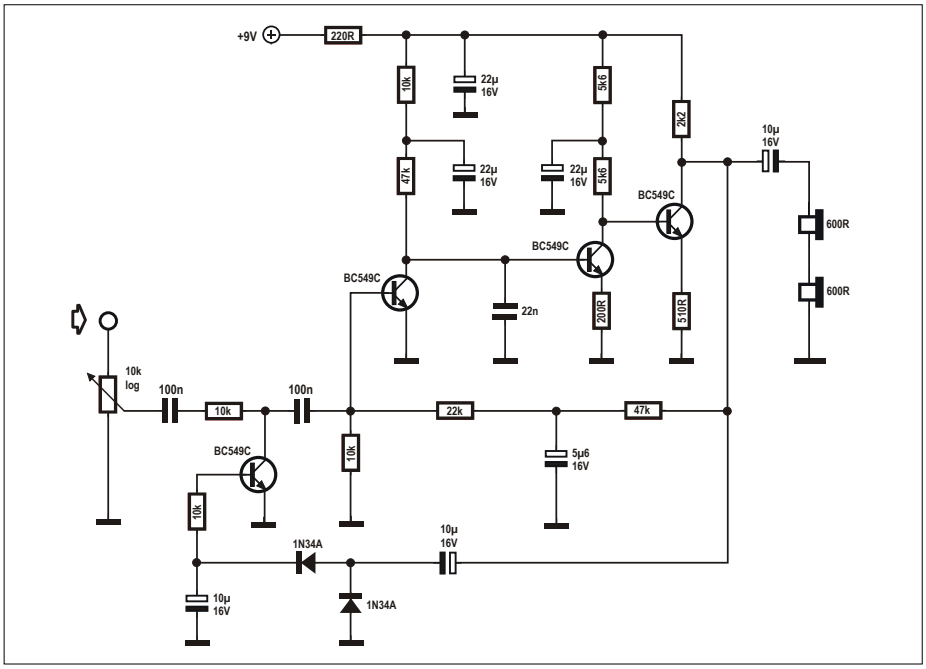


Figure 5: Schematic of the optional AF amplifier.

Practical Use of the Receiver

To really appreciate the features of the receiver you should spend some time operating it. Find some free time, maybe on a rainy Sunday afternoon, away from daily obligations. Sit comfortably in front of your desk with the receiver in front of you and a cup of coffee or your favourite sweet next to it. Don't hurry, relax and enjoy every moment of the receiver you have built and its operation.
Move the resonator storage box in close vicinity and plug in your selected resonators on the ZIF socket. Select the internal antenna or plug in an external antenna and your favourite high-impedance phones. Switch on the power, set the RF attenuation to minimum, select the resonator you want and adjust the volume control for an initial, comfortable noise level. Adjust the regeneration until you notice the oscillation hiss. No worry if you set the regeneration too far since this receiver is unlikely to oscillate at audio. With each resonator setting, find the optimum oscillation level by noticing the audio volume on the phones (maximum is best).
I usually listen to DSB, setting the regeneration to maximum permitting both oscillation and good audio volume. That way I do not have to continuously adjust the regeneration. It is only when nearby interference by other stations occurs that I turn down the regeneration to reject the LSB. It is easy to find the critical point of the regeneration control by noticing the change in the audio response. As you pass from oscillation to no oscillation, the audio bandwidth is reduced, like when you turn down the treble on your stereo, and a characteristic hiss is heard. You will still hear signal tones clearly (sometimes a bit softer) on the USB at that point, but the LSB signals gradually disappear, starting from the higher tones to the lower as you continue to lower the regeneration. If you continue to lower it, the low tones of the USB start to

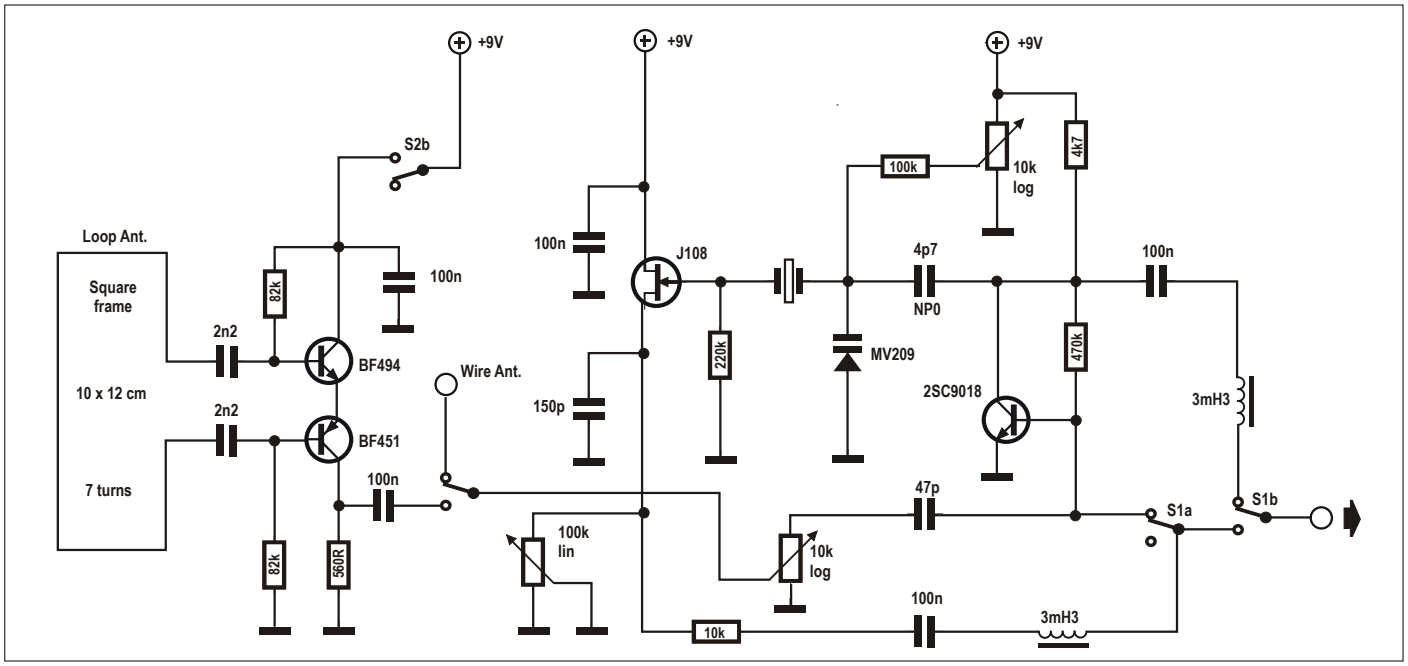


Figure 6: Schematic of receiver front-end with wideband active balanced loop antenna preamp added.

tify it and charge the 10-µF capacitor at the base of the transistor. Depending on the instantaneous charge, the transistor "opens" more or less, which affects the AF output of the voltage divider formed by the transistor proper and its 10-k collector resistor. The component values have been chosen so that only very loud signals at the output of the amplifier (switch clicks, amplifier oscillations) can cause limiting of the input volume. The attack time is very short, which means that the limiter responds rapidly to all these noises, protecting the listener's ears. The decay time is about half a second, which still allows the listener to be informed when these noises occur but does not allow losing much of the following wanted signal.

The 22-nF C from the collector of the second transistor to ground is the anti-hiss filter. Without it, amplifier hiss is very noticeable. Installed, the amplifier is very quiet even at high volume settings. It also helps in preventing amplifier oscillation. Any value from 10 nF to 22 nF can be used. Do not go below 10 nF as the hiss will become noticeable again. Do not exceed 22nF because you will cut off much of the wanted audio bandwidth. 22 nF is probably too much, but it has been found less important to cut off some high audio frequencies to prevent listening fatigue.

The output of this AF amplifier is a high impedance (2.2 k). This saves lots of power from the battery, but it also requires high-impedance phones. Two separate 600-ohm magnetic ear plugs have been used, which were connected in series for better audio volume. 600-ohm phones will work too at slightly lower vol-

vanish as well, similar to what an audio HPF would do. These adjustments are a bit touchy, but they can be done comfortably with a stable hand. Play around by setting the regeneration back and forth at that point, to understand the operation of how the LSB is cut-off by the appropriate regeneration setting. Once you are on a signal, switch back and forth from the external antenna to the active loop and notice the differences in signal level and background noise. The RF attenuator will be useful here. Rotate the loop to see the effect of rejecting interference caused by home appliances while also finding the direction of the interference or the useful signals. Switch back and forth the active filter on narrowband signals and notice the difference on the audio response. Upon switching to the narrow audio filter, you may have to do careful fine adjustment of the frequency (more finely performed by the regeneration or even the RF gain control to some point), in order to centre the signal tone to the filter peak. Toggle the regenerative/reflex-regenerative switch and notice the difference in the audio level but also the background noise. All these controls — interacting occasionally — allow you to precisely define the way of the reception of a signal. Starting from the RF end, affecting the actual way of detection and ending at AF: every stage with its own user selectable settings. The combinations are endless, and it is only up to your available time to explore them. In any case, this receiver will please you not only because of its operation and performance, but also because of its high 'collection' factor. If you happen to be a collector of electronics components you may find it fun to try to dig out different rare resonator frequencies from the web, shops or hamfests to try out in this receiver.



Figure 7: The loop antenna mounted at the back of the receiver and folded in.



Figure 8: The loop antenna folded out.

ume. Any high-impedance phones can be used with this circuit, but if you has the luxury, select phones that provide maximum volume and comfort.

It is too bad to ruin any vintage earphones by connecting them in series in a single connector, so two sockets can be used on the receiver chassis and the phones can be wired in series within the receiver enclosure. Because there are two series connected sockets in the receiver, connection of a computer to the receiver can be accomplished by unplugging one of the two phones and connecting the computer sound card input to it with an audio cable. This allows the listener to monitor the audio signal from one of the two phones, as well as from the computer, although the computer would be normally more sensitive and would require the receiver audio volume to be set to lower levels.

Optional wideband small loop active antenna

The schematic in **Figure 6** shows the schematic of the front end of the receiver with the addition of a wideband active balanced loop antenna. To save power drawn from the battery, the preamplifier power is switched off when the wire antenna is selected instead. This design avoids the use of a matching transformer or tuned circuits, which makes it easy and cheap to build.

The loop antenna is wound on a 10 × 12 cm plastic square frame, with 7 turns of small diameter (standard) insulated copper wire. In the prototype, a plastic frame from an old AM radio was actually used for the purpose (see **Figures 7 and 8**). This frame allows for the antenna to be rotated in the horizontal plane, which could be useful if the receiver must be placed in a fixed position onto a desk. Interference sources can then be rejected by rotating only the antenna frame instead of the whole receiver.

The active loop antenna's performance is very good, considering its small size and the small parts count. Compared to a 12-m 'sloper' wire on the roof, the 10 × 12cm active loop has almost the same gain and this has been tested from 2 MHz and upwards. The loop being balanced, is also less prone to picking up local interference. Additionally, a great advantage is the ability of the loop to suppress unwanted signals or interference coming in from a specified direction, by physically rotating the loop. With this antenna attached onto the receiver enclosure, you can enjoy true portable operation with a receiver that can be taken anywhere, without any dependency on large antennas.

Overall construction of the receiver

The receiver is built into a small plastic enclosure. A metal enclosure would probably be better, but this would be more difficult to drill and it could potentially interfere with the loop antenna's reception pattern. However, some short of RF grounding and isolation from hand effects is needed and this is provided by the large circuit board inside the receiver. The knobs of the controls are chosen to be quite big (but not huge) so as to provide a degree of fine tuning without the need for expensive vernier drives. The regeneration is the most critical setting and this can be very conveniently set for single signal reception even at the higher shortwave bands. To avoid drawing a lot of front panel markings, knobs with scale markings on them were chosen.

The front panel (**Figure 9**) has switches



Figure 9: The receiver front panel.

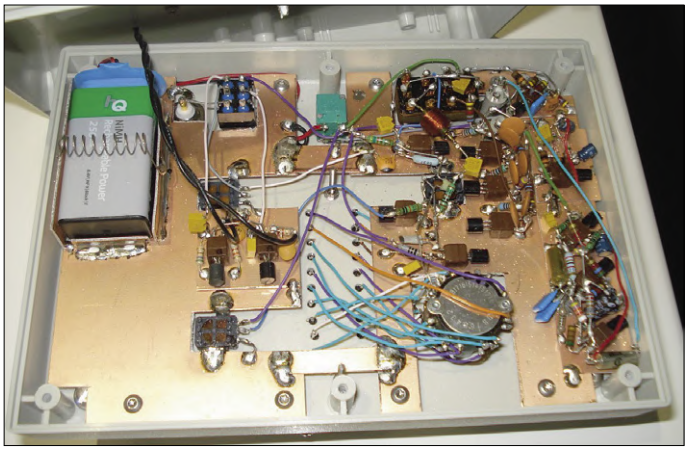


Figure 10: The internals of the receiver, all made using discrete components and no PCBs.

for selecting the antenna and the audio filter type, as well as the topology (regenerative or reflex-regenerative) and of course the external antenna connector and the ZIF socket.

The crystal selector/resonator is a new idea. One can never have enough resonators and the receiver is designed to be broadband (i.e. cover all HF band by just changing resonators). Hence, simply soldering the resonators onto the selector switch is not a good idea. On the other hand, having just a crystal socket on the front panel means that the operator would need to carry all the resonators with him all the time as he goes outside with this receiver. So a compromise between the two was devised. A ZIF (zero insertion force) IC socket is used to hold ten resonators. If a resonator or a set of resonators need to be replaced, you unlock the socket lever and replace the resonators. Another advantage of the ZIF socket is that the pins of the resonators do not wear out as they are inserted many times into the socket. A 40-pin ZIF socket is used, which would normally hold 20 resonators. However, the intermediate socket pins were removed so as to reduce the parasitic capacitance of the socket as well as create some space for the adjacent resonators.

The method of organizing the resonators onto the socket is personal. You may organize them by band or by preferred frequencies, etc. If they are organized by bands then additional DIP sockets (one for each band) may also be used and the band resonators installed onto each of them. When changing band is needed, unlock the ZIF lever and replace the whole DIP socket which holds the band resonators. The possibilities are endless, it is really up to you to decide the most convenient way.

The selector switch should be placed as close to the ZIF socket as possible. A rechargeable battery is used for the receiver and a holder was made for it out from two pieces of unetched circuit board, soldered at the sides to keep it in place. A small spring prevents the battery from dropping out and allows easy replacement.

It is important to use good quality potentiometers in this receiver for smooth setting of the controls. Panasonic miniature plastic potentiometers were used for all the controls except the audio volume. The audio volume potentiometer also has a switch on it, which is used as the on/off switch for the receiver. An on/off LED was not included as there to avoid drawing precious current from the battery. The front panel handles are placed such that they protect the controls if the receiver accidentally drops with the panel faced down.

There are not many things that can go wrong when building this receiver, provided you follow the schematics correctly. After all, there are no hand-wound inductors and the receiver was designed with simplicity in mind. With light reference to the construction pictured in **Figure 10** it should be readily reproducible by most radio amateurs with basic electronics skills. Should any queries arise, do not hesitate to contact me through [1]. 's wensoty's webs so I can provide help. ⏮

180480-01

Web Link

[1] Author's website: http://qrp.gr/

RF Gain and Ear Fatigue

The use of the RF gain control has interesting properties. By reducing the gain of the transistor, the wanted RF signals are preamplified less. However, you will practically see that without the interfering rectified AM signals anymore, you will be able to receive the wanted ones more clearly, despite the lower RF gain. QRM and noise, is also reduced by reducing the RF gain and this helps in receiving weaker signals despite of the lower RF gain. Especially in narrowband modes, it seems most of the signals find their way to your ears, as the noise is less and the ear fatigue less also.

Communication using UDP Packets

No sweat for the ESP8266

By **Gilbert Ghyselbrecht** (Belgium)

To exchange data over a network, a client/server model is usually used. The client sends a request to the server, and the server sends its reply in the form of an HTML page with the requested data. This model requires a lot of software: a server program on a Raspberry Pi for example, and a program on the client side that distils the data from the HTML page.

For many applications such as DIY projects, modelling, radio control and so on, all you need are small modules such as the ESP8266 with Wi-Fi on board for wireless connectivity over short distances. Access to the Internet is often unnecessary and sometimes even undesirable.

UDP broadcast packets are used to convey data to be sent to a randomly selected port number. Such a packet is not redirected by the router to another network segment; any computer that is part of the same network segment (subnet) and that 'listens' to the relevant port will receive all the data.

UDP (User Datagram Protocol) [1] is not a secure connection, as is the case with TCP (Transmission Control Protocol) [2]. UDP does not guarantee that all data packets will actually arrive at the recipient; packets that are lost en route are indeed lost. The great advantage of UDP is that it is a fast protocol — that is why it is especially used when speed is critical and the loss of a packet does not have dramatic consequences, such as in the case of videoconferencing, telephony or some online games. If the data is regularly retransmitted, like every few seconds, or every time a sensor value changes, nothing will be lost in the long run.

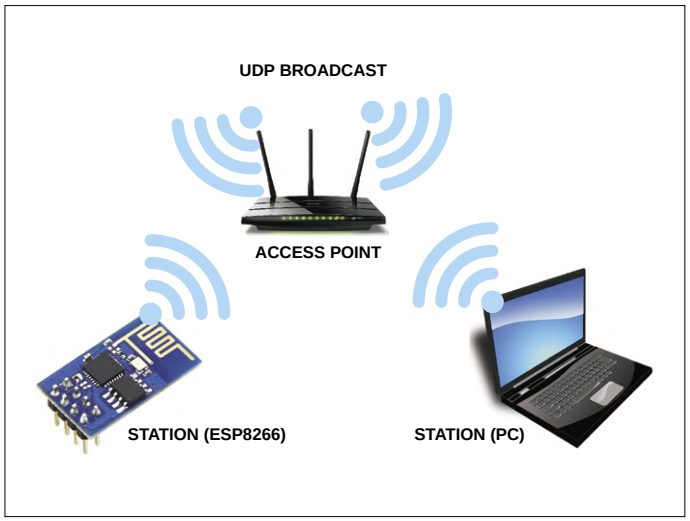


Figure 1: Here the ESP8266 works in Station mode.

The exchange of a UDP packet is extremely fast — normally only 2–10 ms elapse between closing a switch on the transmitter and an LED lighting up on the receiver.

In a client/server application, the client keeps waiting for a response from the server. When this does not happen for whatever reason, often everything must be rebooted to re-establish the connection. In a UDP broadcast network, any module, PC or tablet may be connected or disconnected from the network without affecting the other participants. After all, all data is sent to all participants on a regular basis. **Figure 1** a schematic representation of this setup.

Examples using the ESP8266

The ESP8266 module has a powerful 32-bit processor permitting use as a Wi-Fi station or an access point for other stations. The example programs we use to demonstrate the different possibilities are made with the help of the Arduino IDE 'adapted' for the ESP8266 [3].

In the examples, fixed addresses are used, for the simple reason that the startup is faster while we no DHCP server is called for. It is also easy to use a NetScan or similar program to keep an overview of what is happening. The various modules connect to a router set as a Wi-Fi access point. Note that this router does not necessarily need to be connected to the Internet [4].

The transmitter

Switches as well as most sensors need to send information only; nothing needs to be received. This is therefore a one-way communication. Below is an example program for an ESP8266 module.

At the beginning of the program, some global data are recorded, such as the fixed IP address, the gateway and the subnet. Any value greater than 1024 can be used as a port; the author used port 7923 in his example programs. The broadcast address is the same as the gateway address, but with 255 as station number. This part of the program also contains the SSID and any password for the access point.

```
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>

int led = 2;
int key = 4;
boolean state;

WiFiUDP Udp;

IPAddress staticIP (192,168,1,130);
    // change for every module
IPAddress gateway (192,168,1,130);
IPAddress subnet (255,255,255,0);
IPAddress broadcast (192,168,1,255);

unsigned int UdpPort = 7923;

char* ssid = "MySsid";
char* pasw = “MyPassword”;
```

During the execution of the program, the setup() routine is executed once. In this routine, the Wi-Fi mode is started using WiFi.begin() and WiFi.config(). The program then waits until the WiFi.status() function responds with WL_CONNECTED. The blue LED then flashes in a seconds rhythm. If no Wi-Fi connection has been established after 10 seconds, the ESP module will restart with ESP.restart().

```
void setup(){
  int count = 0;
  boolean ledstate = 0;

  pinMode (led, OUTPUT);
  pinMode (key, INPUT);     // Input switch

  WiFi.begin (ssid, pasw);
  WiFi.config (staticIP, gateway, subnet);

 while (WiFi.status() != WL_CONNECTED)
  {
    delay (500);
    ledstate = !ledstate;
       // flash led until connected
    digitalWrite ( led, ledstate );
    if (++count >20 ) ESP.restart();
       // if no connection restart ESP
  }
  digitalWrite ( led, 1 );
}
```

The subroutine loop() is called again and again; it contains the actual program. In this example, the position of a switch connected to GPIO2 is read; if the switch’s condition changes, either a UDP-ON package or a UDP-OFF package is sent. This is done with the sendUDP() function.

```
void loop() {
boolean P = digitalRead (key);
if   ( !P && !state )  SendOn();
else if ( P && state ) SendOff();
}

void SendOn() {
  state = true;
  digitalWrite ( led, 0 );
  sendUDP ( "1 Switch ON" );
}

void SendOff() {
  state = false;
  digitalWrite ( led, 1 );
  sendUDP ( "0 Switch OFF" );
}
```

The UDP package is created using the Udp.beginPacket() function. The data to be sent is added to the packet in the form of a ‘char’ (character) array using the Udp.write() function. Finally, the packet is sent with Udp.endPacket().

```
void sendUDP ( char * s) {
  Udp.beginPacket ( broadcast, UdpPort );
  Udp.write ( s );
  Udp.endPacket();
  delay (5);
}
```

In this way, a wide variety of sensor and switch information can be transmitted.

The receiver

Here, the Wi-Fi module is being initialized in the same way as the transmitter. At the end of the setup() routine, the Udp.begin(UdpPort) function is added, which listens to whether data packets arrive at the specified port.

```
int led = 2;
int relay = 5;
boolean ledState;
WiFiUDP Udp;

unsigned int UdpPort = 7923;

char inbuf[255];

…

void setup() {
```

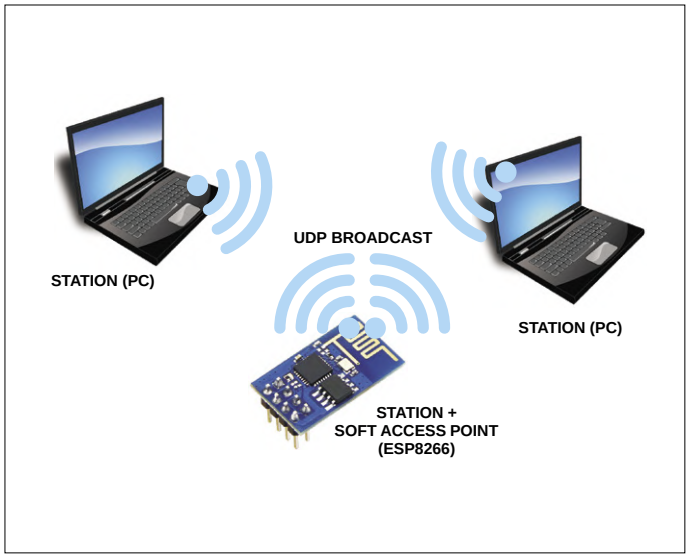


Figure 2: Here the ESP8266 is used as a drive and as a (soft) access point.

```
  int teller = 0;
  boolean led1 = 0;

  pinMode (led, OUTPUT);
  pinMode (relay, OUTPUT);

…

  Udp.begin (UdpPort);
}
```

During the `loop()` routine, the reception of data packets is tested using the `Udp.parsePacket()` function. This function returns `True` when a packet is received. In this case, the data is retrieved and processed with the function `Udp.read()`.

```
void loop() {
if (Udp.parsePacket()) {
    int len = Udp.read (inbuf, 255);

    if (inbuf[0] =='1') ledState = 0;
    if (inbuf[0] =='0') ledState = 1;

    digitalWrite (relay, !ledState);
    digitalWrite (led, ledState);
  }
}
```

The ESP8266 as an access point

The above setup works fine in itself but is a bit over the top when configuring an ESP8266 as an access point (see **Figure 2**). A maximum of five modules or computers can then connect to this soft access point. In addition, the module can also transmit and receive as a station.
With the WiFi.softAP() function, the ESP8266 is configured as an access point (with name and password).

```
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>

int led = 2;
int relay = 5;
boolean ledState;

WiFiUDP Udp;
unsigned int UdpPort = 7923;
char inbuf[255];

void setup() {
  WiFi.softAP ("ELEKTOR-01", "password");

  pinMode (led, OUTPUT); digitalWrite (led, 1);
  pinMode (relay, OUTPUT); digitalWrite (relay, 0);

  Udp.begin (UdpPort);          // listen on port
}

void loop() {
   if (Udp.parsePacket()) {
      int len = Udp.read (inbuf,255);

      if (inbuf[0] == '1') ledState = 0;
      if (inbuf[0] == '0') ledState = 1;

      digitalWrite (relay, !ledState);
      digitalWrite (led, ledState);
    }
}
```

By default, the ESP module launches with the IP address 192.168.4.1. The modules wanting to connect to this access point must therefore use an IP address from the 192.168.4.2 to 192.168.4.254 range.

With the function `WiFi.softAPConfig()` you can change the address of the access point:

```
IPAddress local_IP(192,168,20,1);
IPAddress gateway(192,168,20,1);
IPAddress subnet(255,255,255,0);

void setup() {

  WiFi.softAPConfig(local_IP, gateway, subnet); //
change access point address
  WiFi.softAP("ELEKTOR_01", "password");
```

Even after loading a new program, the ESP8266 module continues to function as an access point. In the setup() part of that program, you need to use the function WiFi.softAPdisconnect() to end the access point mode.
The saved address data can be deleted through the menu item Tools -> Erase Flash: “Sketch + WiFi Settings” of the Arduino IDE.
As you have seen, UDP broadcast packages are not at all dif-

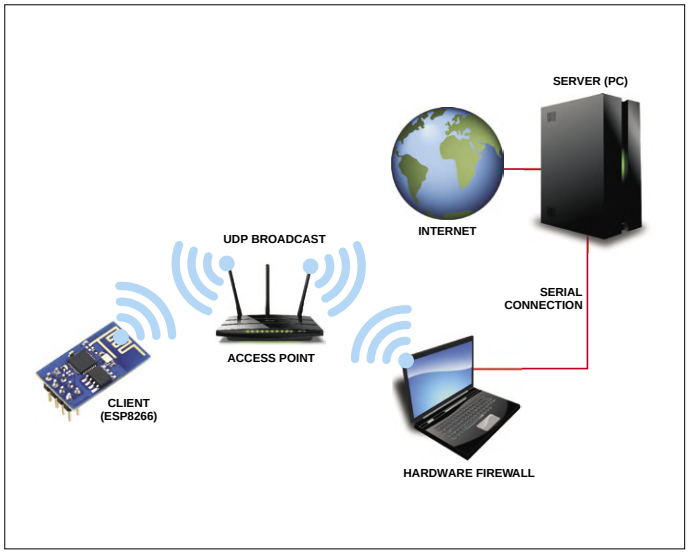


Figure 3: Using this setup a secure connection to the Internet can be established.

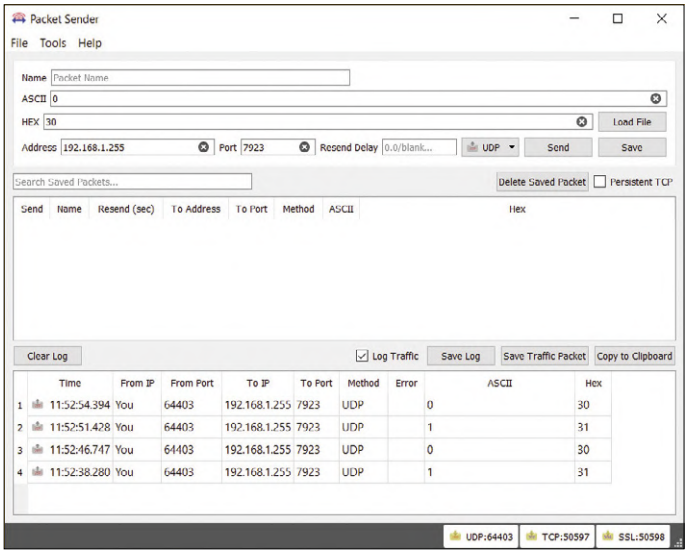


Figure 4: The Packet Sender program allows you to test and try out your own UDP broadcast protocols.

ficult to design a simple proprietary protocol to interconnect devices, sensors and switches. Each station in the network segment receives everything and based on the received data, can decide whether or not, and specifically what action should be taken. A PC or tablet can be used to summarize the data of all modules in one overview. It is conceivable to have multiple panel PCs mounted at different locations in the network to show the same information anytime, anywhere.

Hardware firewall

If a connection to the Internet is required, for example to monitor communications from a completely different location or to remotely set up equipment, this does not necessarily pose a security risk. As outlined in **Figure 3** it is conceivable for an ESP module or a PC to collect all the data from the UDP packets, summarize it in a table and then, via a serial connection, send it to an RPi or PC that is connected to the Internet as a web server. It can be set up in such a way that nothing 'gets through' that has not been sanctioned in advance by software.

Conclusion

The application of UDP packages is so simple that everyone can design their own, tailor-made protocol, as we have shown above. The user is not subject to any restrictions because no software modules or libraries of third parties are used and the user cannot or may not deviate from the protocols provided for this purpose.

And of course that suits our ElektorLabs readers as they can experiment to their heart's content with and build on the examples from this article. With a program like 'Packet Sender' you can follow what is happening (**Figure 4**) [5].

The examples discussed in this article are available for free downloading from the ElektorLabs article support page [6].

(180453-02)

@ WWW.ELEKTOR.COM
→ ESP8266 WiFi-module
www.elektor.com/17326

Web Links

[1] User Datagram Protocol: https://nl.wikipedia.org/wiki/User_Datagram_Protocol
[2] Transmission Control Protocol: https://nl.wikipedia.org/wiki/Transmission_Control_Protocol
[3] Arduino IDE and ESP8266: https://arduino-esp8266.readthedocs.io/en/latest/installing.html
[4] ESP8266 and WiFi: https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html
[5] Packet Sender: https://packetsender.com
[6] Project page for this article: www.elektormagazine.com/180453-02

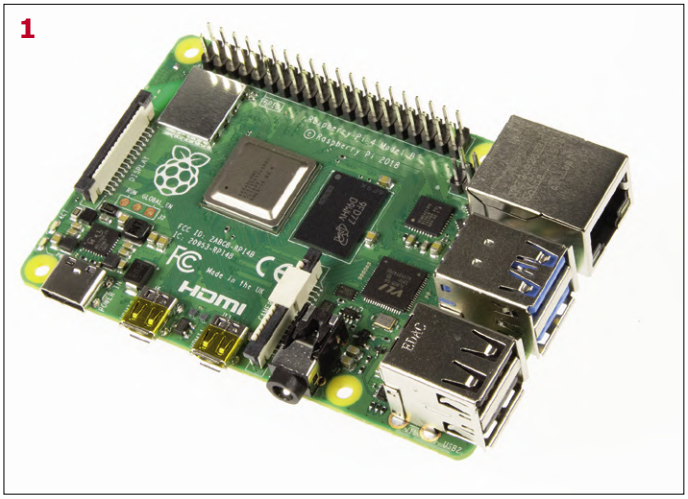
Review: Raspberry Pi 4

All new but still good?

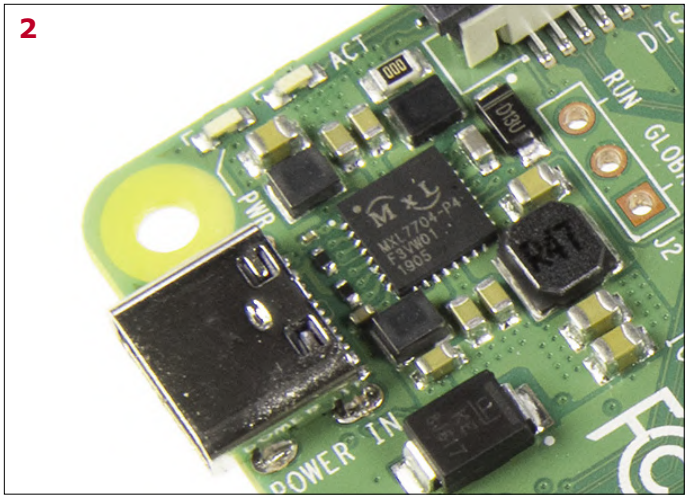
By **Mathias Claussen** (Elektor Labs)

The Raspberry Pi 4 has arrived, and it comes with a lot of long-awaited changes. Let's get hands-on with it and have a first impression of the new hardware.

1



2



At last, the new Raspberry Pi 4 hardware has arrived. The last time new hardware arrived the steps forward were modest if not small. From Raspberry Pi 2 to Raspberry Pi 3 was a nice step forward, but from 3B to 3 B+, to be honest, was more a facelift than a real step. Over time, after the launch of the Raspberry Pi 3 B+, competitors appeared with more powerful and feature-rich hardware like USB3 and native Ethernet, just to name a few improvements. This time the new Raspberry Pi 4 had to be truly new hardwarewise while retaining as closely as possible the form factor we got used to. We first look at the hardware, starting with with **Figures 1 and 2**, and then the software.

USB-C input and PMIC

The first thing you notice if you grab the new Raspberry Pi 4 is that it won't fit any of the usual enclosures. This is due to some major changes in the display connection and power supply. We now have a USB-C connector for power, and two micro-HDMI ports to connect our 4k screens to. So, for the first time we have hi-res dual screen support through a digital interface, and powering from your universal USB-C supply.

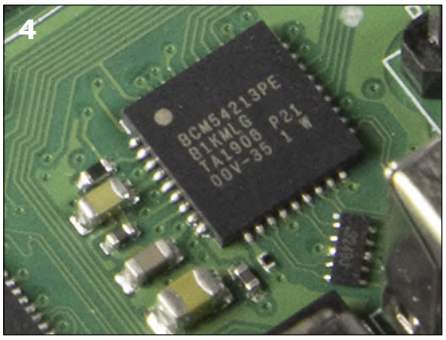
The new SoC on the Pi 4 B

You will also note the two blue USB ports. And yes, we finally got USB 3.0 onboard as well as native Ethernet, effectively ending all troubles while transferring data from our mass storage device to the network. It's all thanks to the new BMC2711 SoC at the heart of Raspberry Pi 4 (**Figure 3**), which sports four Cortex-A72 cores with up to 1.5 GHz speed and a maximum of 4 GB RAM. Meaning a per-MHz performance improvement over the Cortex-A53 at the cost of more supply power. This will result in improved response even with normal desktop applications. But before we come to software and the improvements, we look a bit further. Also new is the VideoCore VI, happily affording some new feature like 4k output and 4k HEVC decoding at 60 fps. The SoC comes without a fan or heatsink attached to it, making it a nice quiet solution like the previous generations. How this works during operation is shown by a few thermal imaging camera shots. Also, with the new SoC the power requirements have changed: unlike the well-established 5 V and 2 A as a good starting point (okay, 2.5 A for the B+), we now have 5 V @ 3 A recommended on the USB-C input, meaning we need to feed a good 15 watts worth of power to Raspberry Pi 4. Besides the CPU we have a few new chips on the board.

The photos show a close-up of the gigabit ethernet phy used (type BCM54213PE, **Figure 4**) and a USB Hub, a type VLI VL805-Q6 (**Figure 5**). If you look at the datasheet of this USB 3.0 Hub you will soon stumble across the fact that there is no USB 3.0 Uplink. Instead, a PCI-Express lane is involved, supporting PCI-E 2.0 and giving backward capability to PCI-E 1.0. Meaning we now have some serious bandwidth to enjoy.

In terms of wireless connectivity, Bluetooth has been upgraded to version 5.0, while WiFi is 802.11AC at 2.4 GHz and 5 GHz. The details for WiFi connectivity will follow later, as we run some data transfer tests with it.







USB 3.0 Hub with PCI-E uplink

Some trusted features that remained on the new Raspberry Pi 4 are the RCA port, display connector and the camera interface. Also untouched and functionally backwards compatible is the 40-pin PCB header. So we can attach HATs like we used to, at least hardwarewise.

A few words of warning — what you read here is the result of using an alpha image for Raspberry Pi 4, meaning some things are broken or simply missing. It is reasonable to assume that the system software will improve over time just as with the Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi 3 B+ and Raspberry Pi Zero W.

The software side of Raspberry Pi 4

This takes us to the software support for Raspberry Pi 4. As we are on a pre-beta, not everything works completely out of the box, while the process of updating parts to Raspberry Pi 4 is in progress. We have two options to choose from: a Raspbian buster image or a NOOBS 3.1. We found that either image boots just fine and will display a desktop properly after a short startup process.

4k video times two

Since we now have two micro-HDMI ports, we hooked up two Full HD screens to the Raspberry Pi 4. Out of the box we got an expanded Desktop on both screens. We also tested the tiny 10" Joy-It screen from the Elektor Store as secondary monitor, but it currently refused to work with Raspberry Pi 4. We should have support for dual 4k video output at 30 fps when using two screens, or 4k at 60 fps using only one screen attached.

But what to do with two screens? You can play video on one screen while browsing the web on the other, or even play some 3D games. And this bring us to the video playback functionality. We ran the first test with the VLC on Raspberry Pi 4 and Big Buck Bunny in 1080p @ 60 fps and 4k @ 60 fps resolution. As a first test we started the VLC and used the 1080p material in 'windowed' mode. Here we sadly missed the nice, fluent playback we used to have. At the same time, we noticed a high CPU load, indicating that parts of the decoding operations are not offloaded from the CPU. Switching to full screen, the CPU workload drops, and the video runs smoothly. With two outputs available we also tried to get two different videos displayed on the screens. Sadly vlc refused to do so — apparently at this time the video decoder can handle just one video stream at a time. Putting a 4k video, in h.264, on the test bench with a 1080p monitor attached causes vlc to produce no image at all. Only sound was played, as 4k playback is only supported using H.265. We used material form libde265.org for the 4k playback. Sadly we only got a black screen using vlc and a high CPU load, meaning the decoding is not running in hardware as intended. As 4k video should be supported, we assume there is a bug in the driver, and the support is subject to improvement in the future. If you ask how the omxplayer worked with the files, well, it plainly refused to decode anything, since with the ongoing changes in the video decoding stack it couldn't access the hardware anymore.

After this little video test with vlc, why not try something with kodi? As most video-centric distributions out there haven't been able to update their images for the new hardware now, only an old version from the Raspbian repositories was available. It made no sense to test this any further as we won't be able to get any video acceleration with kodi on Raspberry Pi 4 at all.

Blue is the new black....

At last we have what many of us have sorely missed: proper USB ports and not just a USB 2.0 with an ugly uplink to the SoC. The two blue ports and the new USB Hub are here to say: we entered the USB 3.0 world (**Figure 6**). Also, Ethernet is no longer attached to just one of the USB ports of the Hub — instead it is connected directly to the SoC, meaning more USB bandwidth for our devices and offering a big step forward towards network throughput.

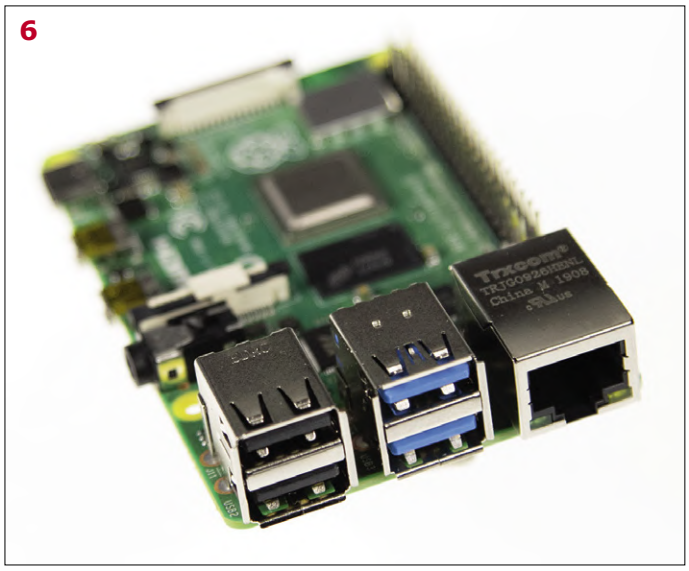
With USB 3.0 and real gigabit Ethernet now available, let's see how this performs. With 1000 Mbit/s you can assume a 125 MB/s per second raw data rate ideally, and start making a neat NAS for your home. We used SAMBA for testing, as this is most likely the preferred way of sharing network data, and something the three major types of OS can work with.

For storage we use one Kingston DataTraveler 100 USB Stick with 32 GB of storage and EXT4 for the file system. As the second storage option we used a RAM disk as we now have enough on our hands to "put to waste" in 1.2 GBytes. To set up the RAM disk we typed:

```
sudo mount -t tmpfs -o size=1200M none /ramdisk
```

After the configuration of samba (Windows Fileshares on Linux) we needed some data to copy. We used "bbb_sunflower_1080p_60fps_normal.mp4" and "bbb_sunflower_2160p_60fps_normal.mp4" from http://bbb3d.renderfarming.net/download.html as this will give us 1,004,968 kB worth of files, and we already had them handy from the video playback tests. We first copied these to the USB stick and then measured the time it took the Raspberry Pi 4 to copy them to the RAMdisk using a simple "cp" command. This will preclude a disc slowing down the transfer due to low writing speeds.

With a small script showing the timestamp at the start of the



transfer and another one at the end we got 9.3 s meaning 108.05 MB/s, sadly a limit of the flash drive used. Copying the files a second time will reveal an increase in throughput as now the additional RAM comes in handy, to work as a cache for the drive. Compared to the speed of a Raspberry Pi 3 B+ that's fast, as with USB 2.0 you will get around 50 MB/s max. On the Raspberry Pi 3 B+ this also means you must share bandwidth with the Ethernet. Speaking of Ethernet, with the RAM disk we can read at 102 MB/s of payload from an SMB Share, meaning it's nearly maximum speed for 1 Gbit/s considering the SMB overhead. For the files we used a Windows 10 Pro machine with an SSD as the target device. The last try was a copy from the directly attached USB Stick. From experience, we can read data from it at speeds exceeding 100 MB/s. For the transfer we got 98 MB/s, again faced with the overhead of the USB stick. For a small, well priced fileserver, this is still okay and shows improvements we were long waiting for.

Let's turn the radio on

As on the Raspberry Pi 3 B+ the WiFi chip proper is buried under a metal cover for better EMI performance. The layout of the antenna is also like the one found on the older models. Let's cast a brief look on the WiFi functionality.

On the Raspberry Pi 4 we again used the RAM disk to copy data to our Windows 10 machine. This time we disconnected the LAN and switched to WiFi in the 2.4 GHz band. Here the Raspberry Pi 4 will run at 150 Mbit air rate maximum linked our test access point, considering the access point could do 300 Mbit/s. We used an ideal setup with the Raspberry Pi 4 and the access point at line of sight and less than 10 meters apart.

It's not as common as in the real world, but a limit set by the lab. We also transferred the same amount of data as before (1,004,968 kBytes) and saw the transfer finished after: 159 seconds. This gives us 6.32 Mbytes per second transfer rate. Okay by all means, but what about the 5 GHz band, should this be faster? Well, yes it is. But we still have the drawback that on 5 GHz your access point must use a channel lower than 64, otherwise the Pi won't be able to find it. The transfer speed achieved here was 7.17 Mbytes per second as a result of the 200-Mbit air rate offered by the Raspberry Pi 4. For video streaming that's workable — to use the Raspberry Pi 4 as access point, it's unlikely to be enough.

Bluetooth! Yes it's there, and it performs as you may expect with Linux. Regarding Bluetooth 5.0 devices, sadly we did not have one to play with but if one arrives, we will update you with the results.

How fast can it be?

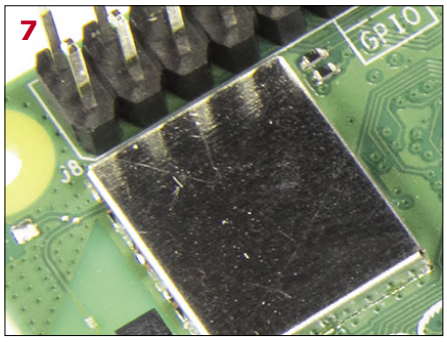
We have now 1.5 GHz maximum frequency for the cores, i.e. 100 MHz up on the Raspberry Pi 3 B+. On paper that's a 7.1% increase over Raspberry Pi 3 B+, and it may not sound much. However, at the same time the cores in Raspberry Pi 4 got changed to Cortex-A72 types. These cores were designed with performance per MHz first, and performance per watt, second. Doing the maths it ups the performance compared to the 3B+. If we use sysbench with one core and enter:

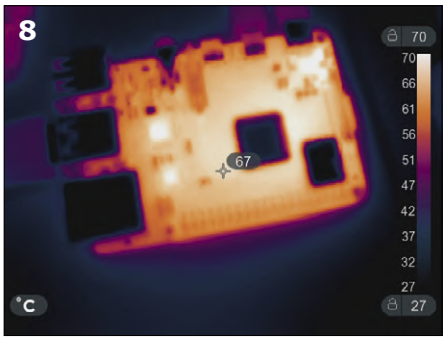
```
sysbench –test=cpu run
```

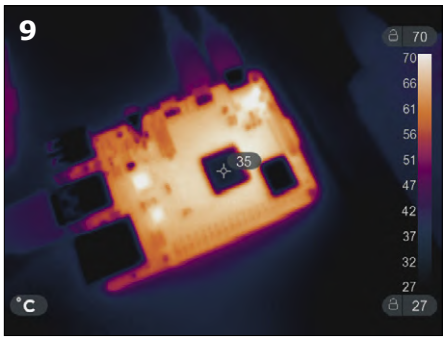
then 121.35 s is needed on the Raspberry Pi 3B+. Running the test on one of Raspberry Pi 4 cores, we measured 92.78 seconds of computation time, i.e. roughly 30% faster than Raspberry Pi 3 B+ for this synthetic test. If we run the test on both systems on all cores we get 34.54 s on the Raspberry Pi 3 B+ and 23.25 s on the Raspberry Pi 4, a vast increase mainly on account of the new cores.

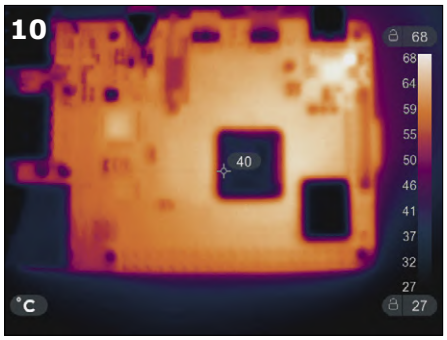
This sounds not so bad for the CPU performance, but comes at a price: upped power consumption. Looking at the thermal images in **Figures 8 and 9** you can clearly see that the Raspberry Pi 4 runs somewhat hotter than the Raspberry Pi 3 B+.

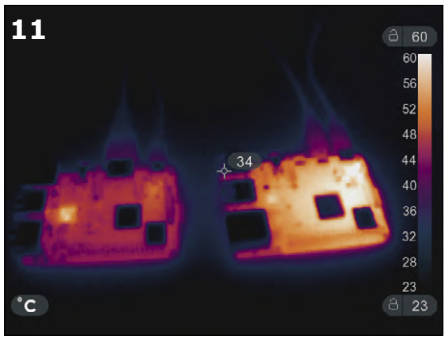
After some 'loading' with software it will reach temperatures well beyond amusing levels. With 68 °C peak and hot spots around the CPU exceeding 55°C it's plain to see that cooling is in order. The software image we got to play with was still in prerelease state and we should expect this to be optimized. Due to the nature of the Cortex-A72, i.e. speed prevailing over power consumption, serious amounts of dissipation were expected, but not to the extent observed in practice. The black spots seen in the thermal image are the metal caps of the CPU and wireless module, but believe me, they are not as cold at the











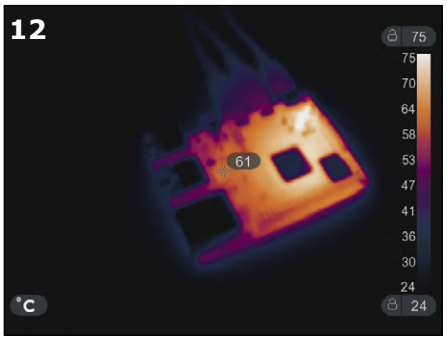


image shows, in fact on the Raspberry Pi 4 you don't want to touch the CPU at all. The second bright, i.e hot spot in the image is the USB 3.0 hub chip — not recommended to touch either. Comparing the Raspberry Pi 4 and Raspberry Pi 3 B+ in pure desktop idle mode, it becomes evident that the Raspberry Pi 4 isn't such a "cool" guy as the little brother. This implies optimisation for power consumption in idle needs to be done, and we expect the situation to improve by the time the final release appears. **Figures 10 and 11** show the idle temperatures of a Raspberry Pi 3 B+ and a Raspberry Pi 4. The cooler one on the left is the Raspberry Pi 3 B+; the one with the bright, warm, spots is the Raspberry Pi 4.

This also means you have to consider where you place your Raspberry Pi 4. A dark dusty corner, otherwise fine for the previous models, is no longer an option. A good airflow around the board should keep the components cool and prevent overheating. Looking at the heat dissipated by the system, you can look forward to a serious amounts of power taken from the supply. On power consumption, the question is how hungry the Raspberry Pi 4 has become. So we measured the current while idle, running the CPU benchmark and also running supertuxkart at 720p resolution. All tests were done with only one display attached using the Raspbian alpha image.

- With desktop idle: 2.8 watts at 5 V supply voltage.
- Playing the 1080p Big Buck Bunny video: about 3 watts in Fullscreen; and 'windowed', 4 watts approximately.
- Running supertuxkart at 720p resolution: around 5 watts when racing around the circuit.
- Sysbench running on one core: 3.9 watts; or 5.2 watts when running on all cores.

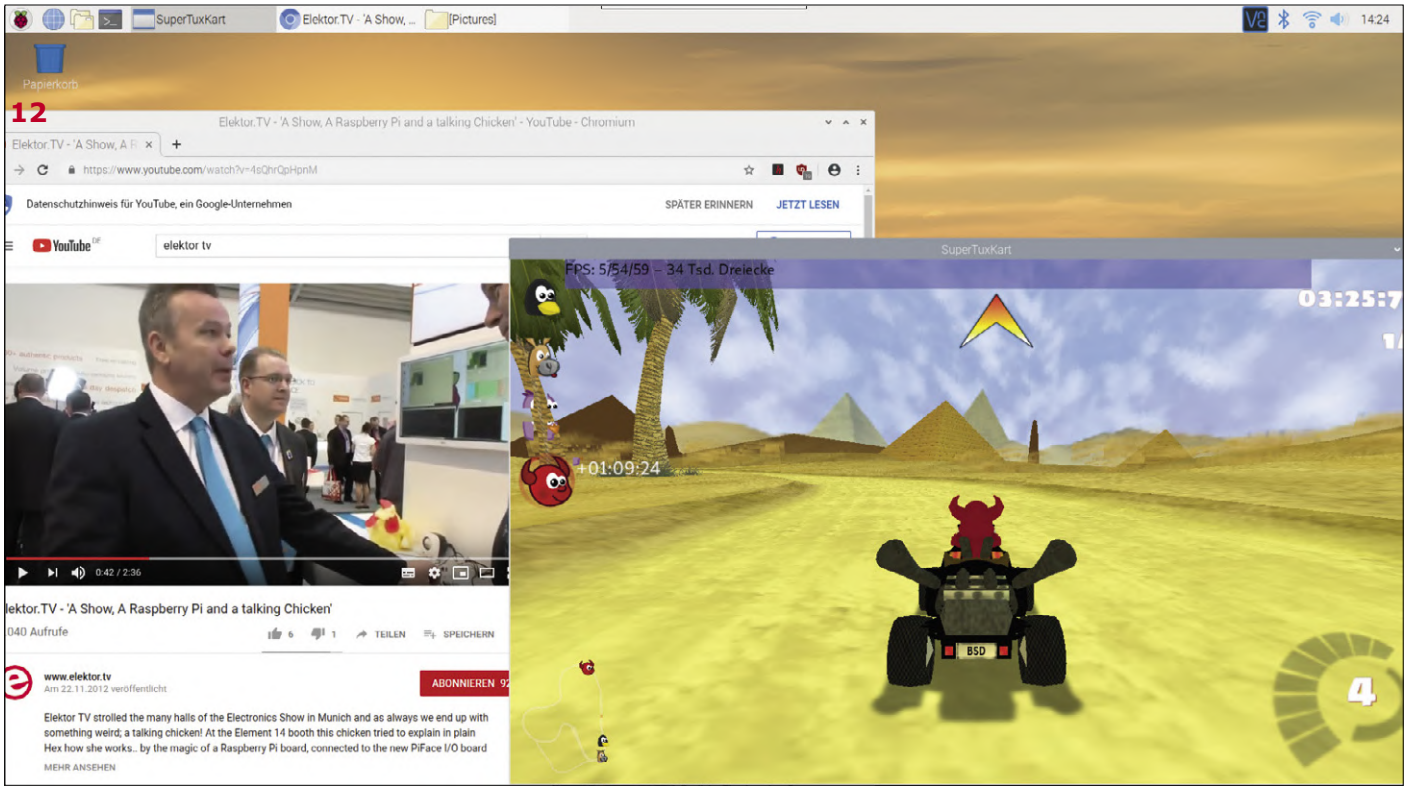
With the increased power demanded by the system, the dissipated heat rises also. If you put a heavy computing load on the system, do make sure to have a constant airflow through the system. Also note that the network port and USB ports get quite hot, too. Furthermore, you need to consider that the USB Hub will also draw some current if you start transferring data over the network, just as the Network or WiFi parts.
For a system mostly idle 24/7 this means we use 24.6 kW per year. If you start loading the system things don't improve either. On the other hand, you get more computation power if you need, but also you get more heat to deal with, especially If you put any HATs on top. With the extra amount of RAM we can also consider running virtual machines on Raspberry Pi 4, having multiple webservers, an MQTT broker, or encrypted file storage on it besides being to able to move quickly from one system to another. Whether or not this works out, only time will tell.

VideoCore VI: 3D is back to the desktop

One thing the older Raspberry Pi versions have lacked for a long time was the ability to use 3D acceleration on the desktop. With the Raspberry Pi 4 and the driver for the graphics part of the VidoeCore VI, it's back finally. As a result, on the desktop we now have 3D acceleration in 'windowed' mode, and it's working! The new core has no problem running things like supertuxkart, which required a lot of tweaks on Raspberry Pi 3 B+, and even then might refuse to run at all.
The Raspberry Pi 4 now starts the game pretty quickly, and with the resolution set to 1024×768 and effects reduced to level 3, we can do a few rounds at 40 fps on average. If we step up to 720p the frame rate will decrease to 29 fps on average and the result is still playable — a clear improvement over Raspberry Pi 3 B+. If you ask how an N64 emulator or PS1 emulator will run, we're sorry to inform that we didn't get them to run timely for the online review. Once they are up and running, we will present the results for the tests, but you can expect it to run with more power than the older versions.

For attaching to a Raspberry Pi 4

For many of you who already own some Raspberry Pi or are looking to start with some existing extension, there is the inevitable question: will add-ons still attach to Raspberry Pi 4? Yes they will, but if you are using one of these TFT Displays that connect to HDMI you need at least a new cable for it.
The camera and display connectivity and support have remained unchanged compared to the Raspberry Pi 3 B+, and so has the beloved 40-pin connector at the board edge. We can attach the same extensions as before, like the Strompi V3. Being able to supply 3 A of current we can cheerfully use StromPi on top of the Raspberry Pi 4. The PoE HAT will fit too but is not completely suitable as we get only 2.5 A output, which is the bare minimum for the Raspberry Pi 4. Meaning: the PoE HAT might work, but if we recall the troubles experienced with the older models, we will leave this experiment for later. What we will do though is a test run of a 3,5" TFT on the Pi, and a test of the Raspberry Pi Foundations 7" touchscreen hooked don the display connector.
The first test using the display port on the Raspberry Pi 4 was not successful: the display was left undetected and showed nothing. We can only assume this is due for patching later.
it is the same for the 3.5" TFT you directly attach to Raspberry Pi models, as some software parts have changed. The screen itself gets recognized after the usual configuration. But if the graphical output is redirected to the display the Raspberry Pi 4 will lock up while booting. This is something that can be fixed with software but isn't ready yet.
After this we did not attach a camera to the Raspberry Pi 4,



in view of the prerelease image we expected some things to not work correctly. Once the final image has arrived and lives a few days in the field, we will take this further to evaluate the support for the components.
Regarding the IOs, and generally SPI and I²C, these things are working as they should, also the GPIOs, as we found after a quick test. Depending on the software and libraries used, the setup you had for older Raspberry Pi models can work on the Raspberry Pi 4. As there is still work in progress, more adjustments may be called for to get it going.

Conclusion

We ran some basic tests covering performance, power consumption, as well as a little light 3D gaming. As mentioned in the beginning we dealt with a beta release image for the Raspberry Pi 4, so the final one should have improvements. For your favourite video-centric distribution, like openelec, some time will pass before the enhancements have been adopted for Raspberry Pi 4. As mentioned in the beginning the software is not complete and needs a few patches. This will take some time but is not unlikely to come about.
Eventually we got the upgrades we waited for, faster USB, faster network, a faster CPU and finally more RAM. This comes at a price, first expressed in currency and secondly in current! With the entry models with 1 GB RAM roughly at the same price level as Raspberry Pi 3 B+, you'll see an increase in cost as you increase the amount of RAM you wish to have. At the same time, you get an increase in power consumption and heat dissipation. Depending on the requirements for your project you may appreciate more RAM and more power consumption, more displays or even 4k, or stick with one of the older models using less power in the long run.
But to be honest, we liked the extra amount of RAM in this thing. Also, the USB 3.0 feature made transferring data to and from the network a fast and easy job. Having two displays attached to the Raspberry Pi 4 whilst supporting 4k makes a nice Thin Client, an all-digital pinball machine, a great companion in the living room, a virtual reality experience, or a small office machine. ⏮

190324-01

Going industry?

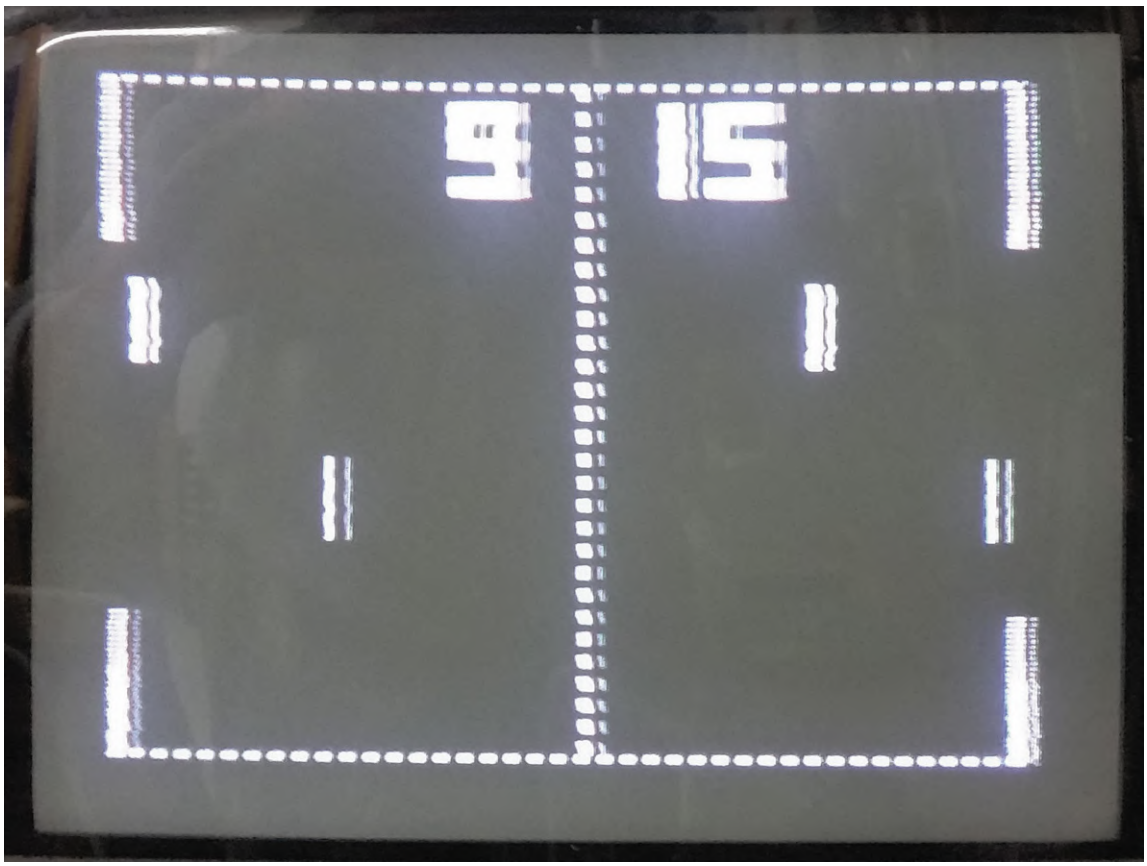
Raspberry Pi 4 is a step forward. When it comes to industrial usage, we are still left with the main issues common to all Pi models. One of these is the micro SD card. If used for a longer time without any optimizations it will simply wear out or break during operation. A SATA port remains on the wish list for that reason. Besides storage, you may have to grapple with the power dissipation problem, which is currently worse than with the Raspberry Pi B+, and calls for better cooling and airflow. For rough environments or hard to reach places the newcomer still isn't the weapon of choice, although the 4 GBytes of RAM permit a few new tricks to reduce the dependency on SD Card mass storage. Those of you active in building digital signage applications, information systems or multimedia-centric applications will surely welcome the Raspberry Pi 4.

Elektor's First Video Game Console

Restart after 42 years

Karl-Ludwig Butte

The construction of Elektor's first video game console using General Instruments integrated circuit type AY-3-8500 appeared in the Summer Circuits 1977 edition. With just a handful of external components and a professional circuit board from the Elektor PCB Service, replicating the project at home was easy — and success almost guaranteed. The AY-3-8500 was not the dawn of video games though, rather the culmination of more than ten years of development. Follow me in the footsteps of the amazing history of video games and experience the recommissioning of an original Elektor circuit after 42 years!



"I … took a mental inventory of all those hundreds of millions of TV sets across the globe that did nothing but play whatever oneway fare the local stations delivered. I had an inspiration - a Eureka! - and Home TV Games were born...a bit early, technically, because low-cost microprocessors weren't available yet and digital I.C.s were still too expensive, so the games had to be relatively primitive."

This quote from Ralph H. Baer describes his first thoughts in the introduction to his book Videogames: In the Beginning (**Figure 1**), which led to the development of video games and

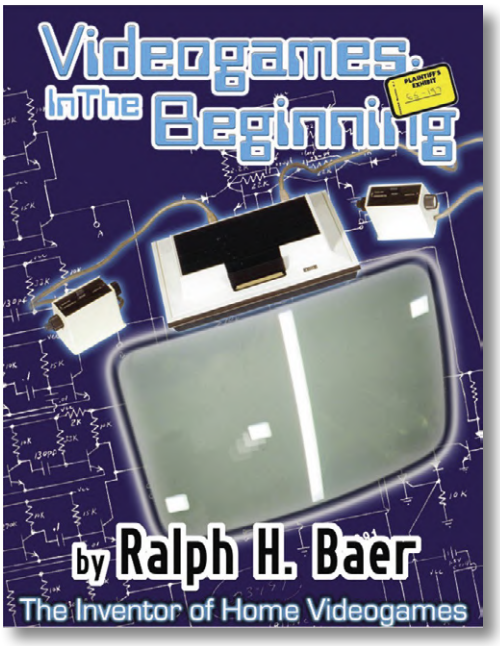


Figure 1: Ralph H. Baer's book *Videogames: In the Beginning*. [1].



Figure 2: Ralph Baer at the Lara Games Award 2009 in Cologne [2].

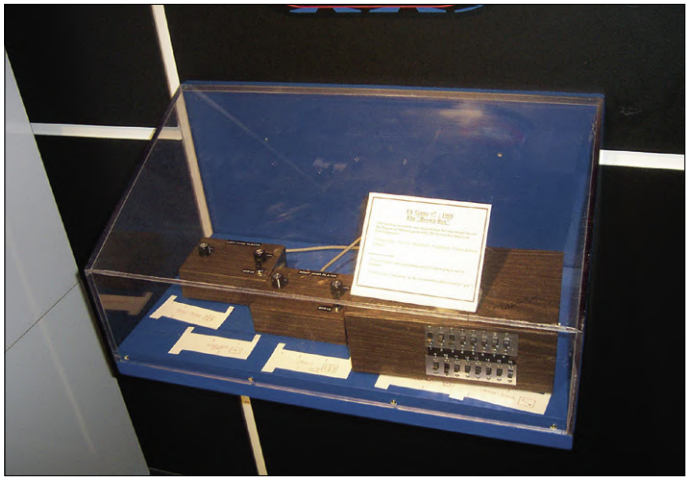


Figure 3: Ralph Baer's prototype 'Brown Box' exhibited at the Smithsonian Institution [3].



Figure 4: 'Odyssey' game console by Magnavox, 1972 (Photo: Evan Amos, Public Domain).

with it, to a multi-billion-dollar industry. That was back in 1966, when England won the Football World Cup, the Soviet Union realized the first unmanned soft moon landing with Luna 9, and Kurt Georg Kiesinger became the third Chancellor of the Federal Republic of Germany.

Ralph H. Baer (**Figure 2**) was born in 1922 in Rodalben near Pirmasens in Germany and with his family fled to America in 1938 just in time to escape the National Socialists because he was Jewish. After the war, he studied at the American Television Institute of Technology in Chicago and graduated in 1948 with a Bachelor of Science degree. In 1955, he joined Sanders Associates, a military electronics supplier, where he worked for 30 years. "But video games have nothing to do with an arms company" you may argue. Baer didn't see it that way. He was enough of a visionary to imagine the same technology he wanted to use as the basis for video games as the basis for simulating and training weapons systems. And — more importantly — he was able to convince his superiors of this vision and got the Go! for this project.

First experiments

Baer set to work on a small budget including a technician. The first experimental circuits were based on tubes like the circuit on page 221 in the book *Videogames: In the beginning* [1] which shows a vertical line with a fixed width on the screen. This could be moved horizontally manually and changed in height at the bottom of the screen. It quickly became clear that the goals set could not be achieved with tubes and that transistors had to be used. On page 223 of the same book, you can admire the first transistor-based design. In 1968, Baer presented the 'Brown Box' to the management of Sanders Associates — a prototype that already offered seven different games, a handset control and a light rifle for shooting games. This prototype is now in the Smithsonian Institution in Washington, DC (**Figure 3**).

Licensees wanted urgently

In order to recoup the cost of the project without having to establish the name ‘Sanders’ in the consumer electronics sector, management decided to look for a licensee for the new video game. This turned out to be more difficult than expected and it took years to reach an agreement with the American TV manufacturer Magnavox. In September 1972 the Magnavox game console called ‘Odyssey’ was launched onto the US market **(Figure 4)**.

Odyssey dropped a bomb! This was followed by further developments with new games, export versions for countries with television standards differing from those in the USA, and imitation products from other companies. Finally, in March 1975, General Instruments released the AY-3-8500 IC (**Figure 5**), closing the circle with the Elektor Summer Circuits edition of 1977.

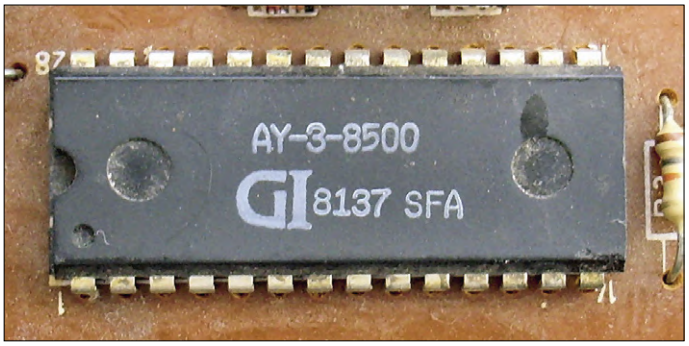


Figure 5: The AY-3-8500 IC from General Instruments, 1975. [4]

Your performance, General Instruments

Founded in 1948, General Instruments (GI) was one of the largest semiconductor manufacturers in the USA. From 1990 to 1993, Donald Rumsfeld, who later became the US Secretary of Defense, was part of the company’s eventful history as CEO. In 1997 it was split into three separate companies, two of which were acquired by Vishay Intertechnology and by Motorola respectivly. Only the cable television division CommScope survived to this day.

Back in 1975 at GI’s Scottish office in Glenrothes, north of Edinburgh, Gilbert Duncan Harrower had combined almost all the circuitry needed for a video game like the Magnavox ‘Odyssey’ into a single integrated circuit: the AY-3-8500 was born! Dr Ed Sacks, General Manager of the GI branch in Hicksville, USA, also heard about this. He had a sample IC flown in from Scotland together with two engineers and invited Ralph Baer to a presentation in Hicksville. With the help of Baer, Coleco became the first commercial customer. ‘Telstar’ was Coleco’s first game console product and was a complete success, which was not a matter of course considering that Coleco was founded in 1932 in New York as the **Co**nnecticut **Le**ather **Co**mpany by the Russian Maurice Greenberg and initially sold leather goods to shoe manufacturers.

Summer Circuits

In November 1975 Elektor published a construction article entitled TV Tennis. This game was equipped with TTL ICs of the ’74 series and several discrete transistors. The massive circuit realized this one game only and the screen design was limited to the ball and two rackets. But this project is worth mentioning because the circuit description so impressively explains the basic principles on which 1970s video games were based. I wasn’t a regular Elektor reader in 1977 yet but I didn’t miss that year’s Summer Circuits issue. While browsing its pages I came across the video game project and was immediately fascinated. The construction info was concise but at least there was a professional circuit board from Elektor’s fabled ‘EPS’ PCB Service. **Figure 6** shows the wiring diagram. Very few external components were required for the AY-3-8500. A comparison with the reference implementation in **Figure 7** [5] also shows Elektor’s close adherence to this reference design. The shooting game with the light rifle was omitted and the version

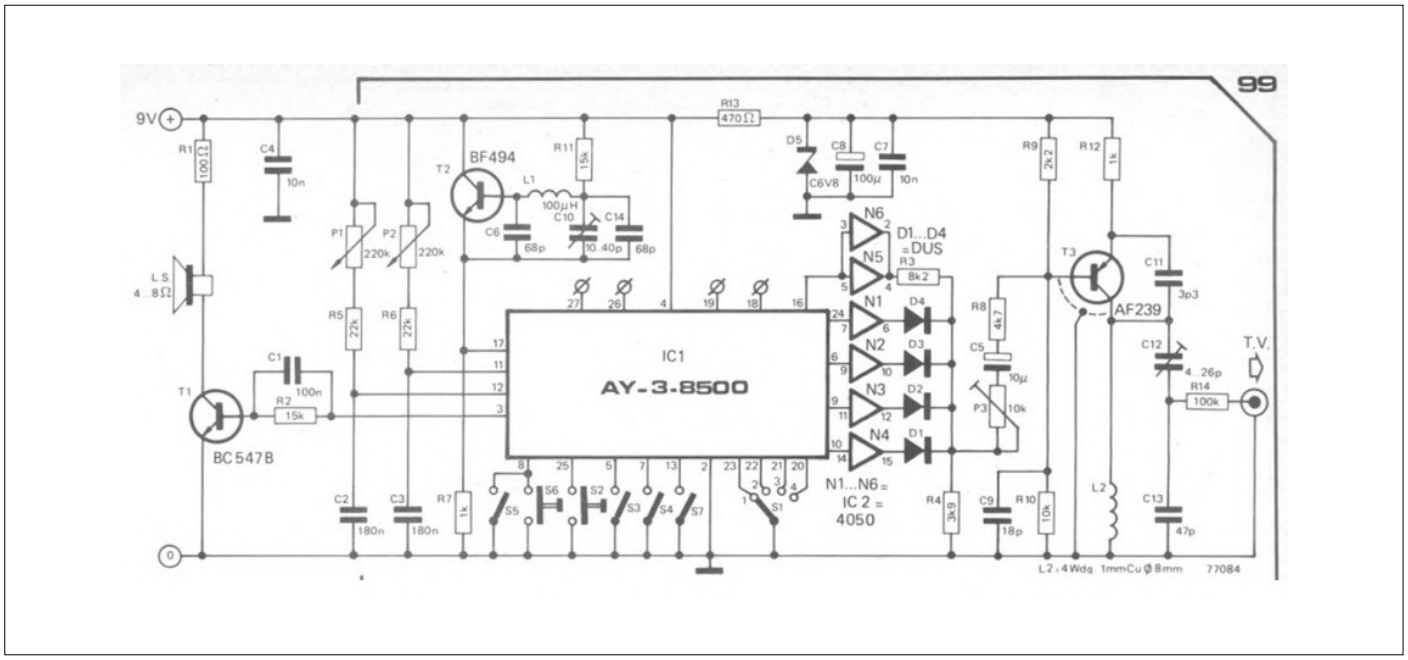


Figure 6: Circuit diagram of the first Elektor video game console from the Summer Circuits 1977 edition.

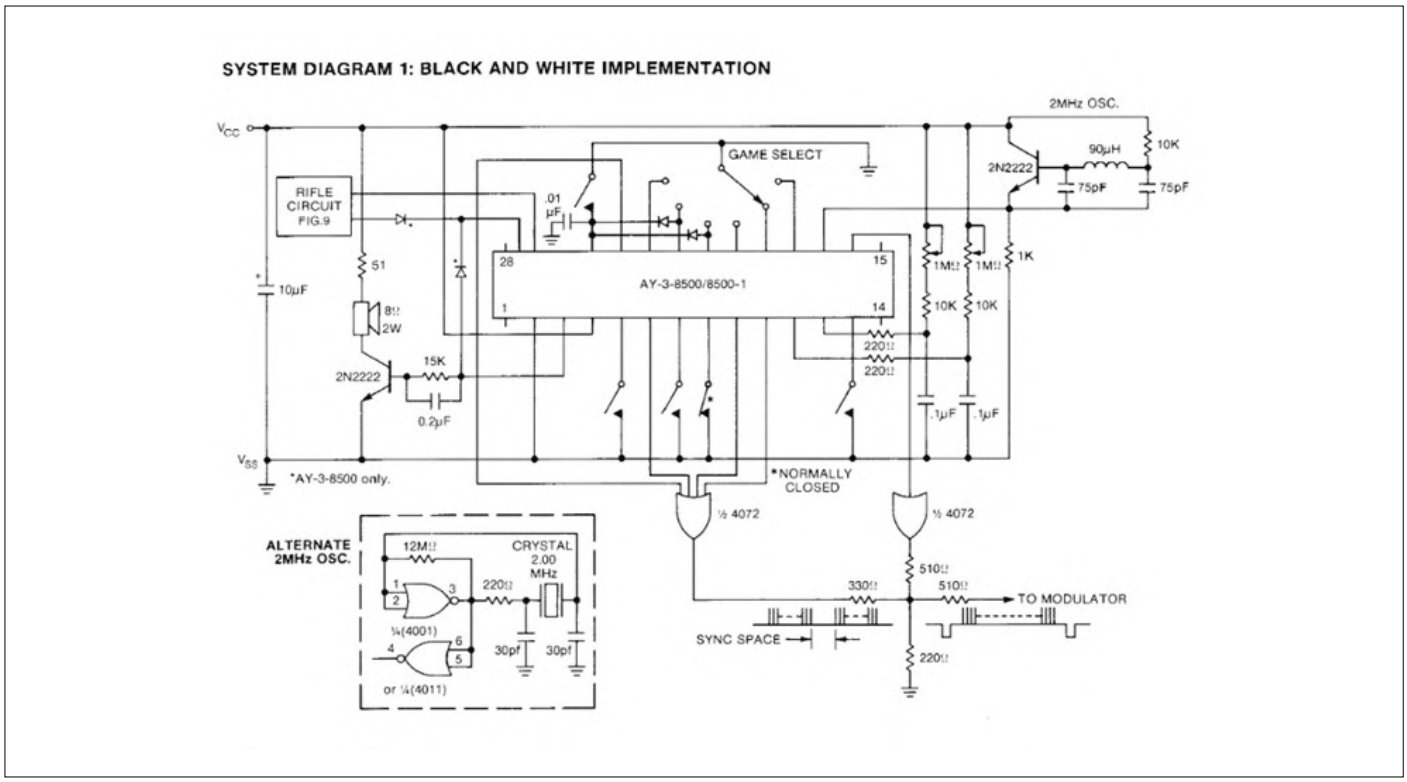


Figure 7: Reference design for the AY-3-8500 (source: General Instruments Corp. – Microelectronics: Gimini TV Game circuits, January 1978).

of the AY-3-8500 was used, which was adapted from NTSC to the European PAL television standard using 625 lines. For the American television standard with 525 lines the AY-3-8500-1 was required.

At approximately 30 Deutschmarks the main IC blew a hole in my limited budget, but I was determined to build such a video game. You can see the result in **Figure 8.** It looks the same as 42 years ago. Would it still work? With apprehension I unscrewed the device to get an impression of the inner condition (**Figure 9**). **Figure 10** shows the assembled board. This all looked very good and promising.

The ceramic capacitor C1 was damaged and got replaced. Fortunately, I also checked the antenna plug: the signal line had slipped out because the fixing screw had come loose. However, the small loudspeaker was a goner, the membrane had largely detached from the holder and the wires leading to the voice coil were torn. Since there was no suitable replacement loudspeaker available, I decided to do without the sound. This was not a big sacrifice, because the constant 'beeping' already got on my nerves back then. After the other minor problems had been solved, I connected the game to the antenna socket of an old tube TV, hooked up the manual controls and the power supply and switched on. The original magazine article stated UHF channel 30 for the sharpest images. After many attempts and a laborious adjustment of the two trimmer capacitors C10 and C12 the best possible picture was in fact obtained on channel 29, as you can marvel at in **Figure 11.** Everything still worked as it did then. However, this required an old tube TV — with newer digital devices you'd have no chance. But then in Elektor 5/2018, Neil Gruending in his Peculiar Parts series took up



Figure 8: The author's device according to the instructions in Elektor's Summer Circuits 1977 edition.

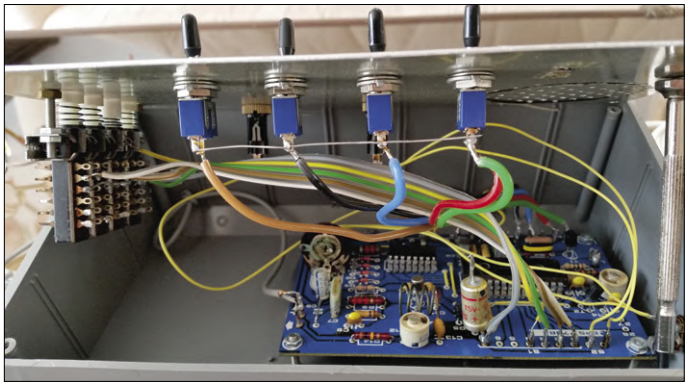


Figure 9: Interior view of the author's device.

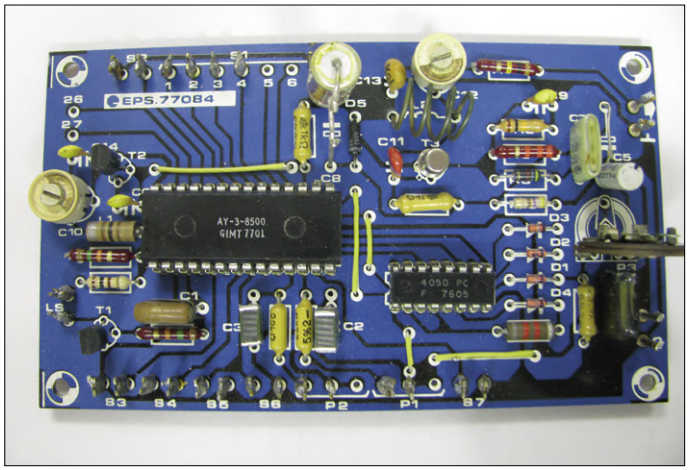


Figure 10: The author's assembled Elektor 'EPS' board.

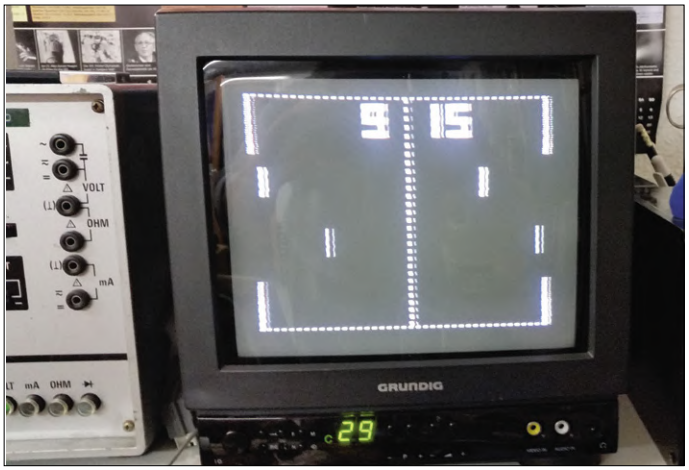


Figure 11: After 42 years the author's project still works on an old tube TV.

exactly this problem and presented a solution that bypasses the circuit's modulator and 'taps' the video signal to enable presenting it to TV's AV input rather than the antenna input.

Ralph H. Baer — millionaire?

In this article we have followed the history of video games by Ralph H. Baer (from the first beginnings in 1966 to the development of the AY-3-8500 IC by General Instruments in 1975 and the Elektor 1977 Summer Circuits publication). TV gaming developed into a global multi-billion-dollar market that still shows considerable growth today. Conversely, does it mean that Ralph H. Baer became a millionaire or billionaire? No. To invent something is easy, as he said ([1], p. 110), but then to sell this invention is much more difficult. Even his recognition as the father of video games was challenged by Atari founder Nolan Bushnell. However, as Baer was used to documenting everything legally, he won a patent dispute against Bushnell in 1982. Among other evidence he was also able to present a visitor list of a Magnavox promotional event for the 1972 'Odyssey' video game console which was also signed by Nolan Bushnell as a participant. Bushnell had got to know the games implemented in 'Odyssey' at the Magnavox event before the development of his arcade device called 'Pong'.

In 2005 Ralph Baer donated all remaining hardware and documentation of his video game development to the Smithsonian National Museum of American History [6] and came to Berlin in 2006 as patron of a special exhibition at the Computer Games Museum [7]. The video interview filmed for this occasion can be found on Spiegel Online [8]. Ralph Baer died on December 6, 2014 at the age of 92 in Manchester, New Hampshire. His ideas, constructions and their modern out-developments still bring a lot of fun today worldwide. ⏮

180574-F-02

ESTD 2004

www.elektor.tv

Retronics is a regular section covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com

Web Links

[1] Baer, Ralph H. Videogames: In the beginning, Rolenta Press, Springfield, NJ, USA, 2005: https://bit.ly/33PHNeT

[2] Ralph Baer at the Lara Games Award 2009 in Cologne (Author: Michael Schilling, Creative-Commons-Lizenz): https://de.wikipedia.org/wiki/Datei:Ralph-Baer.jpg

[3] The 'Brown Box' prototype of the Magnavox 'Odyssey' signed by creator Ralph Baer (author: George Hotelling, Creative Commons Attribution-Share Alike 2.0 Generic Lizenz): https://commons.wikimedia.org/w/index.php?curid=2859746

[4] General Instrument AY-3-8500 - antic pong TV game (Author: Atreyu. Creative Commons Attribution 3.0 Unported Licence): https://de.wikipedia.org/wiki/Datei:AY-3-8500.jpg

[5] General Instruments Corp. – Microelectronics: 'Gimini TV Game circuits', January 1978, p. 4 : https://web.archive.org/web/20120316160455/http://www.pong-story.com/GIMINI1978.pdf

[6] Smithsonian National Museum of American History: https://invention.si.edu/ralph-h-baer-papers

[7] Computer Games Museum Berlin: www.computerspielemuseum.de/1262_Unser_Schirmherr.htm

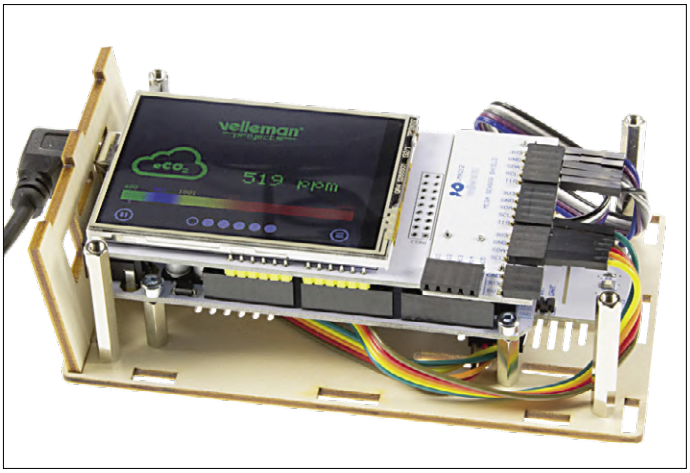
[8] Video Interview on Spiegel Online: www.spiegel.de/netzwelt/games/ralph-baer-erfinder-der-spielkonsole-ist-tot-a-1007139.html

ELEKTOR STORE HIGHLIGHTS

Earth Listener, Function Generator, Lab Power Supply

The climate — everyone talks about it, but those in power do nothing about it. However, with one of the products discussed here, you can at least monitor the microclimate in your home or office and do something about it if necessary.

Velleman Earth Listener



To clear up any misunderstandings in advance: this device does not 'listen' to the earth... The product with the somewhat odd name 'Earth Listener' is a kit from Velleman that allows you to keep an eye on the microclimate in your home or office.
It is an Arduino-based sensor module that measures temperature, air pressure, humidity, CO_2 concentration and volatile organic compounds, and displays the measured values on a TFT touch screen and/or stores them on an SD card. And to cap it all, the circuit also includes a sensitive lightning detector with a range of approximately 20 km.
The Arduino software of the Earth Listener is open source; the sensor shield offers the possibility to add six additional sensors. A lot of possibilities for such a small box - but you still have to build it.
Assembly is not a problem: nothing (!) needs to be soldered, and to tighten the bolts, an Allen key is included. In half an hour or so the assembly should be done.
The firmware has already been pre-programmed; after switching on the power supply (if everything went well, and shouldn't it?) a message appears on the display saying that three sensors have been detected. Built into the laser-cut wooden housing, the whole thing looks very neat.
The source code of the Earth Listener sits on GitHub and is therefore available to anyone wanting to work with it. Meanwhile version 3.4 of the firmware can be downloaded.
In the Elektor Lab, the combined thermo/hygro/barometer annex gas and dust particle sensor happily functions as an electronic referee in discussions about whether or not to open a window...

@ WWW.ELEKTOR.COM

→ Velleman Earth Listener (kit of parts)
www.elektor.com/18917

Siglent SDG1032X function generator

In the previous issue of Elektorlabs we looked at a DDS signal generator from JOY-iT. The Siglent instrument that we would like to introduce to you here is more expensive, but it also has a few additional features to offer.
One of the main criticisms of the JOY-iT generator was the plastic housing, which gives a bit of a cheap impression. The Siglent SDG1032X on the other hand is housed in a solid metal casing with rubber bumpers and can therefore withstand a heavy impact.
At the front of the generator there are two BNC sockets for the two signal outputs; at the back there are another LAN and USB connection and three BNC connectors for the (built-in) frequency counter, for the 10-MHz reference signal, and for optional synchronization with other measuring devices.
The control panel consists of a large rotary knob, a numeric keypad an and a number of function keys. Below the display there are 6 other soft menu keys, the current function of which is shown on the display.
Speaking of the display, this is a clear 4.3-inch type that shows all important settings at a glance, as well as the shape of the set output signal.
The instrument (with a bandwidth of 30 MHz and an output





amplitude of 20 Vpp) offers a wide range of analogue and digital modulation options: AM, DSB-AM, FM, PM, FSK, PSK and PWM. In addition, the Siglent generator offers the possibility to add harmonics of your choice and taste to the output signal. This encourages experimentation! Another great thing is that the sweep and burst function can use not only a sine wave, but also other waveforms.

With square waves and pulses (with correctly terminated BNC cable) the rise time is about 4.5 ns, which results in waveforms that are a delight to the eye. It is even possible to adjust the steepness of each edge.

For everyone who is serious about electronics, the Siglent SDG1032X is worth the investment of just under € 350 (at the moment of writing).

@ WWW.ELEKTOR.COM

→ Siglent SDG1032X Function Generator
www.elektor.com/17891

JOY-iT DPS505 programmeerbare labvoeding

JOY-iT's programmable lab power supply type JT-DPS5005 is a small module, but with 50 volt output voltage and 5 amps output current it's able to deliver a lot of power: 250 watts maximum! With the JT-DPS-Case, JOY-iT delivers a custom enclosure. All required larger and smaller parts are supplied in separate bags: screws, nuts, plastic spacers and nuts, connectors, spade terminals, a fan, cables, a switch and a small circuit board that supplies the power to the fan.

Of course, the module can be installed and used in a different way, but with this housing you have everything you need to make it a neat and safe power supply. For this discussion, the module is built into the JOY-iT housing.

However, the input voltage requires another power source. Depending on the application, an input voltage of 6 V to 55 V can be used; this external power source must be able to supply at least 5 A (for the maximum output current of 5 A of the module). For a first test it is best to use a (different) lab power supply.

Here we will limit ourselves to the most important possibilities and properties. The main menu shows the set voltage and current limit as well as the current values after switching on the output. Finally, the input voltage is also shown.

All settings are made via a combination of the three pushbuttons on the left side of the front and the combined rotary/push button. It will take some getting used to the order in which you have to operate the buttons to change the correct settings quickly. However, in the manual everything is clearly explained.

For our first test we used a dirt cheap AC/DC power supply from Mean Well type RSP-320-48 capable of delivering more than enough power and the maximum allowable voltage. It can deliver 6.7 A at 48 V and the output can be set even higher with a potentiometer, to just over 55 V; even then it can still supply enough power. The combination then forms a 50 V/5 A lab power supply at a very interesting price. ⏮

180689-F-02

@ WWW.ELEKTOR.COM

→ Joy-iT JT-DPS5005 Programmable Lab Supply
www.elektor.com/18708

→ JT-DPS Case
www.elektor.com/18709

HOMELAB PROJECT

8-Channel Timer with OLED Display

Revised with easier to obtain parts

By **Willem Tak** (Netherlands)

In the March & April 2018 issue of Elektor we published a multitimer in two versions: one with eight independently programmable channels and another with sixteen channels. Both designs enjoyed good popularity but the 8-channel version with OLED display suffered from a major problem: some crucial components were already difficult to obtain at the time of publication and by now are rarer than hens' teeth.

The author has also recognized this problem and modified the original design [1] so that it can be built with currently available components. As with the original design, we're not going to discuss the use of the timer — it has so many features that we could fill half of this edition with it. The operating instructions and the updated software can be downloaded (as always, free of charge) from the project page accompanying this article [2].

Hardware

Figure 1 shows the new schematic. This differs from the original in some essential details. We will deal with these changes one by one below.

Controller

In the first place, the heart of the circuit is formed by a different controller: a PIC18F46K22, which is now clocked at 11.0592 MHz. This frequency got selected so that the GPS module (see below) can be read out without any problems. The 18F46K22 was selected because it has a large RAM memory. Attention: this controller no longer runs on +5 V, but on 3.3 V!

Display

The originally used OLED display proved difficult to obtain right from the outset and is now almost impossible to find anywhere. That's why another display is now being used: a 128×64 pixel graphic copy of MIDAS, which is available from Farnell (o/n 2769702). Note: this display has different connections than the original model! As in the original circuit, the brightness is controlled by a potentiometer connected to analogue input A0 of the controller.

GPS time

The original GPS receiver, which was used to set the internal clock and check it for synchronization, also turned out to be scarce in the retail market. Instead, a NEO-7M GPS module is now being used, which is available at least on eBay and at Amazon(.de) for little money. In the Elektor Store the NEO-6M is available which is probably usable as well (although we didn't check that).

Power supply

In the new design, there are no longer any parts that run on 5 V, so the 5-V DC/DC converter (TSR12450) is no longer needed. Apart from the relay driver and the relay proper, all components now happily work from 3.3 V.

The timer requires an input voltage of 12 V (from a power adapter); IC4 (a TSR12433) turns this into a stable 3.3 V at relatively low loss. The 12-V input voltage is only needed for relay driver IC3 and for the relay itself. Nothing has changed in that part of the circuit.

Software

The software for this project was adapted to match the new component selection (display, controller, GPS module). However, it would take too much space here to go into this assembler-written program; the clearly commented software is available for you as a free download on the project page of this article. ⏮

(180171-03)

Web Links

[1] Original Multitimer article: www.elektormagazine.com/150188

[2] Project page for this article: www.elektormagazine.com/180171-03

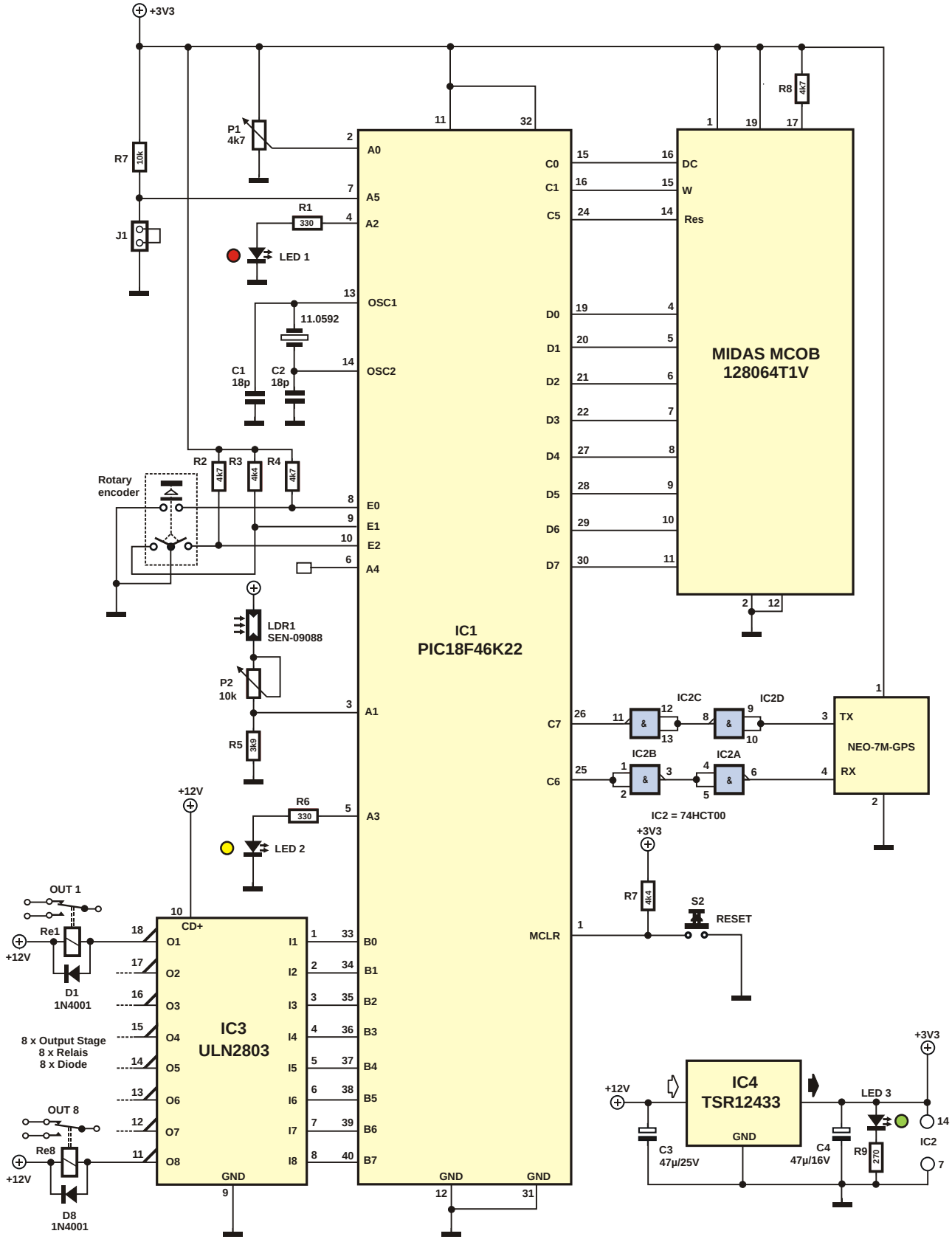


Figure 1. The schematic of the updated 8-channel timer differs in some important details from the original version.

@ WWW.ELEKTOR.COM
→ NEO-6M GPS receiver : www.elektor.com/18723

welcome in your ONLINE STORE

EDITOR'S CHOICE



Weller WT 1014 Digital Soldering Station (90 W)

The WT 1014 is a soldering iron with a pleasant grip and it works well although a finer tip for the more delicate (SMD) work is no superfluous luxury. There are many types and sizes of soldering tips available at a reasonable price. From a manufacturer such as Weller you can expect that all the (spare) parts will be available for many years. All three, the holder, the station and the soldering iron, impart reliability and the ergonomics has been well though through. Admittedly, this set is not particularly cheap, but with the WT 1014 you will have an excellent soldering station on the laboratory table that's sure to give years of service.

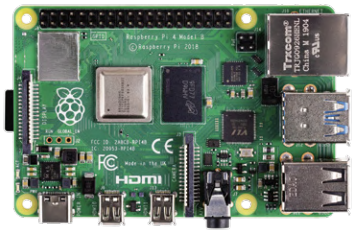
Luc Lemmens
(Elektor Labs)



www.elektor.com/18649

Elektor Bestsellers

1. Raspberry Pi 4 B
   www.elektor.xx/rpi4



2. Camera Projects Book
   www.elektor.xx/18943
3. JoyPi
   www.elektor.xx/18770
4. RTL-SDR
   www.elektor.xx/18949
5. PCBite Kit
   www.elektor.xx/18818
6. Elektor SDR Shield 2.0
   www.elektor.xx/18515
7. Fan SHIM
   www.elektor.xx/19039
8. iFixit Manta Precision Bit Set
   www.elektor.xx/19049

The µArt Kit

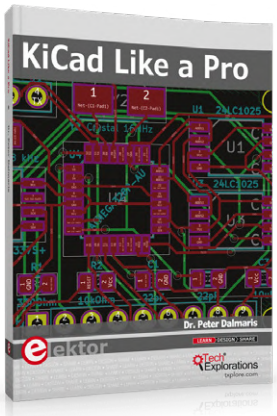


µArt is an isolated USB to UART-TTL adapter which you can use any time you want to connect a UART to your computer's USB port, regardless of voltage levels, electric potentials, and communication speed, and whether it is used for data exchange or for programming Arduinos, ESPs, or other devices.

Member Price: £58.95 • €64.76 • $71.95

www.elektor.com/19029

KiCad Like a Pro



This book will teach you to use KiCad. Whether you are a hobbyist or an electronics engineer, this book will help you become productive quickly, and start designing your own boards. This book takes a practical approach to learning. It consists of four projects of incremental difficulty and recipes.

Member Price: £32.95 • €35.96 • $39.95

www.elektor.com/18822

SDR Hands-on Book



Elektor's SDR-Shield (SKU 18515) is a versatile shortwave receiver up to 30 MHz. Using an Arduino and the appropriate software, radio stations, morse signals, SSB stations, and digital signals can be received. In this book, successful author and enthusiastic radio amateur, Burkhard Kainka describes the modern practice of software defined radio using the Elektor SDR Shield. He not only imparts a theoretical background but also explains numerous open source software tools.

Member Price: £23.95 • €26.96 • $29.95

www.elektor.com/18914

Anycubic Photon S

UV Resin DLP 3D Printer



NEW

The new Anycubic Photon S features Z-axis Dual Linear Rail which improves the stability of Z-axis and stands against any wobble that might occur. The UI is also a vast improvement over the previous models with a clear menu structure and large, legible icons and buttons.

Printing on the Photon S is done by slicing of 3D models using the bundled Photon Workshop software as the proprietary .photons file. The file is loaded is on onto a USB pen drive and then plugged into the side of the printer. The printing volume is increased to a total of 115 x 65 x 165 mm. It also produces less noise during the print making it much quieter.

Member Price: £447.95 • €494.10 • $545.95

www.elektor.com/19052

iFixit Manta Precision Bit Set

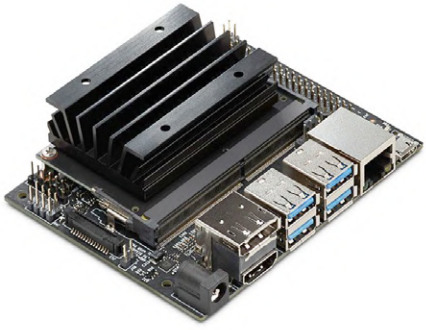


The Manta Precision Bit Set has a very wide variety of bits and driver heads that will get you covered in all your repairs and projects. It comprises of iFixit's 1/4" Drive and 4 mm Drive premium aluminum screwdriver handles with magnetic bit sockets, knurled grips, and swivel tops—plus 112 precision driver bits.

Member Price: £48.95 • €53.96 • $59.95

www.elektor.com/19049

NVIDIA Jetson Nano Developer Kit



The NVIDIA Jetson Nano Developer Kit makes the power of modern AI accessible to makers, developers, and students. When you think of NVIDIA, you probably think about graphics cards and GPUs, and rightfully so. Nvidia's track record guarantees that the Jetson Nano has enough power to run even the most demanding of tasks.

Member Price: £121.95 • €134.96 • $148.95

www.elektor.com/19001

Camera Projects Book



The book explains in simple terms and with tested and working example projects, how to configure and use a Raspberry Pi camera and USB based webcam in camera-based projects using a Raspberry Pi.

Member Price: £23.95 • €26.96 • $29.95

www.elektor.com/18943

The Elektor Community

82	203637	1040	546	235332
Countries	Enthusiastic Members	Experts & Authors	Publications	Monthly Visitors

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, ESP32, IoT, 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.elektor.tv

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

- ✘ 6x Elektor Magazine (Print)
- ✔ 6x Elektor Magazine (PDF)
- ✔ Access to Elektor Archive (Thousands of Articles)
- ✔ Access to over 1,000 Gerber files
- ✘ Elektor Annual DVD
- ✔ 10% Discount in Elektor Store
- ✔ Exclusive Offers

www.elektor.com/green

GOLD
€7.58 per month
£5.50 / US $8.42

- ✔ 6x Elektor Magazine (Print)
- ✔ 6x Elektor Magazine (PDF)
- ✔ Access to Elektor Archive (Thousands of Articles)
- ✔ Access to over 1,000 Gerber files
- ✔ Elektor Annual DVD
- ✔ 10% Discount in Elektor Store
- ✔ Exclusive Offers

www.elektor.com/gold

FREE

- ✘ 6x Elektor Magazine (Print)
- ✘ 6x Elektor Magazine (PDF)
- ✘ Access to Elektor Archive
- ✘ Access to over 1,000 Gerber files
- ✘ Elektor Annual DVD
- ✘ 10% Discount in Elektor Store
- ✔ Elektor weekly e-zine
- ✔ Exclusive Offers

www.elektor.com/newsletter

facebook.com/ElektorLabs

twitter.com/Elektor

instagram.com/elektorlabs

BUILD THE FUTURE WITH DISTRELEC – IT STARTS WITH YOU

BROWSE OUR SELECTION OF STARTER KITS DESIGNED TO EMPOWER THE NEXT GENERATION OF ENGINEERS

BROWSE ALL KITS

distrelec.biz/education

+371 66165147



RND HIGH QUALITY, LOW PRICES, SMART RESULTS.

> **Soldering bundle: soldering station and solder fume extractor 100w 500°c 230v, RND lab**

Leading distributor for
Automation & Electronics

DISTRELEC
Distribution with a difference

Hexadoku The Original Elektorized Sudoku

Traditionally, the last page of Elektor Magazine is reserved for our 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 five 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 five Elektor Book Vouchers worth **$60.00 / £45.00 / €50.00 each**, which should encourage all Elektor readers to participate.

Participate!

Ultimately 26 November 2019, 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 in edition 5/2019 (September & October) is: **4A0FC**.

The €50 / £40 / $70 book vouchers have been awarded to: Jean-Paul Lagaisse (Belgium), Kenneth Steelman (USA); Harald Metzen (Germany); Darryl Jewiss (New Zealand); and Hans van Zanten (Netherlands).

Congratulations everyone!

	C	9		7							8		F	4	
0	A		D				6	1				3		8	5
5	B		F			9			0			1		C	2
8	E			F	A	C			4	6	5			9	D
4		5			E	6			8	D			A		C
1	7	8	B	3		4	C	E	A		9	F	D	0	6
	D				9	8			6	1				3	
	8				0	E			5	A				F	
D	5	3	A	6		F	9	4	1		2	7	8	B	0
F		6			5	7			B	0			C		E
A	1			9	F	2			7	3	4			E	B
B	4		7			A			2			9		D	F
E	6		8				1	A				C		2	7
	9	C		0							E		5	1	

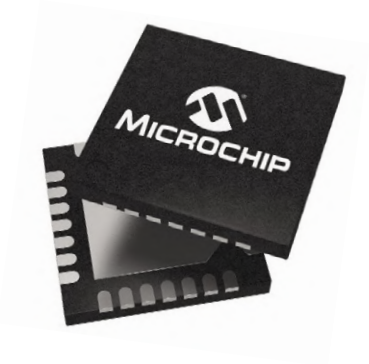
8	3	D	0	C	9	E	1	5	6	A	7	B	2	F	4
4	6	5	E	B	2	3	A	0	D	1	F	C	8	7	9
7	A	C	1	4	8	5	F	9	B	2	3	D	E	6	0
9	F	2	B	0	6	7	D	C	E	4	8	1	3	5	A
A	4	8	F	D	3	0	6	B	7	C	1	2	5	9	E
C	D	3	9	8	1	F	4	2	A	5	E	6	7	0	B
6	1	B	7	2	5	C	E	D	9	F	0	4	A	3	8
E	5	0	2	7	A	9	B	3	8	6	4	F	1	C	D
B	7	9	8	3	4	A	0	F	C	D	2	5	6	E	1
D	C	E	6	5	7	8	2	1	3	9	B	0	4	A	F
1	2	F	4	6	B	D	9	A	0	E	5	3	C	8	7
3	0	A	5	F	E	1	C	7	4	8	6	9	B	D	2
F	8	4	A	1	C	2	3	E	5	0	D	7	9	B	6
0	E	6	3	9	D	B	5	4	1	7	A	8	F	2	C
2	B	7	C	A	0	4	8	6	F	3	9	E	D	1	5
5	9	1	D	E	F	6	7	8	2	B	C	A	0	4	3

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.



World-Class Analog from a Microcontroller Leader?

If You Only Know Microchip as an MCU Supplier, We're About to Blow Your Mind



Microchip's success story wouldn't be complete without including our analog solutions. Our history as a leading solution supplier providing comprehensive design support and a broad product portfolio doesn't only include our microcontroller products.

We also offer high-performance, easy-to-implement linear, mixed-signal, power management, thermal and interface products. When combined, Microchip's extensive portfolio can be used in numerous applications with various performance requirements.

You will have the power, flexibility and confidence to choose the right solution for your design, regardless of design constraints. Take advantage of our experience and complete system solutions to save time and simplify your design effort.

Start your own success story at www.microchip.com/Real-Analog

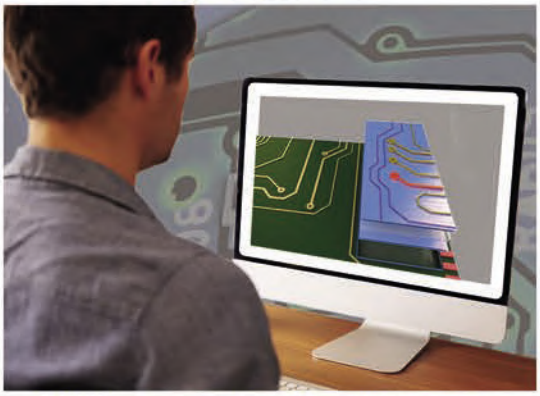
MICROCHIP

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2019 Microchip Technology Inc. All rights reserved. DS20006063A. MEC2243A-ENG-08-19

PROTEUS DESIGN SUITE

Advanced PCB features for professional board design

DESIGN ROOMS



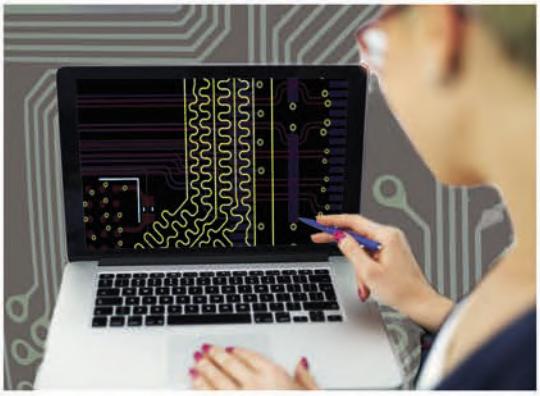
Set design rules that apply in user specified areas of the PCB.

LAYER STACKUP



Control the layer stackup and drill ranges for smarter routing.

SERPENTINE ROUTING



Easily length match tracks against each other or to a target distance.

DESIGN VARIANTS



Edit the fitted status of parts or re-place with pin compatible alternatives.

The Proteus Design Suite provides advanced features at an affordable price. Try it today!

Visit: www.labcenter.com

Tel: +44 (0) 1756753440
E-Mail: info@labcenter.com
youtube.com/c/LabcenterElectronicsLtd

Labcenter Electronics www.labcenter.com