elektor MAG
design > share > sell
502
EDITION 4/2020
JULY & AUGUST 2020
ELEKTORMAGAZINE.COM

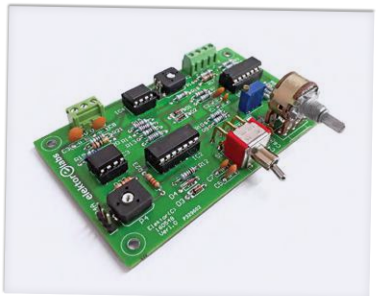
# Summer Edition

with extra many projects
for starters and pros

How To Take (Successful) Photos of Electronics 44

great info for makers!


Elektor Summer Giveaway: Participate and Win! 18



Simple Function Generator 20

all-analogue electronics

Review: Joy-iT RD6006 Benchtop PSU 28

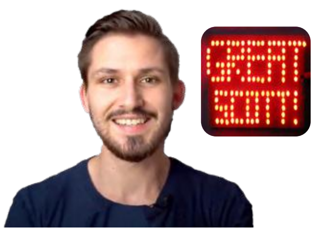
Lab Tours: The Kinetic Clock Maker 76

## In this edition



and much more!

Artificial Intelligence for Beginners (2)
Neural Networks with Python 110




YouTube-Celeb GreatScott!
builds a LoRa Alarm System 24

Home Information System
Using Windows on a Raspberry Pi 8

# COLOPHON

Elektor Magazine,
English edition

Edition 4/2020
Volume 46, No. 502
July & August 2020

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

www.elektor.com
www.elektormagazine.com

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

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

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

**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.




EDITORIAL

# Extra Circuits To Go

The Corona crisis has a firm grip on most of us and looks here to stay for quite some time. But as I write these lines, at least there is a glimmer of hope on the horizon — the social contact restrictions are being eased, and good news is also coming from the business world, including the electronics industry. Of course, we all hope that things will continue to improve and that the *electronica Fast Forward Award powered by Elektor* can also take place as planned at the world's largest electronics trade fair in November 2020 — at the moment the outlook is favourable, which we are delighted about together with the partner companies and the participating start-ups, 40 of which have already registered.

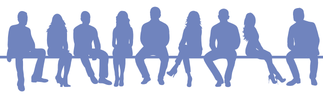
Still, the months in which we had to cope with major uncertainties have not gone by without leaving their mark. For example, we had to reduce laboratory operations because the health of our colleagues was our first priority. We also had to keep a general eye on our expenses - not least owing to a sharp drop in newsstand sales and cautious advertisers reducing their marketing budget to zero for the crisis months (which we really can't blame them for). We therefore decided early on to adjust the concept of the current edition a little. However, my colleagues and I have been working hard over the last few weeks to ensure that it's by no means an "economy issue". In keeping with the summer – and in the long tradition of Elektor's *Summer Circuits* edition – we now present a magazine with an extra dose of small projects and circuit concepts that are suitable for beginners as well as advanced users. There are many purely analogue circuits not requiring a PC or programming code; while friends of Raspberry Pi, Arduino & Co. will also get their money's worth. We also have a special treat: a bunch of valuable tools for developers to grab (see page 18).

I hardly need to mention here that the crisis also offers opportunities — surely you will follow us online at www.elektormagazine.com, where we have frequently reported on initiatives from the world of electronics that are helping people in the worst-hit countries. Among the most important international electronics magazine for professionals and hobbyists, Elektor is acutely aware of its key position in the transfer of knowledge and information. You can read more about the *Elektor Helps!* initiative on page 6.

Stay with us and stay healthy!

*Jens Nickel*
*International Editor-in-Chief, Elektor Magazine*


## The Team

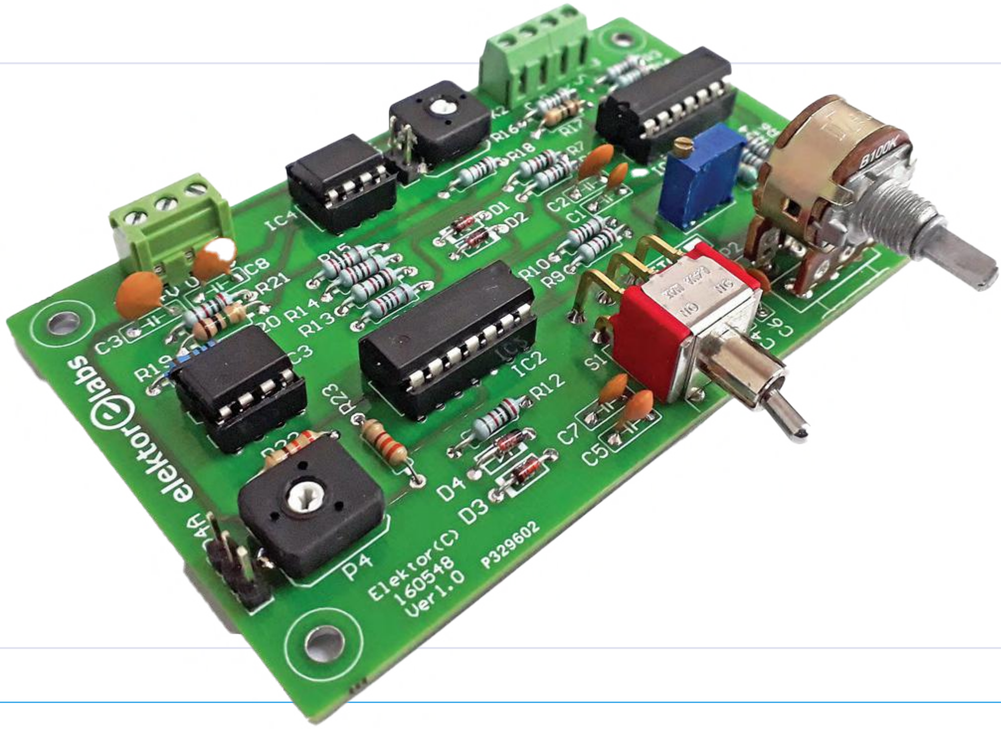


| | |
|---|---|
| International Editor-in-Chief: | Jens Nickel |
| Editor-in-Chief, English edition: | Jan Buiting |
| Membership Manager: | Denise Bodrone |
| International Editorial Staff: | Eric Bogers, Rolf Gerstendorf, Denis Meyer, Dr Thomas Scherer, Clemens Valens |
| Laboratory Staff: | Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens, Jan Visser |
| Graphic Design & Prepress: | Giel Dols |
| Publisher: | Don Akkermans |

# THIS EDITION

Volume 47 – Edition 4/2020
No. 502 – July & August 2020

Labs Project
Simple Function Generator
20




## Regulars



## Features



## Projects

# Next Edition

**Elektor Magazine Edition 5/2020 (September & October)**
As usual, the next issue is packed with circuits, projects, foundations and tips and tricks for electronic engineers.

**From the contents:**
- Nixie Bar Graph Thermometer Update
- HV Power Supply with Curve Tracer
- Home Automation with Home Assistant
- AI for Beginners (3): Develop your own Neural Network
- Open-Network Weather Station: Software
- Tube Signal Generator
- FFT on the Maixduino
- Programming PICs with Assembler
- NeoPixel Jewel Control with ESP8266

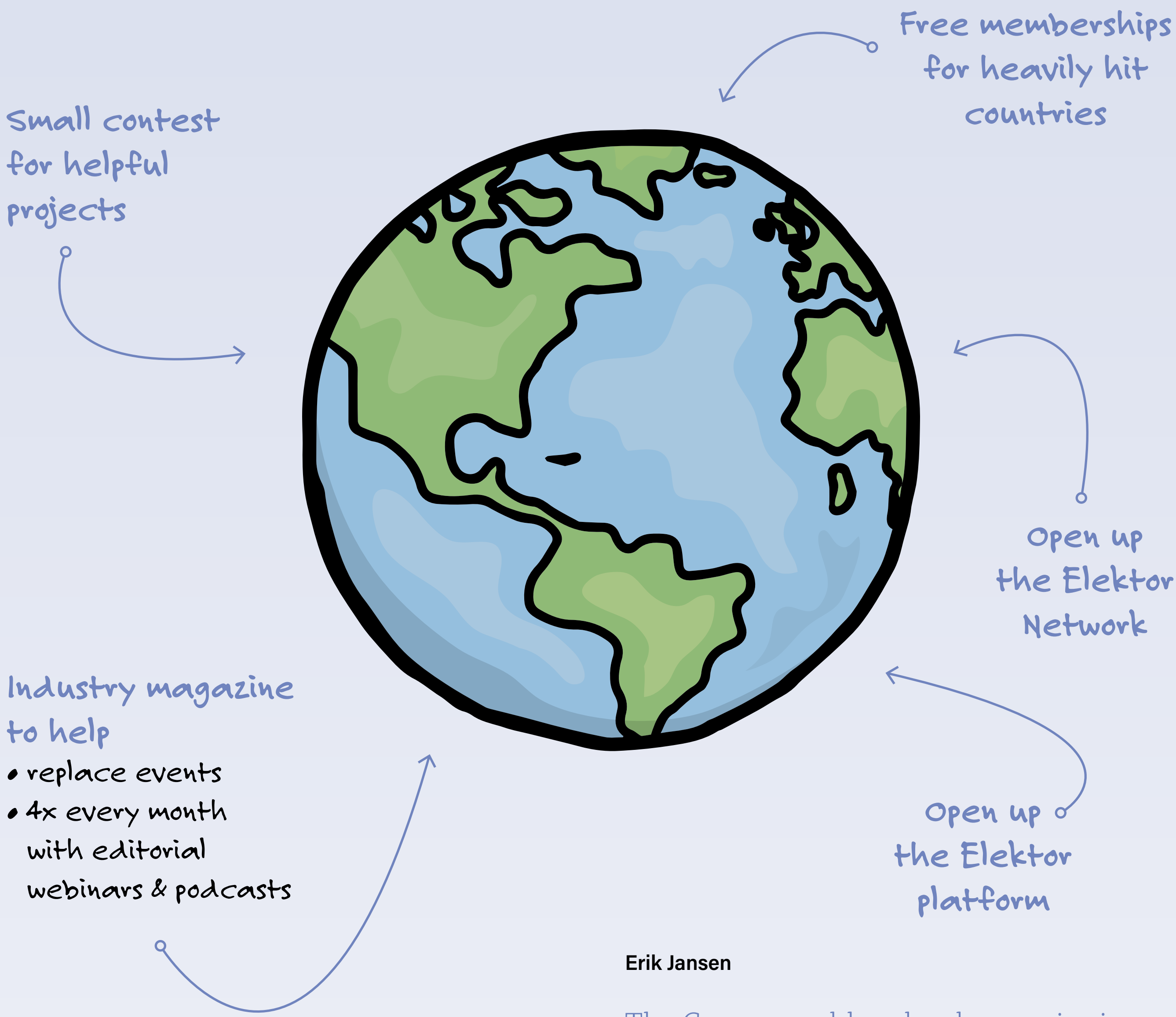
**And much more!**

Elektor Magazine edition 5/2020 covering September & October is published around 3 September 2020. Arrival of printed copies with Elektor Gold Members is subject to transport. Contents and article titles subject to change.

Check out the new LABs site

# elektor Helps

Electronics in challenging times




Erik Jansen

The Corona problem has been gripping the world. Of course, at first it's a disaster unlike any one of us has seen before. Personal tragedies, countries hit hard and a global impact on the economy and many industries. The electronics industry too, cannot escape a lasting impact.

Elektor's business is also affected in various places, and it requires an alert and creative approach to get through this time together and without serious harm. Our entire team is dedicated trying to manage this as well as possible and with as little setback as possible for our members. But we feel we are going to succeed!

Times like these also show us beauty. A strong sense of solidarity and the many initiatives that come from it can be seen nearly everywhere. Big and small heartwarming projects help to make the world a little better. And despite the fact that at Elektor we have to pay close attention to our own issues, Elektor is keen to make a contribution as a team and as a company. If we can't do that by investing ourselves, then we'd like to do so by opening up our help to the many initiatives we see around us.

Under the label **Elektor Helps**, we therefore open up our platforms to anyone who can use our tools, our platform and our knowledge logically and meaningfully for a good cause. Free of charge.

## Elektor Helps: Content

As a business we have noticed that in these times it is not self-evident that our content and services are easy to come by. Due to various lockdowns, limited postal services, and, of course, new and different priorities, not every engineer has access to our content at the moment.

So, even though more and more countries are now lifting some of the biggest limitations, we feel we can still contribute with some relief. That's why, again under the label Elektor Helps, we are offering to help engineers and makers in the most affected countries with a fitting solution ranging from *reduced shipping costs* to a *free full Elektor Green membership* for those in difficult circumstances. And we will continue to do this for as long as the virus makes daily life difficult, at least three months but for a year or more if necessary.

In each country where we launch the Elektor Helps program, we will offer a fitting special offer **available exclusively to our paying members** such as an unconditional voucher for the Elektor Online Store.

For our industry partners it is also a difficult time now that important events are closing their doors and it is difficult to plan ahead in terms of marketing and communication. We are currently adjusting our approach in this area as well. It doesn't make much sense now to plan an edition linked to an event. However, it does make sense to offer a platform to these events and their visitors to flesh out the message that should have taken place there in an alternative way.

The upcoming editions of our *Elektor Industry Magazine* will therefore be tackled more frequently and in a multimedia way. A monthly theme, prompted by the current needs and situation, will be designed online, as an e-publication, but also in the form of editorial webinars and podcasts. Moreover, we make all this content freely available in our growing community and beyond. The special *Elektor Helps - Electronics in challenging times* electronic editions, over the next four months, will be actively sent to at least 120,000 engineers worldwide.

## Elektor Helps: Projects

Elektor has, of course, been helping engineers with projects on its own online LAB platform for years. It is a place where a wonderful diversity of projects come together. This forms the core of our existence and we are not going to change that, of course.

We feel that in the Elektor LABs environment too there are many people that have put their capabilities to work since the lockdowns resulting in great new ideas that may be helpful for others. Whether for fun or for society, we want to encourage our community to share these projects on the Elektor LAB platform. That doesn't mean they have to be a respirator or a ventilator. It may also be a nice little project that is fun to make and to share with a large group of like-minded people, quite a few of whom are currently at home against their will. Something fun to do can help too!

We are, of course, well aware that there are already many initiatives out there that call on engineers to use their knowledge and skills for good causes. Still, Elektor wants to give a small push here. Not to take away the spotlights of other good initiatives, but rather to put more attention on them, or to give that last bit of motivation to rethink that one project that is already on the Elektor LAB again. You can check out our little contest here!

## Elektor Helps: Network

As a company that has been active for sixty years (celebrating fifty years in Germany this year!), Elektor has of course built up a great network. Not only with purely business ties, but more than once we hear from top people at very large companies that their passion for electronics started with reading Elektor. And while every company always has the flexibility to start its own initiative, a number of parties in the background have already indicated that they would like to offer help in an appropriate way for selected initiatives within the Elektor Helps project, from production capacity to brainpower. We are not alone.

## Beyond Corona

The Elektor Helps initiative suits us well and feels organic. One could say it's always been there. Of course, at the moment all eyes are focused on Corona, but we don't want to let this go either. A few years ago, Elektor was the first to put the subject of 'Ethics in Electronics' on the map, and a little earlier we had already used our combined thinking power for the Dutch Asthma Foundation, by developing a $CO_2$ meter with them. This new Elektor Helps initiative fits in well with that. So you can expect from us that, although probably in a different format, we will continue to support projects large and small. From schools that may have an issue to complete circuits for a better world: **Elektor Helps.**

200194-01

• www.elektormagazine.com/helps-contest
• www.elektormagazine.com/helps-content

# Home Information System

## Using Windows on a Raspberry Pi

By **Dr. Veikko Krypczyk** (Germany)

An Information display system installed at home can show live weather forecast, the latest news, departure times of local light rail services, and other current data. In the workplace such a display can show operational metrics and upcoming events at a glance. A clever selection of hardware and software makes this system flexible.

Information systems are in vogue and can be used for a wide variety of purposes in private and business environments. A private home information system or *HomeInfoSystem* (HIS) can, for example, be setup to provide the user with the following information:

- The current local weather conditions and, if required, a forecast for the days ahead.
- National, global or local news stories, possibly restricted to specific topics such as sports or culture.
- Photos, presented in the form of a slideshow, from the cloud.
- The evenings TV schedules or local cinema screenings.
- Service times of local buses or trams with live delay information.
- The current location of family members provided a geolocation function is active (and consented to) in their mobile device.
- A calendar showing upcoming appointments.

In a corporate or business setting there are a number of function-dependent applications that can run with an information system

(*EnterpriseInfoSystem*) to display business metrics and production information. Here are just a few examples:

- Display of work schedules and deadlines, automatically fed from the team calendar in the cloud.
- Target and actual values of the ongoing production processes.
- Current in-house announcements and information.
- Information for customers, participants in corporate events and so on.

The list can be expanded as required with many more interesting applications. The examples given so far only describe passive (displayed) representation of information; add a touchscreen and we can also integrate interactive functions into the system.

Such information systems can be bought off-the-shelf or manufactured in-house. As ever the correct interplay of hardware and software is important. When building such a system yourself, you retain maximum flexibility to tailor the system to your own particular needs and are not dependent on a third-party software provider. This article shows how you can build an easy to implement HomeInfoSystem (HIS) that can be modified according to your own requirements. The software is provided for use and is free for further development [1]. The principle is explained using existing web services such as weather information, news and photo storage to give an insight how you can expand the HomeInfoSystem according to your own needs.

## The system structure

The system should be designed for continuous 24/7 operation so its energy consumption — regardless of the selected display (screen size) — should be kept as low as possible. For this reason, the use of a mini PC in the form of a single-board computer such as a Raspberry Pi, Rock Pi 4, Asus Tinkerboard S or a Banana Pi M3 is recommended. On price alone the Raspberry Pi looks like a good choice; it has more than enough power, good support for the intended application and above all this mini computer is well proven. **Figure 1** shows the structure of the system with a Raspberry Pi at the centre. The model 3B is best suited for this application although the older model 2B can also be used to run Windows 10 IoT as its operating system. The latest RPi model 4 offers significantly more performance, but is not yet officially supported by Windows 10 IoT.
The block diagram in **Figure 1** shows the Raspberry Pi powered by a stabilised plug-in 5 V power supply adapter rated at 2 A (minimum). Hookup to a network can be achieved via a LAN cable or wirelessly via Wi-Fi. When it comes to specifying the display monitor, there is a wide selection in terms of size, resolution and the availability of a touchscreen interface. The screen size is of course dependent on the application, the range extends from a small screen suitable for close-up use on a desk to a 'display panel', readable from a greater distance. The Raspberry Pi (models 2 and 3) achieves a maximum resolution of 1,920 x 1,080 pixels (Full HD). A higher resolution offers a sharper display allowing more content to be displayed, but it is of course a question of cost which resolution you choose, especially if you plan to use a touchscreen. The need to use a touchscreen is governed by various criteria:

- Can/should the user/viewer be able to select something?
- Is an interactive control feature planned?
- Is the screen physically within the user's reach?

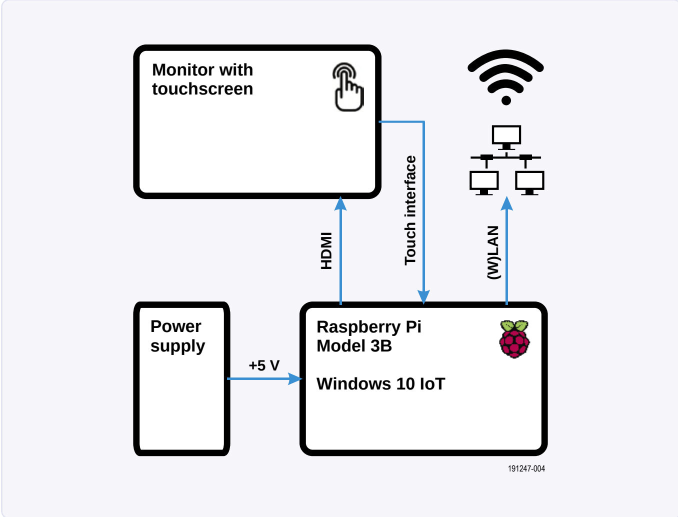



*Figure 1: The HomeInfoSystems (HIS) block diagram.*

The connection between the Raspberry Pi and the monitor is via an HDMI cable. If a touchscreen is used, an additional cable is required between the monitor and the Raspberry Pi. The Raspberry Pi uses a USB port for this function but the touchscreen cable connector differs for each make of monitor. Newer models mostly use a USB-C connection.

Of course, the monitor must also be supplied with power. For my HomeInfoSystem I use a 13" mobile full-HD touch screen with an HDMI connector, USB-C for touch, USB-C for the power supply and an integrated battery. One mounting option is to piggyback the RPi onto the rear of the monitor and then attach the whole assembly to a wall or ceiling. This arrangement allows you to use a really short HDMI cable.

## The software architecture

Windows 10 IoT is installed as the operating system on the Raspberry Pi. From the information-system point of view this OS offers:

- Active and direct 'kiosk' mode, which means that a user can run an app directly, but to which they have only limited access rights.
- Comprehensive options for designing the user interface using the XAML description language.
- Easy administration of the Raspberry Pi via a graphical web interface.
- Development, testing and deployment directly from the Visual Studio integrated development environment.
- Automated update of the app possible via Store (a scenario for commercial applications).

First we need to install Windows 10 IoT on the Raspberry Pi (see **Install Windows 10 IoT on the Raspberry Pi**). A PC (desktop/notebook) with a current version of Windows 10 is required for the development, on which the app can be created and also tested. Visual Studio is used as the development environment.
Only cross-device apps for the runtime environment Universal Windows Platform (UWP) can be run on Windows 10 IoT, standard Windows applications will not run. Apps for the UWP are based on the .NET Core, and created in C # or in Visual Basic .NET with the help of the

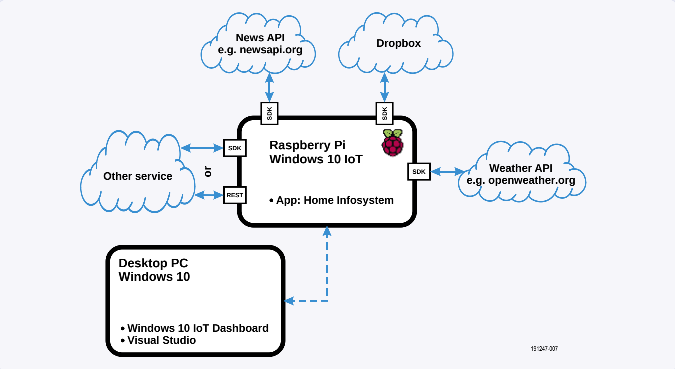



*Figure 2: The HomeInfoSystem software and information flow.*

development environment Visual Studio. The free Community Edition [2] is sufficient for our purposes. You can select the corresponding Workload for the development of apps for the UWP from the Visual Studio Installer. Now you can start developing the first app. Note that you will need to change the type of application from X86 to ARM, because the Raspberry Pi runs an ARM processor.

The apps are fully tested on the development PC, once the version runs without errors, you can create the so-called app package and install it via the web interface on the Raspberry Pi. During development, there is also the possibility to test and use the Debug app directly on the Raspberry Pi. The only requirement for this remote debugging is that both computers are in the same local network. Since apps for the UWP are based on the .NET Core, the entire performance of the associated APIs is available. Ready-made classes can be used for many standard situations. As programming languages you can choose between C # and Visual Basic .NET.

An app runs on the Raspberry Pi in the foreground and in full screen mode (kiosk mode). The user interface is designed declaratively in the description language XAML. To do this, predefined control elements are inserted into layout containers. This creates a relative layout that automatically adapts to different screen sizes and resolutions. The range of available controls includes the usual screen elements such as buttons or text fields. Start a new project, design a minimal user interface, test run the app on your PC and then transfer the app package to the Raspberry Pi. This completes the development cycle.

The core element of the software system is the *HomeInfoSystem* app (**Figure 2**), which is installed on the Raspberry Pi using the App-Package over the web interface. This primary function of the app is to display the content supplied by external web services. Exceptions to this are Static functions and content such as the clock display.

A menu down the left edge of the app allows you to select the content currently displayed. The displayed content is continually switched after a selectable time interval. In the active presentation mode this menu is completely hidden. Only the 'hamburger' menu icon ( ☰ ) at the top of the display serves as a reminder to let you know you can open it. Automatic change between the contents can be configured individually for each service. The presentation of photos from Dropbox does take around 45 s while 20 s is average to update current weather information. Content and services can be integrated into the app in three ways.

- ***Internal:*** The functionality receives no further input data, such as a clock showing current time. The entire function, including the data, is completely statically integrated into the app source code.

- ***External (SDK):*** For many information sources, external service providers can be used. These include, for example, current weather data, news, access to cloud storage services such as Dropbox and calendar integration. So-called software development kits (SDK) are available for many of these services. An SDK is specially tailored to the programming environment, in this

**Listing 1: An example of weather information from https://openweathermap.org.**

```
{"coord": { "lon": 139,"lat": 35},
   "weather": [
  {
         "id": 800,
         "main": "Clear",
         "description": "clear sky",
         "icon": "01n"
  }
  ],
   "base": "stations",
   "main": {
      "temp": 281.52,
      "feels_like": 278.99,
      "temp_min": 280.15,
      "temp_max": 283.71,
      "pressure": 1016,
      "humidity": 93
   },
   "wind": {
      "speed": 0.47,
      "deg": 107.538
   },
   "clouds": {
      "all": 2
   },
   "dt": 1560350192,
   "sys": {
      "type": 3,
      "id": 2019346,
      "message": 0.0065,
      "country": "JP",
      "sunrise": 1560281377,
      "sunset": 1560333478
   },
   "timezone": 32400,
   "id": 1851632,
   "name": "Shuzenji",
   "cod": 200
}
...
```

# INSTALL WINDOWS 10 IOT ON THE RASPBERRY PI

The operating system is installed on the Raspberry Pi SD card (as is the software development) from a Windows 10 PC. If necessary, update the version of Windows 10 in advance. Windows 10 IoT Core is currently version 1809 (build 17763). Your PC must at least be at this level. The SD card is inserted into a suitable (internal or external) SD slot on the PC.

› **Dashboard:** If everything is ready, download the setup file for the dashboard of Windows 10 IoT from [4], which reloads the entire application.

› **Write to the SD card:** In the dashboard (**Figure A**), select the menu item *Setup a New Device* and select the correct device type (Raspberry Pi 2 or 3) as well as the target drive of the SD card. It is a good idea to write the current WiFi profile, name and password for the Raspberry Pi directly onto the SD card here. After accepting the license terms, the download starts and the dashboard writes the operating system to the SD card.

› **Boot the Raspberry Pi:** The SD card described is inserted into the Raspberry Pi and the intended hardware (monitor, touch, plug-in power supply) as well as a mouse, keyboard and possibly a LAN cable are connected. The Raspberry Pi has no on/off switch and starts immediately power is applied. The initial setup takes a little longer. After selecting the language, the Windows 10 IoT start screen appears after a few moments. No custom app is running yet.

› The **Raspberry Pi is configured** from the PC via the dashboard or the Windows Device Portal (WDP) (**Figure B**). With the Raspberry Pi Model 3, Wi-Fi is available on board. Windows 10 IoT automatically detects the hardware, so that the setup goes smoothly. With model 2 without on board Wi-Fi you will need to plug in and set up an external Wi-Fi dongle, which is not quite as convenient as using a model 3. The higher computing power of the RPi version 3 is also advantageous.

› The Raspberry Pi with Windows 10 IoT can be accessed via a browser. The configuration dashboard can be reached via the IP address of the Raspberry Pi. Here you can instal, start and end an app, view status information of system utilisation, manage network connections, or shutdown and restart the system. Alternatively, you can also use the power shell (command line). This type of configuration enables most setting options, but is not very convenient. However, if additional hardware drivers are to be set up, this is the only way to go. We do everything here via the web interface.

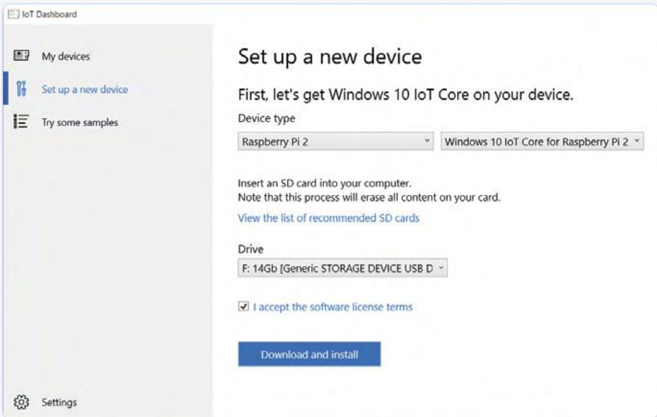



*Figure A: Windows 10 IoT Dashboard.*

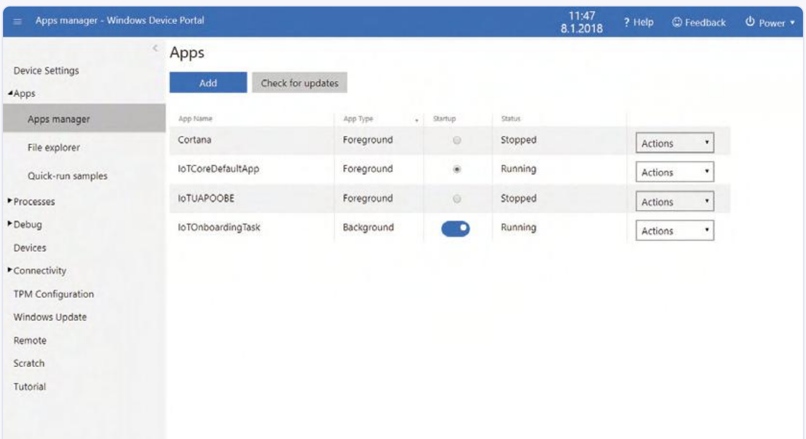



*Figure B: Webinterface to configure the Raspberry Pi.*

**Use and further development of the source code**

The software for the project [1] can be used freely without any problem. It contains: start page (display of some welcome text), clock, a slideshow with Dropbox integration, weather data display and news display. The services require a minimal amount of configuration before use:

▪ ***Dropbox***: Switch to the developer area (console) in your own Dropbox account and create an app. Allow access to the directory of the app and generate the app-secret (key). This key will be entered later in the app's settings.

▪ ***News***: Register with [5] and generate a personal key for access to the news area free of charge. The key cannot be made generally available because the number of accesses is limited. You store the key directly in the *NewsControlViewModel.cs* file (line 95).

▪ ***Weather***: Generate a key for free access and store it in the file *OpenWeatherMapProxyForecast.cs* (line 13).

You insert the keys for the news and weather services directly in Visual Studio then recompile the project and create the app packages for installation on the Raspberry Pi.

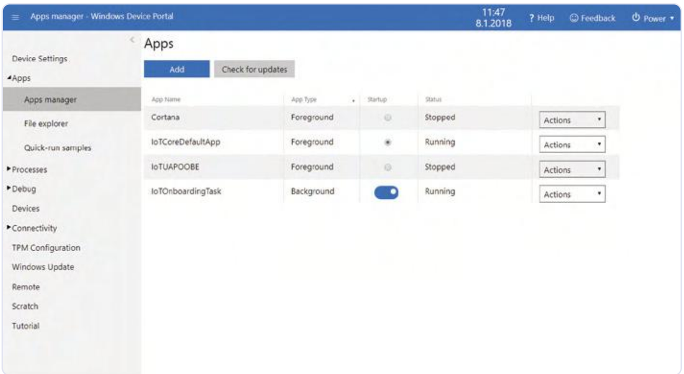



*Figure 3: Options menu for the OpenWeather-API.*

case .NET Core and C #. Compared to a generic programming interface, for example based on REST, it has the advantage that integration and programming is greatly simplified. Such an SDK is usually either provided by the service provider or is available as a library through the community. In Visual Studio, an SDK is integrated via the NuGet package manager.

- *Extern (REST):* If there is no SDK, any service can usually be integrated as a REST interface via the general web interface. The usual web method (GET) is used to send a request to the service and receive a response in a structured data format (XML, JSON). This data can now be parsed, interpreted and graphically prepared and displayed according to your wishes. Cloud services, which can be used by third-party applications, offer such an interface together with the corresponding documentation.

When using external services - regardless of whether via SDK or REST - it is usually a requirement that you register with the service. When you register you will be given a unique key as identification that you must use whenever you request data. Many services are offered free of charge for private and limited commercial use, even if, for example, the number of calls to the data (calls/day) is limited or the data quality provided has to be downgraded. Nevertheless, there are a wide variety of services which can be used very easily and comprehensively, even as the free version.

## Services

The HomeInfoSystem app consists of a combination of several content options (services), which can be added to and combined as required. In this section we describe the connection and implementation of some services. From this you will be able to derive expansion options:

- *Integration of the weather service:* Here we rely on the use of the weather service OpenWeather [3]. This facility offers various services for retrieving weather data. In addition to chargeable offers, you can use the *Current weather* and *5 days/3 hour forecast* services free of charge (**Figure 3**). After registration, an API key is generated with which the weather data can be requested. The convention for such a call is as follows, for example:
  `api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}`
  here you will need to enter the parameters `{city name}` and `{your api key}` according to the location and API key provided. The information returned is a data record in JSON format (**Listing 1**), which can also be easily read 'manually.' It starts with the coordinates of the location, followed by the information about temperature, air pressure, cloud cover and so on. The service website describes the data format in detail. An SDK was developed for the app, which enables easy use of the service. **Listing 2** shows the corresponding short extract from the source code.

**Listing 2: An extract of the SDK using openweathermap.org service.**

```
// Note: The RootObject data type contains the weather data in a tree structure.

public async static Task<RootObject> GetWeather(string location)
  {
    try
    {
      var httpClient = new HttpClient();
      string uri = basisUri + endPoint + "?q=" + location + "&appid=" + apiKey + "&units=metric";
      var response = await httpClient.GetAsync(uri);
      var result = await response.Content.ReadAsStringAsync();
      var serializer = new DataContractJsonSerializer(typeof(RootObject));
      var ms = new MemoryStream(Encoding.UTF8.GetBytes(result));
      var data = (RootObject)serializer.ReadObject(ms);
      return data;
    }
    catch
    {
      return null;
    }
  }
```

*Integration of cloud storage:* To display photos, it is best to make use of one of the standard cloud storage facilities such as Dropbox or OneDrive. These services offer their own API for access via external apps and an SDK for many programming languages. The free version of Dropbox provides 5 GB of space which you can use to store pictures for example. You can access the developer area directly in the online portal. Set up an app at Dropbox (**Figure 4**). You can configure the app via the portal, for example to which folder within the Dropbox structure you want to allow access. From the outside there are several ways of accessing the data (photos) in Dropbox, via user name and password (authentication) or without user interaction using a unique key (Key, App Secret). The latter variant is optimal for us because the app simply identifies itself with such a key in dropbox. We use the SDK for .NET Core, which we can then integrate directly into our app. Reading the photos from Dropbox

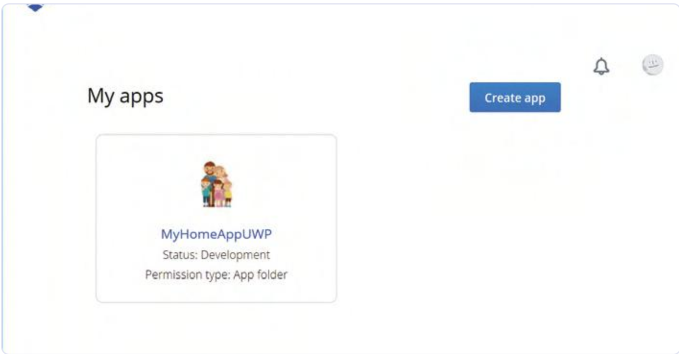



*Figure 4: Set up an app in the Dropbox developer portal.*

**Listing 3: An extract of the SDK to get files from Dropbox.**

```
public static async Task<ObservableCollection<string>> GetFilesFromDropBoxAsync()
  {
    ObservableCollection<string> itemsList = new ObservableCollection<string>();
    // Path for storing the files
    StorageFolder storageFolder = ApplicationData.Current.LocalFolder;
    try
    {
      using (var dbx = new DropboxClient(ProgrammSettings.DropBoxAppToken))
      {
        // List all files in the App folder
        var list = await dbx.Files.ListFolderAsync(string.Empty);
        // loop through all of files
        for (int i = 0; i < list.Entries.Count; i++)
        {
          string localFileName = storageFolder.Path + "/" + list.Entries[i].Name;
          if (File.Exists(localFileName) == false)
          {
            // Download the files
            var file = await dbx.Files.DownloadAsync(list.Entries[i].PathLower);
            // Extract the data stream
            Stream stream = await file.GetContentAsStreamAsync();

            // Write the data stream to a local file
            using (var fileStream = File.Create(localFileName))
            {
              (await file.GetContentAsStreamAsync()).CopyTo(fileStream);
            }
          }
          itemsList.Add(localFileName);
        }
      }
    }
    catch
    {
    }
    return itemsList;
  }
```

is then quite simple: You can see the corresponding code extract in **Listing 3**. For example, users can take photos with their smartphones and save them directly from the mobile device into Dropbox.
- *Static content:* Static content is implemented with the program logic directly in the app. This includes, for example, a clock display with a custom visual representation.

This list shows just some examples of content for the HomeInfoSystem app that will be useful in a domestic installation. For applications in commercial environments (EnterpriseInfoSystem) you will need to find the links to the specific services required.

## Extensions

The services described above are only examples and should encourage you to start your own HIS project. You will no doubt come up with your own ideas of what sort of services to implement. Looking at it from my perspective I can see how the basic system can be sensibly expanded:

- The HomeInfoSystem could display the values from different sensors. The sensor values can be transferred to an IoT backend and then read at specific intervals by the app. Electronic engineers could build a wide range of sensor expansion boards and flexibly integrate them into the system.
- *Show content according to time-of-day* Some content can be assigned a time reference. It might be useful to show live information about local public transport connections between 7 a.m. and 8 a.m. on weekdays but at the weekend you are more likely to be interested in what's showing at your local cinema, which can be determined via a cinema program API.
- *Mobile Services:* You could add a mobile app to the system environment. Users can send data to a server (backend) and the app retrieves this data. Various applications are conceivable. Another exciting application is the use of geoservices. For example, mobile users can (after service requested and consent is given) transmit their current geoposition to a backend. The app can regularly update the position and display it on a map. This way you can check where all your family members are at any moment.
- *Online Services:* Some other services that can be integrated via a public API are: Calendar integration of various services such as Microsoft or Google calendars, railway timetables or transport association services (after specifying the relevant locality).
- *Alternative interface structure:* Instead of a manual or time-triggered change of displayed contents, you could also create an overall view. This should be particularly interesting if you are using a larger screen. For example, in a home-based system you could display the current weather on one side and the news and current photos alternately on the other side.
- As you can see, there are enough ideas to expand the HomeInfoSystem. An important facility would be to input and show information from external (IoT) sensors. An example of this would be to display images from a remote security camera.

## Conclusion and outlook

An information display facility such as this can be useful in a domestic setting as well as in a industrial production environment. At home it can provide current information (weather, news, calendar information) and display important live information while in a commercial workplace environment it can show targets, production information, upcoming team meetings and events etc. The attraction of using a system like this is that you can adapt it to your own particular needs quite easily. After selecting the hardware configuration — in particular the display monitor — the software described here provides a basic set of functions which can then be flexibly expanded and adapted. The source code is open source. You can open the code [1] in Visual Studio 2019 and further develop and modify it according to your own needs.

191247-02



## WEB LINKS


[1] **Project software**: www.elektormagazine.com/191247-02

[2] **VisualStudio**: https://visualstudio.microsoft.com/downloads/

[3] **OpenWeather**: https://openweathermap.org/

[4] **Windows 10 IoT-Core**: https://docs.microsoft.com/en-us/windows/iot-core/downloads

[5] **News-API**: https://newsapi.org/

# Starting with Node-RED

## An open-source visual block-based programming tool

By **Dogan Ibrahim** (United Kingdom)

Node-RED has become very popular recently as it simplifies the task of IoT based project development considerably. In this article we look at what it is and how it can be used in a very simple project such as controlling an LED.

In recent years there has been an increase in the development and use of block-based visual programming tools. The idea here is that the programmer is given a set of visual blocks and all that is required is to connect these building blocks in a logical way to create the required application program. Node-RED is an open source visual block-based programming tool that includes nodes for doing very complex tasks, including web access, Twitter, E-mail, HTTP, Bluetooth, MQTT, controlling the GPIO ports of popular computers such as the Raspberry Pi, Arduino, ESP32, etc. The nice thing about Node-RED is that the programmer does not need to learn to write complex programs. For example, an email can be sent by joining a few nodes together and writing only a few lines of simple code.

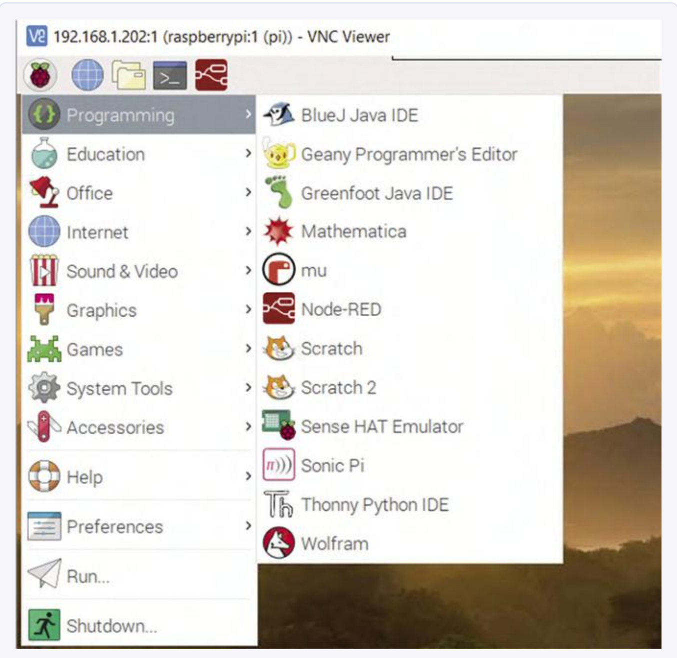



*Figure 1: Starting Node-RED on Raspberry Pi.*

### Installing the Node-RED

Node-RED is already installed on Raspberry Pi and can be started either from the Desktop (**Figure 1**) or by entering the following command from the command line (**Figure 2**):

```
pi@raspberrypi:~ $ node-red-start
```

Node-RED is started as a service and a list of valid commands to start and stop it are displayed on the screen when started, as shown in Figure 2. Notice here that 192.1681.202 is the IP address of author's Raspberry Pi. You can easily find the IP address of your Raspberry Pi by entering the command *ifconfig* from the command line and looking at the line starting with *wlan0:*.

At this point you should start your browser and enter the following link on your PC to start the Node-RED GUI on your screen:

https://<your IP address>:1880

For example, for author's Raspberry Pi, this will be:

https://192.1681.202:1880

**Figure 3** shows the Node-RED startup screen. The screen is basically made of up 3 sections: at the left hand side is a list of nodes available by default when the program is installed. Users however can add additional

```
pi@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.1.202:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use   node-red-stop                          to stop Node-RED
Use   node-red-start                         to start Node-RED again
Use   node-red-log                           to view the recent log output
Use   sudo systemctl enable nodered.service  to autostart Node-RED at every boot
Use   sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
```

*Figure 2: Starting Node-RED from the command line.*

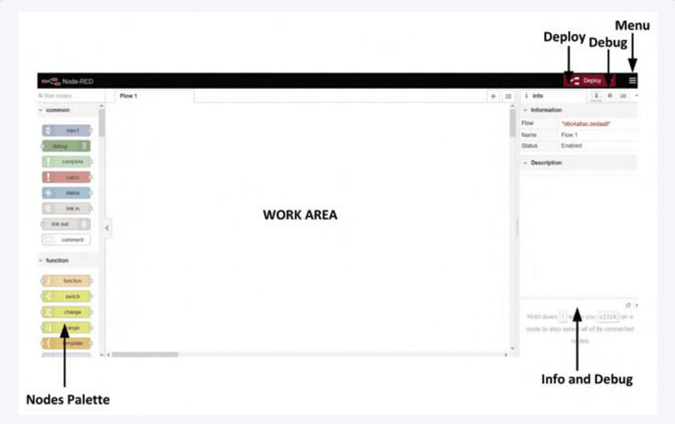



*Figure 3: Node-RED Startup menu.*

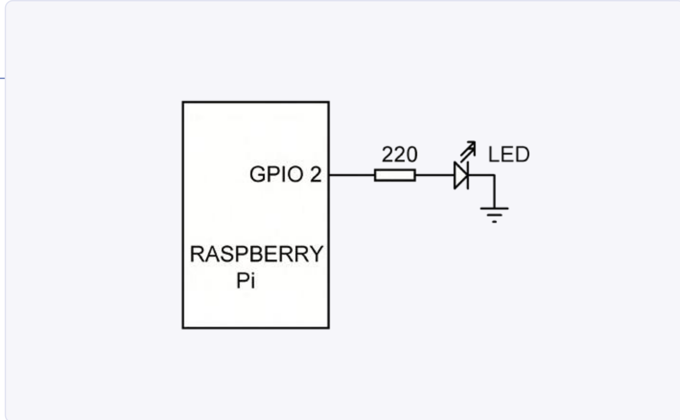



*Figure 4: Circuit diagram of the example.*

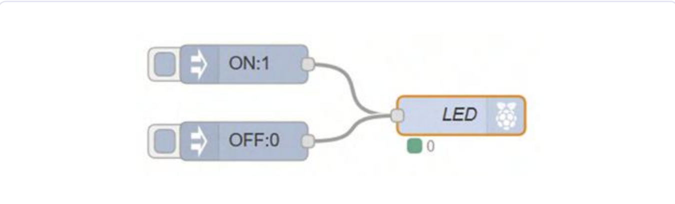



*Figure 5: Flow diagram of the example.*

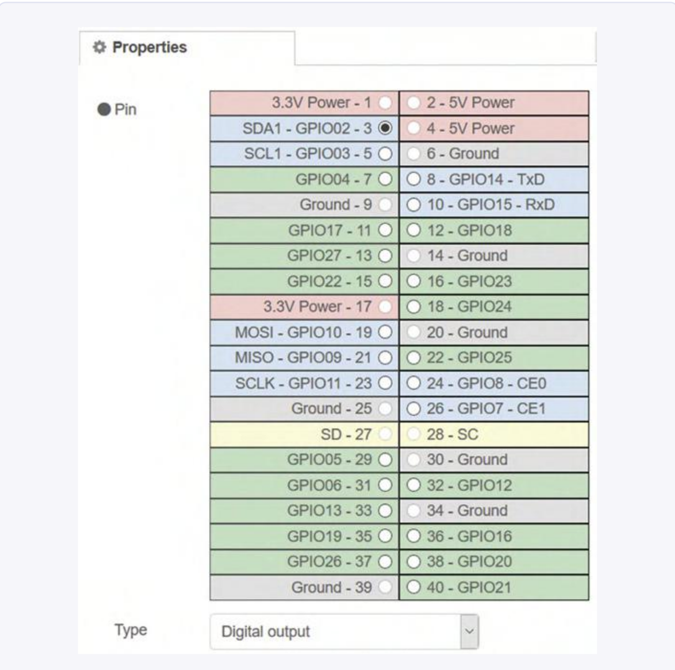



*Figure 6: Configuring node RPi's gpio out.*

nodes from various sources on the Internet. The middle part is the work area where the required nodes are dragged and dropped and connected to form a flow program. Right-hand side is the information and debug area which can be very useful during program development.

Perhaps the easiest way to understand how to use the Node-RED is to look at a simple example. In this example, an LED is connected to GPIO 2 of the Raspberry Pi through a 220-ohm current limiting resistor as shown in **Figure 4**. In this simple project we will be controlling the LED from a Node-RED flow program. The steps are as follows (see flow program in **Figure 5**):

- Click, drag and drop an ***inject*** node to the work space. Inject nodes are used to inject messages (string, number etc) into a flow. In this example we will be injecting 1 or 0 to turn the LED ON or OFF respectively.
- Double-click on the ***inject*** node to configure it. Set its *Payload* to 'number' and enter 1. Clicking the square box at the left hand edge of this node will output 1 to turn ON the LED.
- Set the *Topic* of the injection mode to ON so that we can see the function of this node. Click *Done* to close the configuration screen.
- Click, drag and drop another ***inject*** node as shown in Figure 5 and set its *Payload* to 0 and its *Topic* to OFF. Clicking this node will output 0 to turn OFF the LED.
- Click, drag and drop an ***rpi gpio out*** node to the workspace. This node is used to send digital or PWM data to a GPIO port of the Raspberry Pi. Connect the node as shown in Figure 5.
- Double-click on ***rpi gpio out*** node to configure it as shown in **Figure 6**. Set the *Pin* to GPIO2, *Type* to Digital output, click ***Initialise pin state*** and set the initial pin state to 0. Set the name of the node to LED. Click *Done* to close the configuration screen.

We have now completed our flow program. Click *Deploy* to compile the program and generate the executable code. Make sure that there are no error messages at this stage of the design. Click the square button at the left hand side of the ON inject node and you should see

## WEB LINKS


[1] Node-RED official site: https://nodered.org

[2] Getting started: https://projects.raspberrypi.org/en/projects/getting-started-with-node-red

[3] First flow: https://nodered.org/docs/tutorials/first-flow

[4] Writing functions: https://nodered.org/docs/user-guide/writing-functions

[5] Programming guide: http://noderedguide.com/examples/

the LED turning ON. Clicking the button on the other inject node will turn OFF the LED.

## Further aspects

In this introduction and very simple example we have seen what Node-RED is and how it can be used. One of the powerful points of Node-RED is that it is supported by large communities and there are many nodes developed by third parties that is available free of charge and they can be used in complex projects. For example, the *openweathermap* node gives the local weather conditions such as the temperature, humidity, air pressure, wind speed, precipitation data etc anywhere on Earth. A weather forecasting project can be built in a matter of less than an hour using this node and a few additional nodes. Doing such a project using external sensors is usually costly and may also take considerable amount time to develop. Node-RED supports UDP and TCP based communication with a single node over a Wi-Fi link. This feature enables users to send and receive wireless data from other devices connected to a Wi-Fi, such as mobile phones, tablets, PCs etc.

Some other nodes that the readers may find of interest that can be installed from the Internet free of charge are:

- Dashboard
- Bluetooth
- Amazon Alexa
- I²C and parallel LCD
- Worldmap
- A/D converter
- Ultrasonic sensor

Node-RED is not only for the Raspberry Pi. It can also be used (although limited) with the Arduino and ESP32 processors, or with a combination of the Raspberry Pi and the Arduino, or Raspberry Pi and the ESP32. One of the very important Node-RED nodes is the function node. This node can be configured to have multiple outputs and users can write program codes during the configuration of this node. For example, after getting the local weather data using the openweathermap node, we may want to extract the local temperature and humidity data. This can easily be done by writing couple of statements inside the function node. Interested readers can get more information by referring to the web links [1]-[5] or to the Node-RED book published by the author.

200164-01

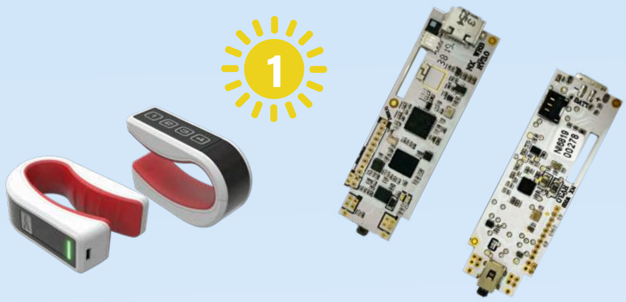


### 4 SALE @ WWW.ELEKTOR.COM

A new book written by the author entitled *Programming With Node-RED: Design IoT Projects with Raspberry Pi, Arduino and ESP32* includes a large number of projects on using the Node-RED together with the currently popular microcontroller development boards.

- **Buy the book from the Elektor Store at**
  www.elektor.com/programming-with-node-red
- **e-book version**
  www.elektor.com/programming-with-node-red-e-book

# the elektor summer giveaway



## 1

**Our friends at NextIND went all out!**

- 2x TACTIGON ONE boards
- 1 kit composed of 3 TACTIGON ONE boards + 1 Raspberry board
- 1 TACTIGON SKIN developer kit (without robot)

## 2

**Offered by Gudeco:**

- 2 x Weller WT 1010 Soldering station

## 3

**iFixit offers not one, but five times:**

- Pro Tech Toolkit

## 4

**Simac / Joy-IT makes the sun shine with:**

- 1 x JOY-iT JT-RD6006 DC Power Supply Set
- 1 x DMSO2D72 3-in-1 Handheld Oscilloscope, Signalgenerator and Multimeter


### Franzis Verlag gave three (German) kits:

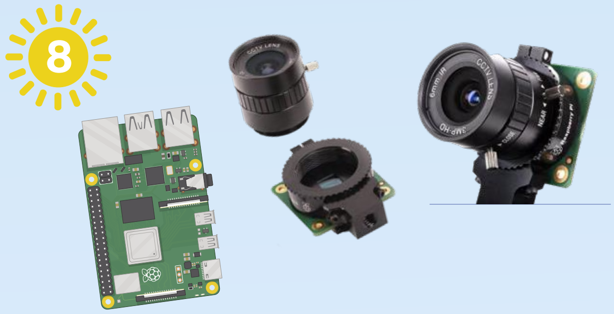
- 1 x MAKER KIT for ESP32
- 1 x MAKER KIT for SMART HOME
- 1 x MAKER KIT for ARDUINO®


### From Peak Electronic Design:

- 1 x ZEN50 - Atlas ZEN Zener Diode Analyser
- 1 x ESR70 Atlas ESR PLUS - Equivalent Series Resistance Meter


### LabNation offers you:

- 1x SmartScope - The oscilloscope of the new generation




### The MagPi - the official Raspberry Pi magazine sponsored:

- 1 x Raspberry Pi 4 - 8GB
- 1 x High Quality Camera Module
- 1 x 6 mm CS-Mount Camera Lens

# How to get it:

Are you looking to get your hands on one of these products? Visit our website and notify us of your preference, that's it! We will select the 'winners' in the last week of August. More information on the products and this little give away, visit:

**elektor.com/giveaway**

This Giveaway is proudly sponsored by

LabNation IFIXIT GUDECO ELEKTRONIK FRANZIS

# Simple
# Function Generator

## With reverse-order signal creation

By **Michael A. Shustov** (Russia) and **Andrey M. Shustov** (Germany)

### PROJECT DECODER

**Tags**
function generator, sine, square, triangle, lab, analogue, opamp, 741

**Level**
entry level – intermediate level – expert level

**Time**
2 hours approx.

**Tools**
older iron

**Cost**
€35 approx.

### FEATURES

- sine, triangle, rectangular waveforms
- Reverse-order waveform generation
- Sinewave ranges 50-500 Hz, 500–5,000 Hz
- Triangle & squarewave ranges 100–1000 Hz, 1000–10,000 Hz
- Frequency ranges easily adaptable
- Low-budget project
- Through-hole parts only
- 100% microcontroller-free

In a single instrument, sine, triangular, and rectangular waveforms are usually generated using R-C charge/discharge networks followed by appropriate filters. Alternatively and certainly more fashionably these days, such signals can be synthesized by a microprocessor. The function generator described here follows a conceptually different approach and matching circuitry.

A function generator is an instrument that generates more than one waveform of variable frequency and, optionally, amplitude. This is a useful device necessary in labs for testing, tweaking, faultfinding, repairing and tuning electronic devices.

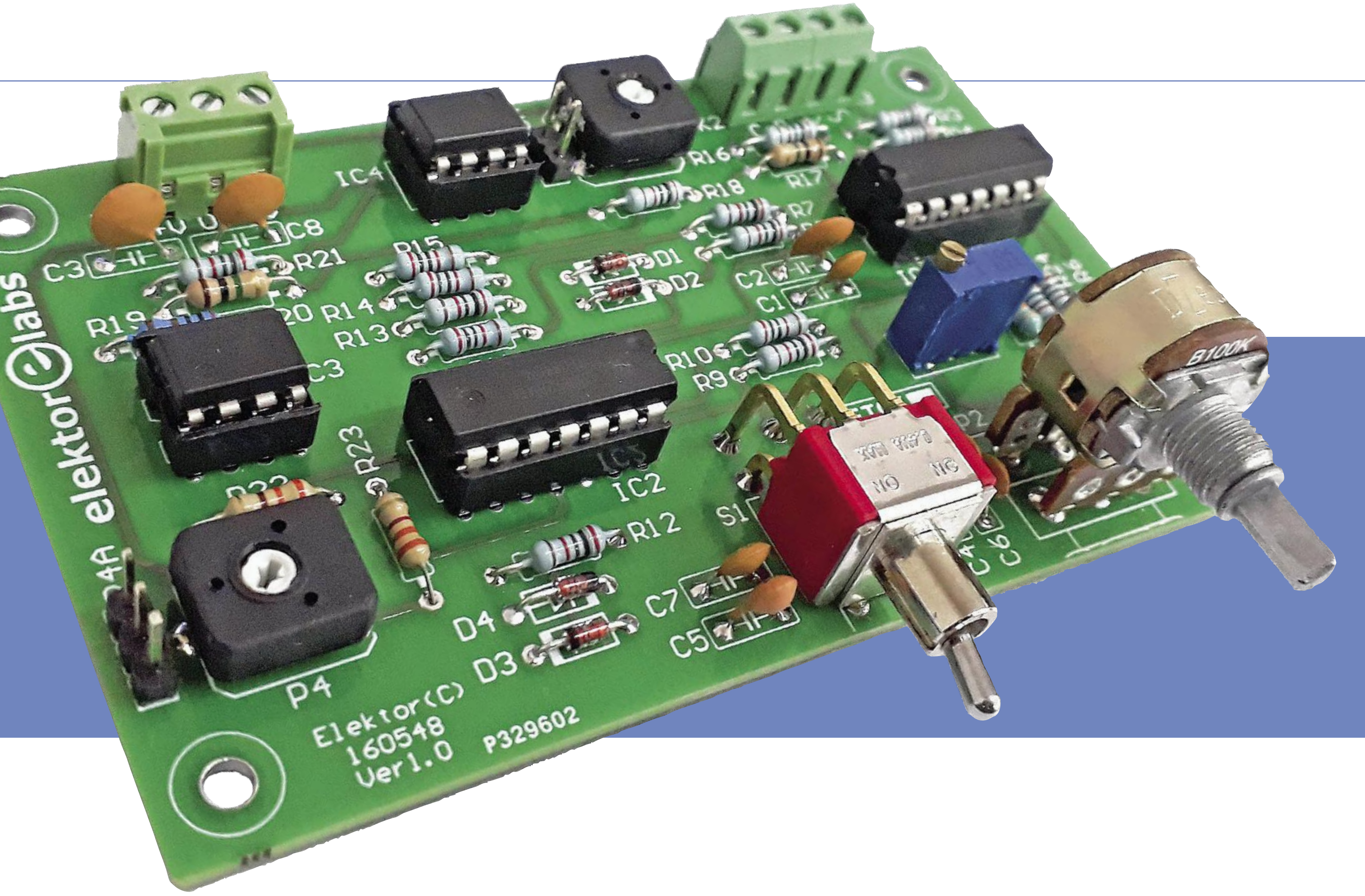
In most function generators of classic design, a rectangular-pulse generator is used as the starting point. Next comes a rectangular-to-triangular voltage converter usually based on the charge/discharge process. Next, the triangular waveform is transformed into a kind of sinewave, usually including good suppression of the first harmonic.

The main disadvantage of this three-step process and the associated circuitry is the inherent nonlinearity of the charge-discharge processes, which is especially noticeable when the generator frequency is tuned. In particular, the distortion of the sinusoidal signal increases accordingly owing to sub optimal filtering of the higher harmonics of a complex signal.

### The other way around

The function generator described below is off the beaten track in that the converting of signals occurs in the reverse order. First, a sinusoidal waveform is created, which is then converted into a triangular waveform. Next, a bipolar signal of rectangular shape is obtained from the triangle.

There are two frequency bands which can be selected by a selector switch. The sinewave




signal is generated by an opamp-based oscillator. The triangular wave is generated by converting the sinewave to a triangular shape by rectifying and inverting the sinewave signal. Finally, the squarewave signal is generated by a differential comparator.

## Circuit description

Proceeding from theory to practice, the schematic of the Simple Function Generator is shown in **Figure 1**. The circuit works on a ±5 V (symmetrical) power supply. Opamps IC1.A, IC1.B, and IC1.C form the sinewave oscillator. It actually outputs two sinewaves with a 90-degree phase difference. The coverage of the two frequency bands depends on the capacitor values C4 and C6 (33 nF) and their respective counterparts C5 and C7 (3.3 nF). Oscillator feedback is provided by resistor R24. Preset P1 is used to set the feedback to a level where a proper sinewave is obtained without clipping the signals. The output frequency of the generator is continuously adjustable on dual-potentiometer P2 with its sections P2A and P2B. S1 switches between the two frequency ranges, assuming equal capacitance for C4 and C6 on the 'low' range, and C5 and C7 on the 'high' range.

The $A_{OUT}$ and $B_{OUT}$ signals from the sinewave oscillator are fed to two practically identical rectifier circuits IC2.A/IC2.B and IC2.D/IC2.C which not only rectify the sinewave but 'flip' the clipped part of the waveform to the positive and negative side. That is, the sinewave is fully rectified above and below

## THE CIRCUIT IN MATHS

The various waveforms that exist in the circuit may be described by the formulae given below. Note: $U_{A\text{-}G}$ (t) corresponds to node marked by circled letter in circuit diagram.

$$U_A(t) = U_0 \sin(\omega t)$$

$$U_B(t) = U_0 \sin(\omega t + 90)$$

$$U_C(t) = U_0 \sin(2\omega t)$$

$$U_D(t) = U_0 \,\mathrm{abs}[\sin(\omega t)]$$

$$U_E(t) = -U_0 \,\mathrm{abs}[\sin(\omega t + 90)]$$

$$U_F(t) = U_0 \{\mathrm{abs}[\sin(\omega t)] - \mathrm{abs}[\sin(\omega t + 90)]\}$$

$$U_G(t) = \begin{cases} -U_0 \text{ if } U_F(t) > 0 \\ U_0 \text{ if } U_F(t) < 0 \end{cases}$$

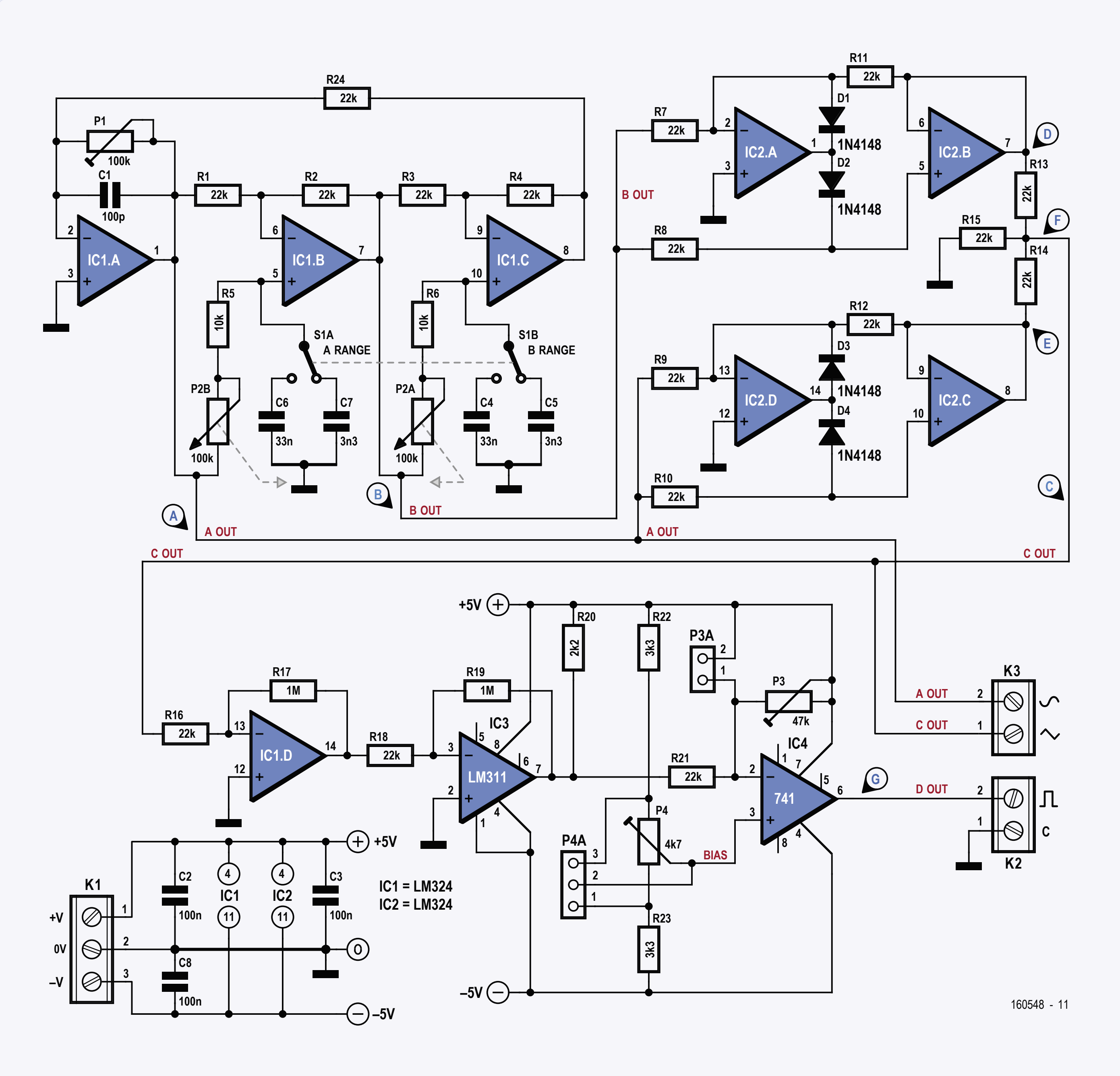



*Figure 1. Circuit diagram of the Simple Function Generator with labels shown for the main internal signals (refer to maths inset). Unusually the signal generation order is: sine → triangle → rectangle.*

'zero.' These two rectified waveforms are 90 degrees out of phase and have double the frequency of the original sinewave signal. The rectified outputs are added to give signal $C_{OUT}$. This (mathematic) addition conveniently forms the triangular wave, albeit of lower amplitude than the originating sinewave.

The triangular wave is next converted to squarewave shape by IC1.D (another LM324 opamp) in combination with IC3 (an LM311). Preset P4 is used to adjust the bias voltage on IC4 (a 741) which in turn adjusts the offset voltage of the squarewave. Finally, preset P3 is used to adjust the amplitude of the squarewave on $D_{OUT}$. All three output signals are available on 2-way PCB screw terminal blocks K3 and K2.

With S1 set to the Low range (i.e. with C4 and C6 in circuit) the generator covers 50–500 Hz for sinewave output, and 100-1,000 Hz for triangular and rectangular wave output due to the doubling of the original frequency. By modifying the frequency-determining capacitors or adding additional ranges on S1, frequencies down to sub-Hz can be provided.

When S1 is operated to switch capacitors C5 and C7 into circuit the frequency will rise by a factor of 10. With C5 = C7 = 3.3 nF as shown, the range of generated frequencies is:

- 1,000–10,000 Hz for the triangular waveform and rectangular waveforms. The effective range is 1,000 to 8,000 Hz; it can be extended upwards slightly by changing C5 and C7 to 2.2 nF.
- 500-5,000 Hz for the sinusoidal waveform; again the effective range is 500 to 3,500 Hz roughly; upwards of that may be achieved by changing C5 and C7 to 2.2 nF.

## Building

Elektor Labs designed a printed circuit board for the Simple Function Generator. The board design is pictured in **Figure 2** along with the parts list. Construction should be plain sailing even for relative beginners as all parts are through-hole, the board is spaciously laid out, and there are no microcontrollers to program. The pinheaders marked P3A and P4A allow external potentiometers to be connected through connecting wires. In that case, omit presets P3 and P4 from the board.

### Testing

Assuming you have successfully built the board, the recommended test procedure for the project is as follows.

- Connect the ±5 V (symmetrical) supply to connector K1.
- Select the required frequency range Low or High on switch S1.
- Connect an oscilloscope to line $A_{OUT}$ on K3 (sine), and C (GND).
- Adjust P1 to get a sinewave signal that's as clean as possible.
- Move the 'scope input to $C_{OUT}$ on K3 and check the presence of the triangle signal.
- Move the 'scope input to $D_{OUT}$ on K3 and check the presence of the rectangular signal.
- Operate P2 (dual pot) to confirm it controls the generator output frequency.
- Adjust P3 to set the amplitude of the signal.
- Adjust P4 to set the rectangular wave offset voltage.

You're all set now to apply a sinusoidal, triangular or rectangular, frequency and amplitude adjustable signal to your equipment and see on the 'scope how it responds!

(160548)

### 4 SALE @ ELEKTOR.COM

- **Simple Function Generator PCB, bare**
  www.elektor.com/function-generator
- **Book: Electronics for All**
  www.elektor.com/electronic-circuits-for-all
- **Book: Analog Circuit Design, Volume 1**
  www.elektor.com/analog-circuit-design-1
- **Book: Analog Circuit Design, Volume 2**
  www.elektor.com/analog-circuit-design-2
- **Book: Analog Circuit Design, Volume 3**
  www.elektor.com/analog-circuit-design-3

## COMPONENT LIST

**Resistors**
R1-R4,R7-R16,R18,R21,R24 = 22kΩ
R5,R6 = 10kΩ
R17,R19 = 1MΩ
R20 = 2.2kΩ
R22,R23 = 3.3kΩ
P1 = 100kΩ trimpot
P2 = 100kΩ ganged (stereo) potentiometer
P3 = 47kΩ trimpot
P4 = 4.7kΩ trimpot

**Capacitors**
C1 = 100pF, 50V, C0G, 5%
C2,C3,C8 = 100nF, ceramic, 50 V, MCFY Series
C4,C6 = 33nF, 100V, SkyCap SR Series, ±10%
C5, C7 = 3.3nF, 50V, C0G, 5%

**Semiconductors**
D1-D4 = 1N4148
IC1,IC2 = LM324
IC3 = LM311
IC4 = 741 (uA741)

**Miscellaneous**
K1 = 3-way PCB screw terminal block, 3.5mm pitch
K2,K3 = 2-way PCB screw terminal block, 3.5mm pitch
P3A = 2-pin pinheader, 0.1" pitch, vertical
P4A = 3-pin pinheader, 0.1" pitch, vertical
S1 = DPDT toggle switch, non-illuminated
8-way DIP IC Socket
14-way DIP IC Socket

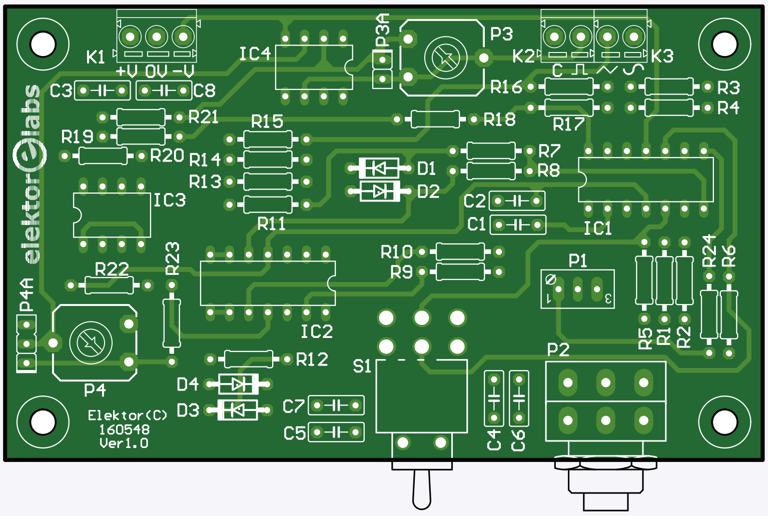
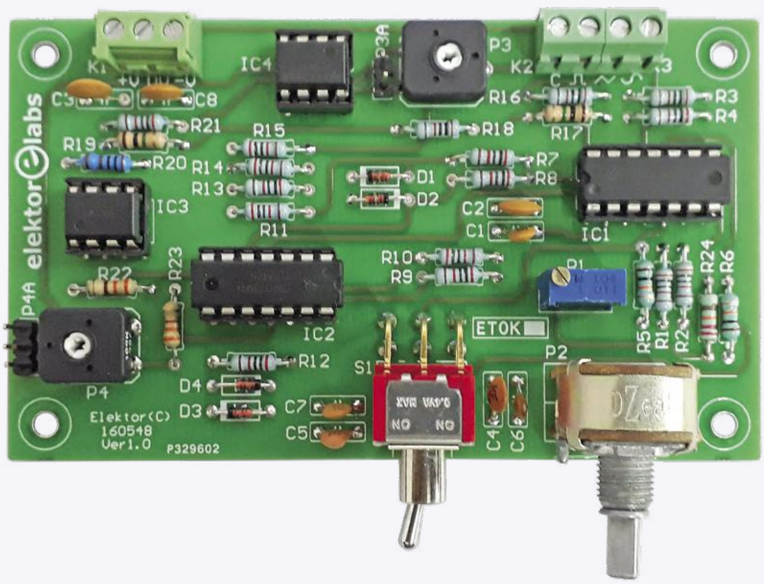
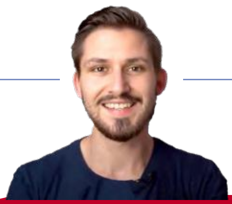



*Figure 2. Printed circuit board designed for the Simple Function Generator. Note no SMDs in sight.*





# GreatScott! Builds a LoRa Alarm System

Link to YouTube

By **GreatScott!**, with an intro & outro by **Luc lemmens** (Elektor Labs)

In this article we present a project designed by Youtuber GreatScott! to transmit a notification signal from the intruder alarm system installed in his garage box to his apartment using LoRa, covering a distance of 600 metres. He used two STM32 Nucleo boards and two RFM95 LoRa modules (with Elektor breakout boards) sponsored by Elektor to build and test the hardware.

According to Wikipedia, "Great Scott!" is "*... an interjection of surprise, amazement, or dismay. It is a distinctive but inoffensive exclamation, popular in the second half of the 19th century and the early 20th century, and now considered dated*". Some of our readers may remember that this exclamation was often used by Emmett 'Doc' Brown in the well-known movie trilogy "Back to the Future", but nowadays maybe even more of you know the *GreatScott!* Youtube channel, which has featured loads of electronics projects and tutorials since its launch back in 2013.

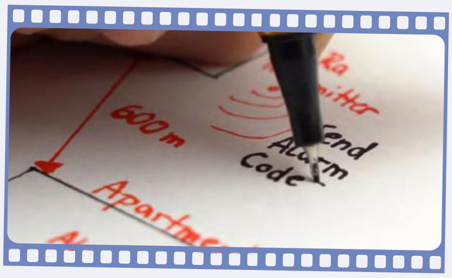
## GreatScott! has the story

*The two Arduino sketches needed to complete the alarm extension are surprisingly simple, thanks to the RFM95 Arduino libraries you can pull from GitHub or the web page for this article [1]. In the next stretch of this article, GreatScott! tells the story about the way the project developed.*

Even though my garage has an alarm system installed consisting of a light barrier, stroboscope light, siren and a Controllino PLC, somebody decided to break in about four months ago. Since then I improved the garage lock but the main problem is that my apartment is about 600 m (1,800 ft) away from my garage and I want to get notified there when somebody triggers the alarm system.

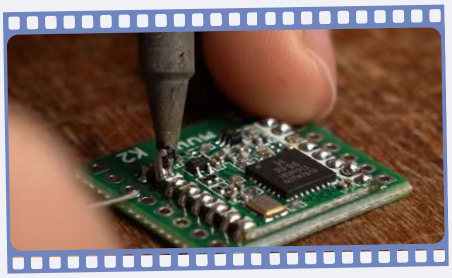
As a solution to my problem Elektor sent me a handful of components consisting of two RFM95 LoRa boards with fitting breakout PCBs, two STM32 based microcontroller development boards, and one LG02 Dragon LoRa compatible Gateway.

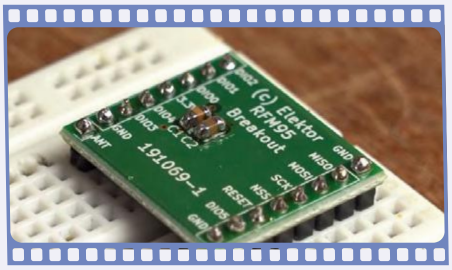
In this project I will not only play around with the LoRa technology but I will also create a small receiver and transmitter system based on LoRa which will trigger a siren in my apartment when the alarm in my garage gets triggered.



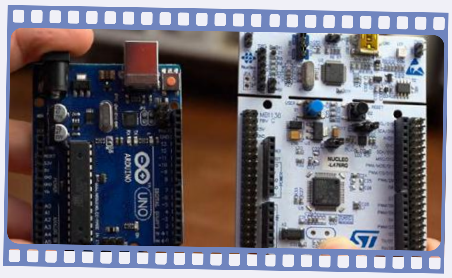

LoRa stands for Long Range and its inventor, the company Semtech, describes this wireless radio frequency technology as a long-range, low-power wireless platform that has become the *de facto* technology for Internet Of Things (IOT) networks worldwide. That already sounds quite promising for my project since I want to transmit very little data over a relatively long distance.

For the hardware, I started with soldering the RFM95 boards to their breakout boards together with some male headers and decoupling capacitors.

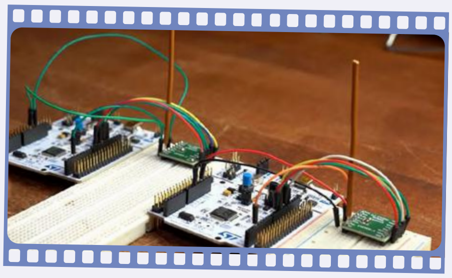





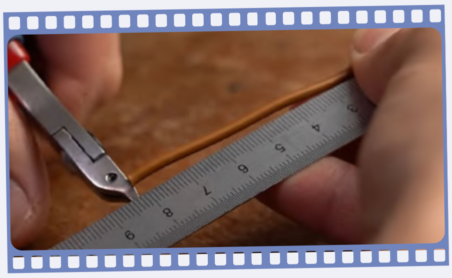
According to the datasheet the RFM95 modules are Low Power LoRa Transceiver modules which basically means they can transmit and receive LoRa modulated data and thus they are the key components for my project.
To interact with them they got an SPI interface which obviously means we need a microcontroller to talk to them and some code to write and read their registers.
For the microcontroller part I unpacked the two STM32 Nucleo L476RG development boards that look like Arduino Uno boards.

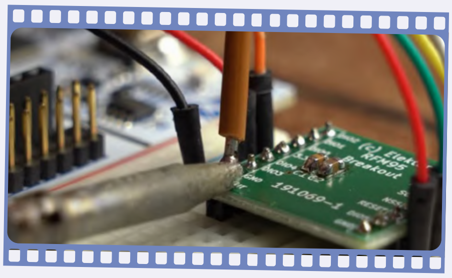


And yes you could use an Arduino to interact with the RFM95 board, but the advantage of the STM32 board is that it works with 3.3-V logic levels instead of 5-V logic levels like the Arduino and the RFM95 is only rated for 3.3 V.
Time to connect the two development boards to the two LoRa modules according to the wiring scheme mentioned in the Arduino LoRa library which I will be using to make coding a lot simpler.

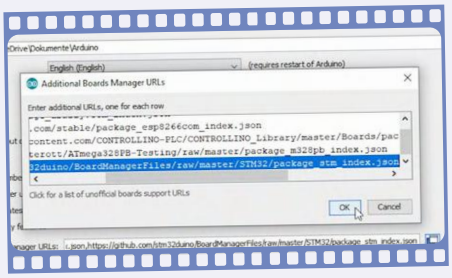


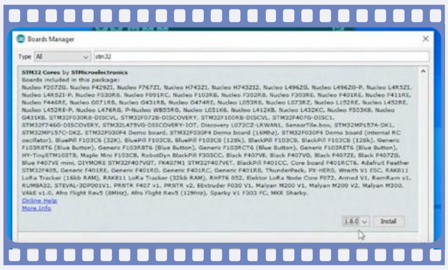
Of course, we need antennas for the LoRa boards which we can easily make. According to these simple calculations I found in an Elektor magazine article we just need a piece of wire with a length of around 82 mm. So I cut two of those wire pieces and soldered them directly to the antenna pin of the LoRa boards.

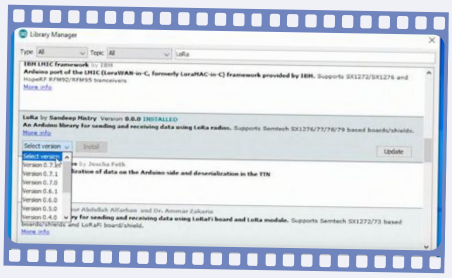




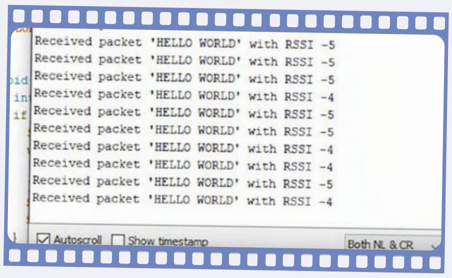
And just like that our test hardware setup was complete and it was time to connect both development boards to my computer. After including the URL for the STM32 boards library and installing it under the Boards Manager, I also installed the Arduino LoRa library.



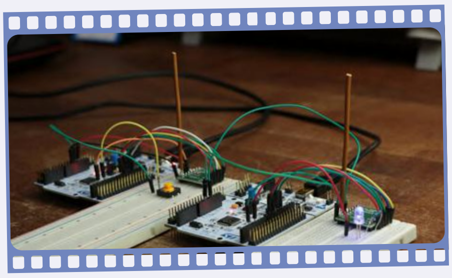




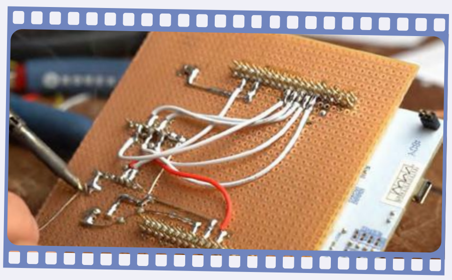
Then I wrote a bit of example code for a transmitter which sends out a simple HELLO WORLD and some example code for a receiver which outputs its received message over the Serial Monitor along with its received signal strength indicator.
After selecting the correct STM32 Nucleo boards and uploading the codes, we can see that the receiver is successfully getting the messages sent from the transmitter, awesome!

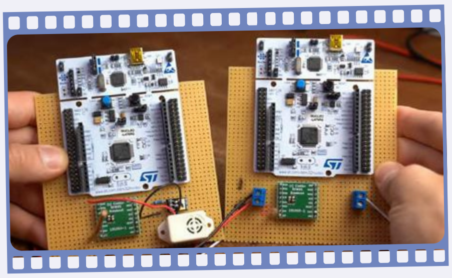


But I was not done yet, because the transmitter will have to send out a proper alarm code when it gets activated by the alarm system in my garage and the receiver will have to turn on a small MOSFET and thus a siren when it receives the correct alarm code.
For testing purposes, I first just added a push button to the transmitter and an LED with current limiting resistor to the receiver side to signal reception of the alarm notification.



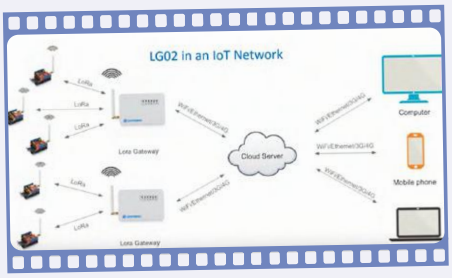
For the transmitter sketch, all I had to do was to define an alarm message and use the pushbutton input to trigger transmission. The sketch for the receiver was a just bit more complicated though, because I needed to store the received message in an array which then gets compared to the predefined alarm message and only if they match, the LED gets activated.
After uploading both sketches to the boards and doing some tests, it looked like everything worked just fine and it was time to draw proper schematics for both the transmitter (**Figure 1**) and receiver (**Figure 2**). After that, I soldered and connected all components on two pieces of Veroboard in order to create sturdier receiver and transmitter hardware.



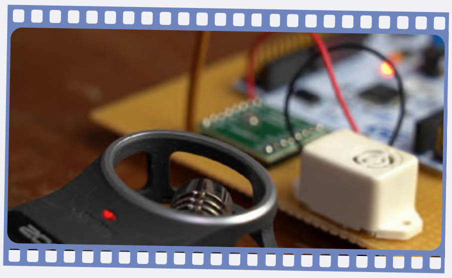


After an hour or so of soldering the Veroboard systems were finished, but before I headed to my garage with the transmitter board I realized that I completely forgot the LoRa Gateway that was included in the package from Elektor and I started wondering whether it could improve my project.


To cut a long story short: for this simple application it is not worth the effort to struggle through the manuals and menus of the gateway. It can basically receive data from LoRa nodes which it can upload to a cloud server through your home network. These uploaded data can then be accessed by other devices or the gateway can send the data to other LoRa nodes.




I would say this concept is useful if you need to log, say, lots of temperature or humidity data for a greenhouse for which you can use a LoRa WAN Server, MQTT or a TCP-IP Server. And I know these terms all super complicated if you are - like me - not into this stuff, but the user manual actually gives you very good instructions on how to use these three methods, even including Arduino or STM32 sketches. My intended system however is so simple that it does not require server upload methods. Also, I don't want to be forced into having a stable Internet connection in order to make my system work.
However, the gateway can be used directly to send or receive LoRa messages and it even features an option to use a customized script that can react to specific LoRa messages. That actually got me thinking whether I should implement it as a kind of central point (hub) for my system, but after trying for hours and hours to send out the correct alarm code and rewriting the custom Linux shell script, I gave up on the idea because it would just overcomplicate things. So for now I will not be using the gateway, but maybe I will in another project.
Returning to my simple application, I powered up my receiver and went to my garage to connect the transmitter to my Controllino PLC, for which I also had to change the code just a bit.
After triggering the alarm, the buzzer of the receiver station in my apartment started screaming as well, which means this project was a success!



If you have range problems, you can always try tweaking the power, spreading factor and bandwidth settings of the LoRa system, about which you can easily learn more on the Arduino LoRa GitHub.

```
void setup() {
  pinMode(D4, INPUT_PULLDOWN);
  LoRa.setTxPower(20);
  LoRa.setSpreadingFactor(12);
  LoRa.setSignalBandwidth(62.5E3);
  Serial.begin(9600);
  if (!LoRa.begin(868100000)) {
```

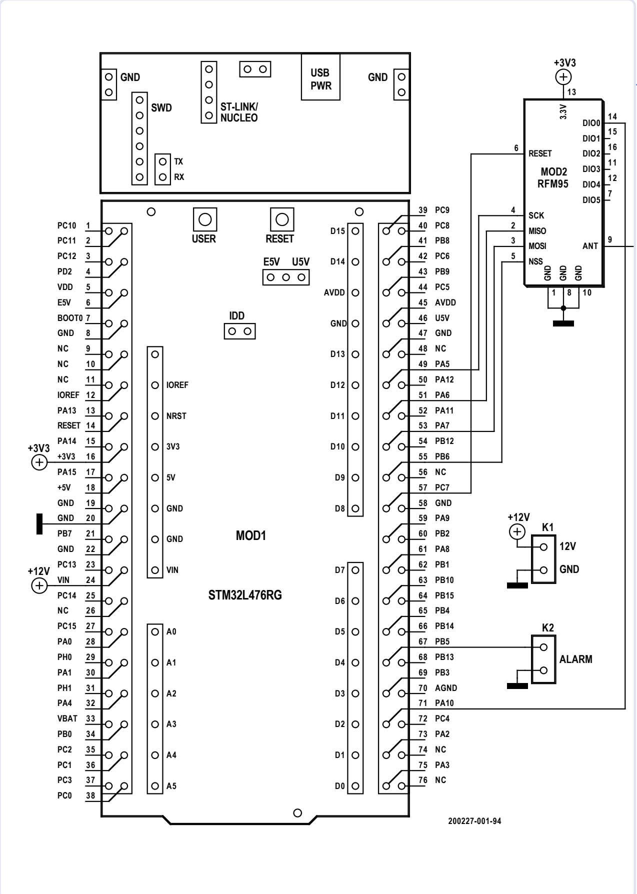



Figure 1: Transmitter schematic.

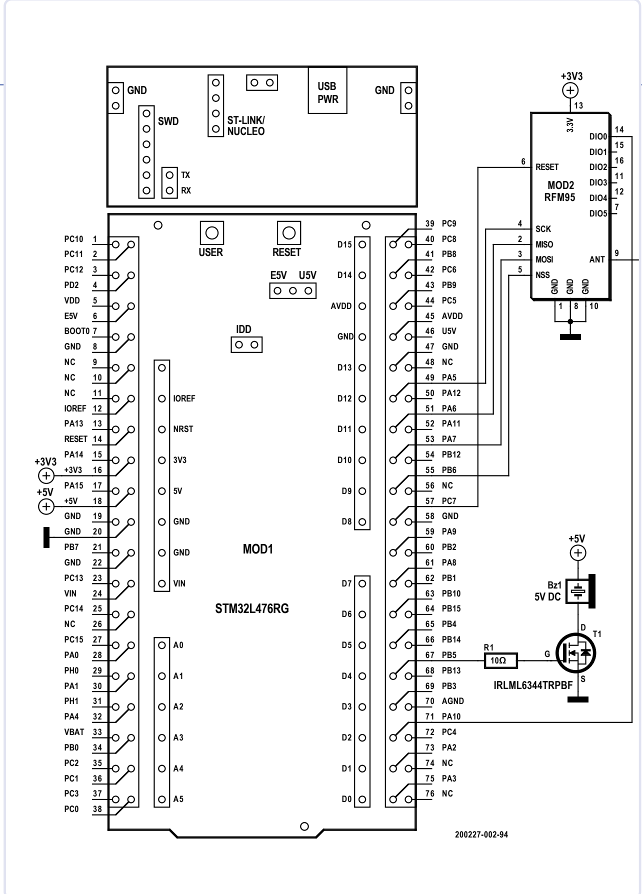



Figure 2: Receiver schematic.

## Conclusion

*With this project GreatScott! shows that LoRa can be an easy solution for simple long-range, low speed, communication. Note that the FET he used on the receiver board is a SOT-23 type (mounted on a breakout board) with exceptionally good specs for 3.3-V logic (max. 5 A and only 800 mV threshold voltage!), but in this case a TO-92 style TN0702 will work too. If you want to build this project, the main components are available in the Elektor Store. As you can see in the* ***4 Sale @ www.elektor.com box****, the STM Nucleo board is part of a starter kit, but it can also be bought separately.*

*The complete development process GreatsScott! went through can also be viewed in the video on Youtube [2].*

200227-01

## COMPONENT LIST

Note: parts required for receiver AND transmitter (i.e. order 2 pcs of each item):

STM32 Nucleo L476RG Board
RFM95 Ultra LoRa Transceiver Module
Elektor RFM95 Break-out Board no. 191069-1
C1 = 100nF, 10%, 10V, MLCC, X7R, SMD 0805
C2 = 10µF, 20%, 10V, MLCC, X5R, SMD 0805
2 pcs 8-way pinheader 0.1" (2.54mm) pitch
82 mm wire (antenna)

Additional parts for transmitter:
K1,K2 = 2-way PCB screw terminal block 0.2" (5.08mm) pitch

Additional parts for receiver:
R1 = 10Ω
T1 = IRLML6344TRPBF (with optional breakout board) or TN0702 (TO-92 case)
Bz1 = 5VDC buzzer

## 4 Sale @ www.elektor.com

- **STM32 Nucleo L476RG Board (SKU 17946)**
  www.elektor.com/stm32-nucleo-l476rg-board
- **Elektor STM32 Nucleo Starter Kit (SKU 19205)**
  www.elektor.com/elektor-stm32-nucleo-starter-kit
- **RFM95 Break-out Board – Bare PCB (191069-1) (SKU 19142)**
  www.elektor.com/rfm95-break-out-board-bare-pcb-191069-1
- **RFM95 Ultra LoRa Transceiver Module (868/915 MHz) (SKU 18715)**
  www.elektor.com/rfm95-ultra-lora-transceiver-module-868-915-mhz
- **Dragino LG02 Dual Channels LoRa IoT Gateway (SKU 18624)**
  www.elektor.com/dragino-lg02-dual-channels-lora-iot-gateway

## WEB LINKS

[1] LoRa / Arduino sketches download: http://www.elektormagazine.com/200227-01
[2] GreatScott! video: Transmitting an Alarm Signal with LoRa (600m)! Improving my Garage Alarm System: https://youtu.be/ItZwa1AdrpU


# Joy-iT RD6006 Benchtop Power Supply Kit

By Dr. Thomas Scherer (Germany)

A heavy-duty benchtop power supply from Joy-iT uses switched-mode digital regulation to provide a DC output supply in a range from 0-60 V at 0-6 A. It's supplied in kit-form but all you need for assembly is a standard Phillips screwdriver. Is the kit any good?

Even if it is true that engineers and hackers can never have enough power supplies, I am now beginning to run out of bench space. I had good feedback after I reviewed the small linear PeakTech 6080 A lab power supply (see review at [1]), and was keen to look at a more powerful benchtop power supply which employed switched-mode technology. I looked at the RD6006 benchtop power supply - this time from Joy-iT. One reason to choose this model is that it is supplied as a kit of parts and not as a finished piece of equipment. As any true electronicist or maker will tell you, putting something together yourself always gives that added level of satisfaction to any project (especially when it works).

## The kit

The Joy-iT JT-RD6006 DC Power Supply Bundle [2] is available to order from the Elektor Store. It is not a conventional kit where you would be expected to solder components onto a PCB. This is more of a bolt-together assembly job; all the supplied modules just need to be wired together inside the enclosure using just a screwdriver and without the need to power up a soldering iron. The individual modules have already been tested and calibrated, but more on that later.

Key features of the Joy-iT JT-RD6006 DC:

- Mains voltage: 115/230 V (standard)
- Output voltage: 0 to 60 V
- Output current: 0 to 6 A


*Figure 1: The benchtop power supply bundle comes in an impressive black case. An edition of Elektor magazine is shown for comparison.*

*Figure 2: Three parcels are lurking inside.*

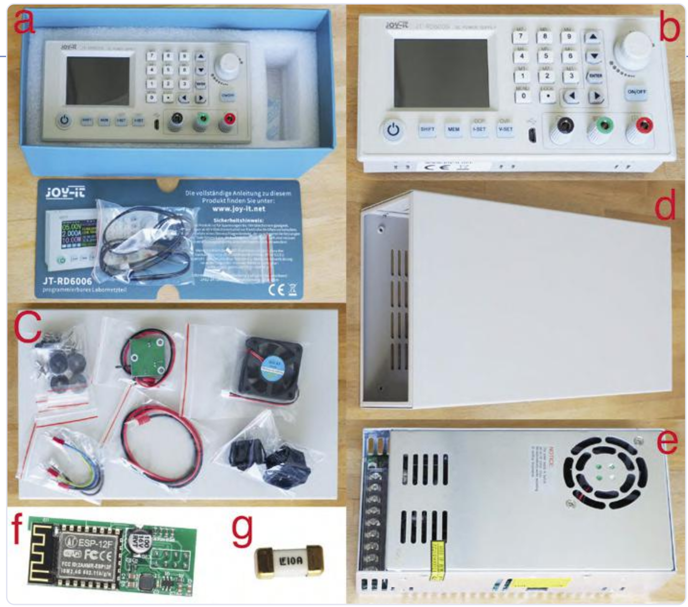




*Figure 3: Unboxed: 3a + 3b: the electronics module including front panel as well as temperature sensor and SMD fuse; 3c: small mechanical parts, extra fan including control board, cable, power socket and switch; 3d: the sheet steel housing, 3e: the 400 W industry-standard open frame power supply; 3f: the ESP12F Wi-Fi module; 3g: the included 10 A SMD replacement fuse in all its glory.*

- Maximum output power: 360 W
- Output voltage resolution: 10 mV
- Output current resolution: 1 mA
- Charging function: 0 - 9,999.99 Ah; 0 - 9,999.99 Wh
- Residual output ripple: 100 $mV_{PP}$ (at max. load)
- Display: 2.4" colour LCD
- Operation: keypad, rotary encoder, USB, Wi-Fi
- Wi-Fi module: ESP12F

From its spec this benchtop power supply offers quite impressive capabilities. It is more capable than an older digital 480 W benchtop power supply of Chinese provenance I have. The JT-RD6006 is even a little cheaper at €175.46 (for Elektor members). An output capability of 60 V and 6 A will probably be sufficient for around 95% of all typical applications in the lab.

## Unboxing

The power supply bundle comes in an impressively large box measuring 50 x 36 x 15 cm (**Figure 1**). Inside the three 'packages' (**Figure 2)** are many smaller parts (**Figure 3**). I really was beginning to doubt whether I would be able to get away without using any solder.

As you can see in **Figure 3a**, there are no instructions included and unfortunately no direct URL pointing me to some; only a link to the manufacturer's website. To resolve this, Google *RD60006* and you will find the operating instructions as a .PDF file with a click on a picture [3].

Once you have a copy of the file you will see it's just an operating manual with no assembly instructions. The assembly and wiring of the many parts is fairly logical if you have an engineering bent, but not entirely trivial.

## Assembly instructions

In the absence of supplied assembly instructions these guidelines should help you through. All you need for this is a medium-sized Phillips screwdriver, a very small flat-bladed screwdriver and some small needle-nose pliers for gripping the nuts. I can confirm that you won't need a soldering iron; I was able to put it all together reasonably quickly on a kitchen table.

First remove the eight M3 x 5 countersunk Phillips screws with which the cover of the gray sheet metal housing is attached. The mains switch and IEC socket can now be inserted into the cutouts on the rear panel (**Figure 4**). The switch snaps into place, but the power socket requires two M3 screws and nuts. Here the needle nose pliers are used to hold the nuts while tightening the screws. Finally, the small circuit board with the fan controller using be attached using three black M3 x 5 screws.

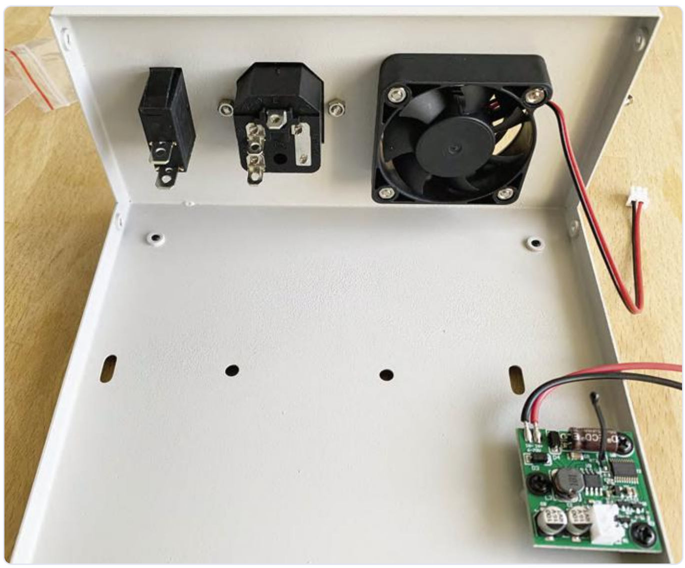


*Figure 4: Mounting the power switch, IEC socket, fan and the circuit board with the fan control.*

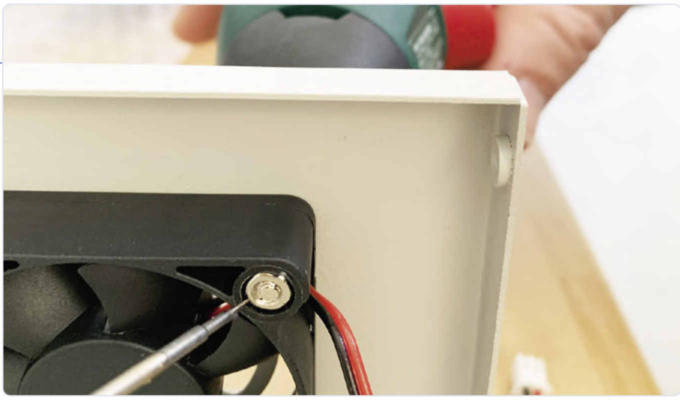


Figure 5: A fine-tipped screwdriver stops the nuts from spinning while tightening.

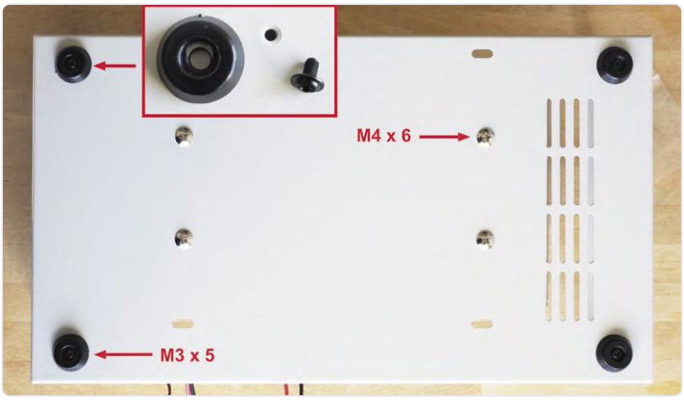



Figure 6: The back of the metal case. The power supply is fastened with four M4x6 screws and the four rubber feet with M3x5 screws. The power supply connections are on the left here.

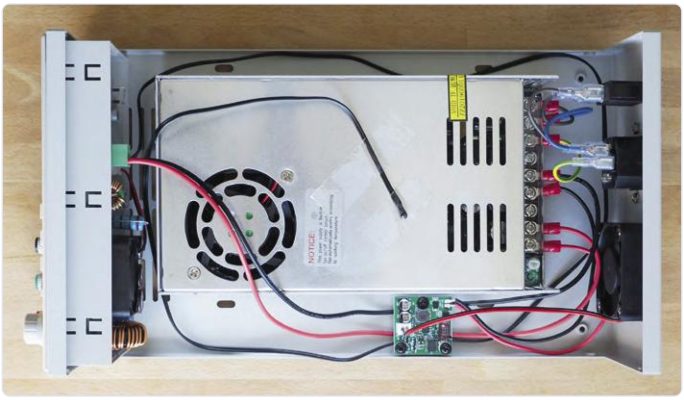


Figure 7: All parts installed and wired.

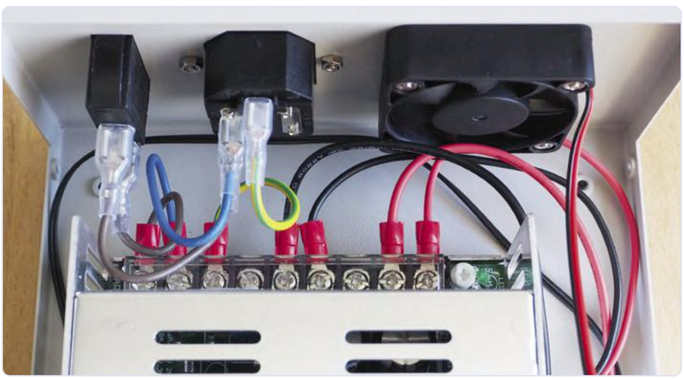


Figure 8: Detailed view. Take extra care fixing the wires that will be carrying mains voltage.

Mounting the fan in the case is a bit tricky: the M3 nuts fit nicely onto the fan frame but the nut recesses are circular so the nuts just rotate and won't tighten unless you can pinch them somehow. The space around the nut is too tight for my needle-nose pliers to gain purchase. **Figure 5** shows the blade of a very fine screwdriver being used to stop the nut turning. The fan is positioned so that its cable is on the right as shown.

Now you can screw down the open-frame power supply (**Figure 3e**) into the enclosure using four M4 x 6 pan-head screws. Make sure that the connections of the power supply point to the rear (left). The four rubber feet can be fitted in the corners using M3 x 5 screws. **Figure 6** shows how it should look.

**Figure 7** shows how all parts are wired. The terminals on the industrial power supply in particular should be tightened well, but not too tightly. **Figure 8** shows the close-up. You should always double-check when wiring equipment which will be powered from the mains. All wires are supplied finished to a suitable length with their ends either tinned, provided with cable lugs or plugs.

Before you push the front module into its recess in the housing, as shown in **Figure 7**, you should plug the supplied ESP12F Wi-Fi module (**Figure 3f**) onto the back of the front module board. **Figure 9** shows the front module from behind with the Wi-Fi module attached. Under this module is a battery compartment to accept a CR1220 button cell. this does not need to be fitted. The green 2-pin terminal block labelled 'Voltage in' can be removed and screwed with the longer red and black cable and then plugged in again.

If you now screw the cover back onto the housing, the finished benchtop power supply should look like the image in **Figure 10**. It is interesting that I have a few screws and two cable-end lugs left over after the build. There is no power cord included in the kit.

## Operation & adjustment

**Figure 11** shows four different operating states. But before I describe them, a few more comments: the coloured display is very informative and shows more information than strictly necessary. In the standard mode shown (there is another one that shows 'trends') you can view the current four-digit values of voltage (green), current (blue) and power (red) on the left. At the top right you can view the input voltage to the regulator section. This is the output voltage of the industry-standard closed-frame power supply which is displayed here as the 'INPUT'. This is followed by the set voltage (U-SET) and current (I-SET) and then the set protection against overvoltage (OVP) and overcurrent (OCP).

In **Figure 11a**, the current values are all still zero since the output is switched off. When you press the ON/OFF button on the right beneath the rotary encoder, it lights up and the output is switched on. In **Figure 11b** you can see the currently applied and actually measured voltage of 5.00 V, which is exactly the same as the requested set voltage. No current flows without a load - the power supply is in constant voltage mode (CV at the bottom in white). My measurement showed exactly 4.996 V, which amounts to a negligible error of -0.08% indicating that the power supply was properly calibrated. My older 480 W power supply can only show

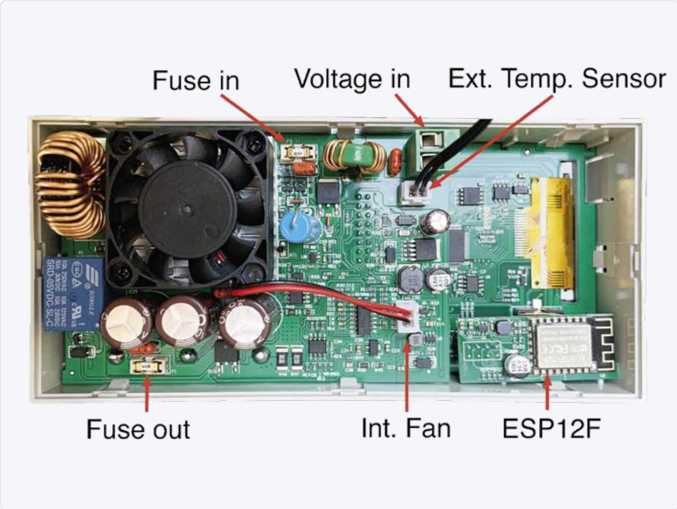



*Figure 9: The front module from behind with an attached ESP12F module. This module contains a holder for a lithium button cell (not required).*

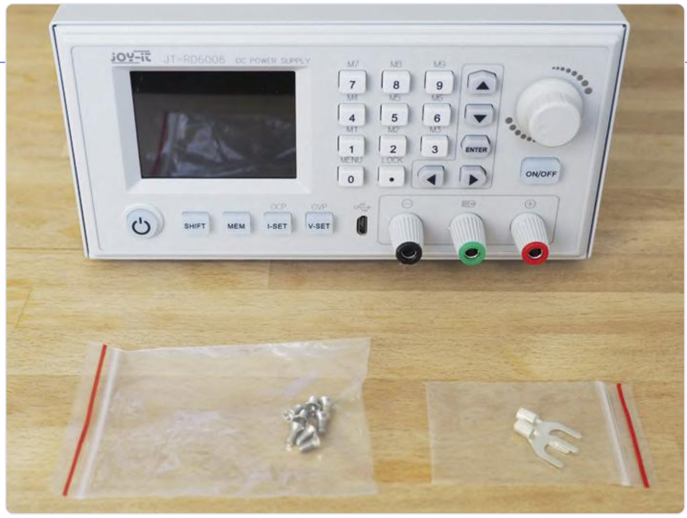


*Figure 10: The finished benchtop power supply. I ended up with eight M3 countersunk screws and two cable lugs leftover.*

output voltage and current levels, here you can see any differences, and that's useful.

**Figure 11c** shows my attempt at setting the maximum possible output voltage but only achieving 59.18 V instead of the rated value of 60.00 V. Why the discrepancy? The complete PSU is made up of an industry-standard closed frame power supply outputting a fixed 60 V supply fed to a regulator section. The regulator takes this 60 V and outputs a voltage according to the values between 0 and 60 V dialled up on the front panel controls. A small voltage loss introduced by the regulator means that its maximum output voltage cannot be as high as the input voltage. We can remedy the situation by tweaking the output voltage from the closed frame power supply to just over 60 V. I increased this input voltage by adjusting the small, white potentiometer at one end of the terminal strip shown in **Figure 7**.

Now with the input voltage adjusted to 61.5 V you really can set the output voltage up to 60.00 V. In order to avoid any problems under load, I increased the value further to give a small reserve. **Figure 11d** shows that an input voltage of 62.56 V is sufficient to ensure 60 V output even under full load. At 5.885 A the output is

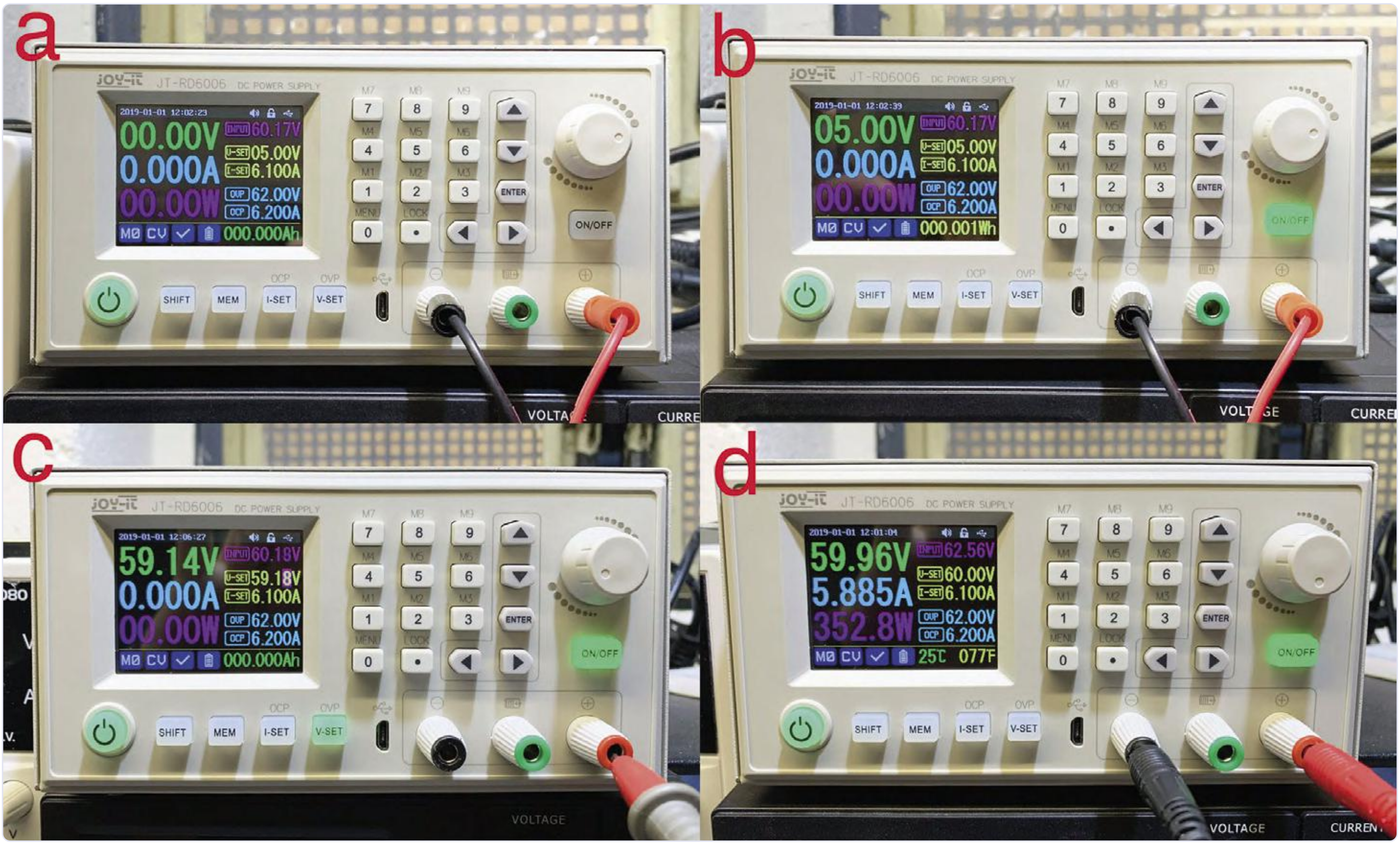



*Figure 11: Display in four different modes. 11a: output switched off; 11b: output with 5 V active; 11c: output max. 59.14 V with input = 60.13 V; 11d: Output 59.96 V with input = 62.56 V.*

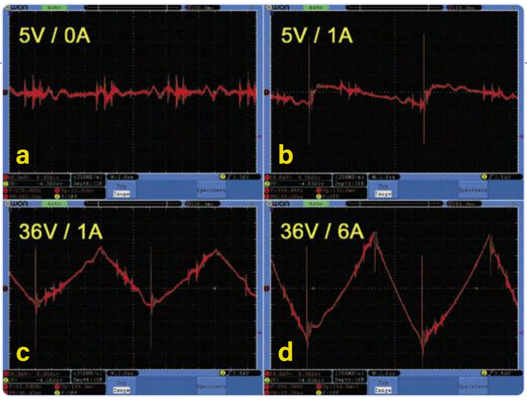



*Figure 12: Screenshots of the output voltage noise levels under four different load conditions.*

measured at 59.96 V and cables to my load resistors became warm. My multimeter measured 59.94 V which indicates a completely negligible error. So by making a few adjustments the benchtop power supply runs quite perfectly. The cooling fans are also barely audible and the internal temperature only reached 29 °C after 15 minutes at 353 watts output power.

## Output noise levels

It is generally true that the output voltage produced by a budget-level switching power supply will be electrically noisier than that produced by a supply designed using linear technology. I hooked up my 'scope to the output terminals to measure the noise levels. **Figure 12** shows noise on the output voltage under four different load conditions. The first thing to note was that no 50 or 100 Hz signal was discernible on the output voltage - so far so very good. Using AC coupling and by increasing the sensitivity of the scope's input amplifier I was able to see noise on the output DC level. I adjusted the trigger level and time base setting until the noise waveform was stationary on the scope display. With the time base set to 2 µs/div I could see periodic signals with a repetition rate of about 115 kHz — this is likely related to the switching frequency of the regulator. My scope shows the noise waveform on the output voltage level with no load connected. Ripple and noise amount to around 22 $mV_{pp}$ (**Figure 12a**).

Using a relatively small 5 W load (**Figure 12b**), the repetition rate of the noise spikes drop to around 70 kHz with an amplitude of 130 $mV_{pp}$. Additional filtering at output would help reduce the noise amplitude. At a moderate load of 36 W (**Figure 12c**), the repetition rate of the spikes is the same, but the noise waveform is more triangular as more energy is delivered to the load and supplied by the regulator resulting in a noise level of about 70 $mV_{pp}$. By increasing the load power to 200 watts you can see in **Figure 12d** that the amplitude of the triangular noise waveform almost doubles to around 130 $mV_{pp}$. This exceeds the maximum 100 $mV_{pp}$ specified by the manufacturer but it would be acceptable for the majority of tasks undertaken by a high-power digital benchtop power supply.

## Conclusions

I did not try to control the power supply remotely via USB or WiFi. The information given in the manual was rather cryptic and to be honest I've never had a need to do anything like that either. For my needs I like to be able to adjust voltage and current limit by hand directly via the front panel controls. I simply ignored the other built-in luxury features because I can't imagine when I would need them. This benchtop power supply is also said to be able to charge batteries at defined values. There is even a separate green output socket for the positive pole connection. Here the current is switched off when it falls below 10 mA so that the battery cannot be overcharged. I couldn't work out why this feature requires a dedicated socket. To understand how the battery charging feature works we need the skills of an Alan Turing to decode the relevant passages in the German manual. The English version doesn't make much sense either; it was probably translated from Chinese too.
As far as its features are concerned, I liked this benchtop power supply better than the one I have been using for years. The display in particular is great because it shows target and actual values at the same time. I also like that (up to ten) presets for voltage and current limit can be preprogrammed and selected easily. The setting of the values is via a rotary encoder (fast & intuitive) or via the keypad (exactly) is also very useful. I like the ability to adjust the voltage and current limit very precisely. The electrical noise level on the output voltage is higher than can be expected with a linear power supply, but in practice (except when powering sensitive RF parts) it is not high enough to cause problems. After switching on, the microcontroller boots really quickly so that the power supply is ready for use within a second if you switch off the display. For me that's a revelation; my old heavy-duty power supply, takes almost 15 s before its ready to go. The price-performance ratio of this supply is good and if any component part fails it will be easy to replace individual parts.
Now to the downsides: It's annoying that no mains kettle lead is included in the kit but even more frustrating is the poor quality of the documentation. This is however what we have come to expect from all budget-level equipment sourced from the Far East. It turned out that this was not too much of a handicap because control of all the supply's basic features is fairly intuitive and I didn't feel I needed the additional 'mystery' functions.

200130-02



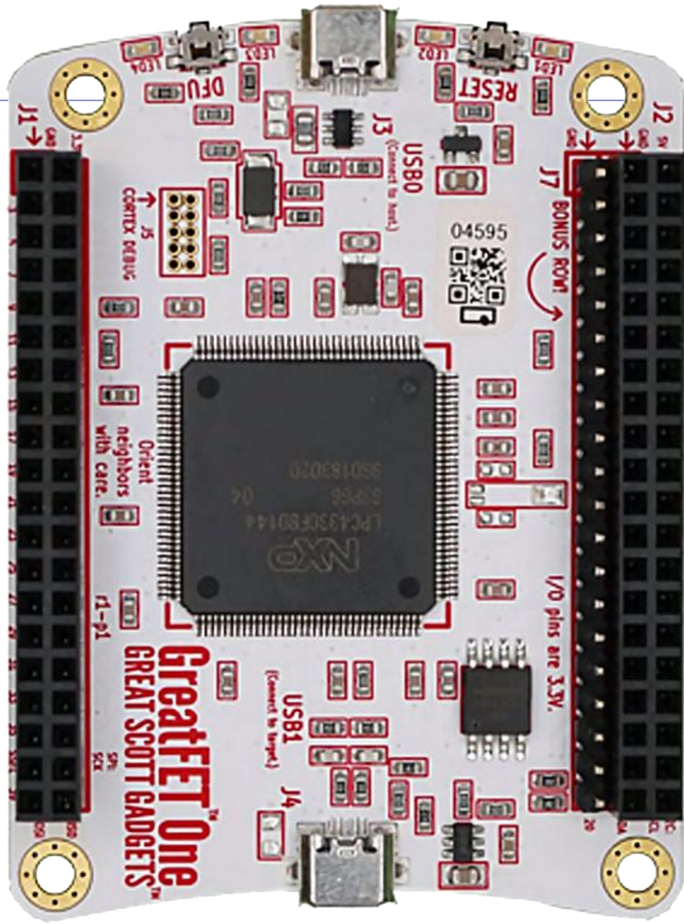
## WEB LINKS

[1] **Review**: https://www.elektormagazine.com/news/reviewthepeaktech6080alabpowersupply

[2] **Joy-iT RD6006**: https://www.elektor.com/joy-it-jt-rd6006-dc-power-supply-bundle

[3] **Operators Manual**: https://joy-it.net/files/files/Produkte/JT-RD6006/JT-RD6006%20Manual-A4.pdf

# The GreatFET One Interface Board



By **Tam Hanna** (Slovenia)

In days of yore before all PCs came with USB ports we were busy toggling pins on the PC's parallel port and communicating via RS-232. Nowadays we outsource that activity to a processor and talk to it via USB. The Great Scott GreatFET One board connects to the USB port and allows you to hack all sorts of intimate USB stuff, using Python code.

Back in the day before USB became the *de facto* communication port for PCs you would most probably develop control software to run on the PC and interface to external events via the parallel port. Nowadays we don't have that option; while interaction with the gameport, serial interface or printer interface was comparatively easy, working with the USB interface without the support of dedicated chips requires good programming knowledge.
With the GreatFET One, Great Scott Gadgets now offers a USB development board that allows you to control general GPIO signals from a PC much like you could toggle pins on a PC parallel port in the "good ole days".

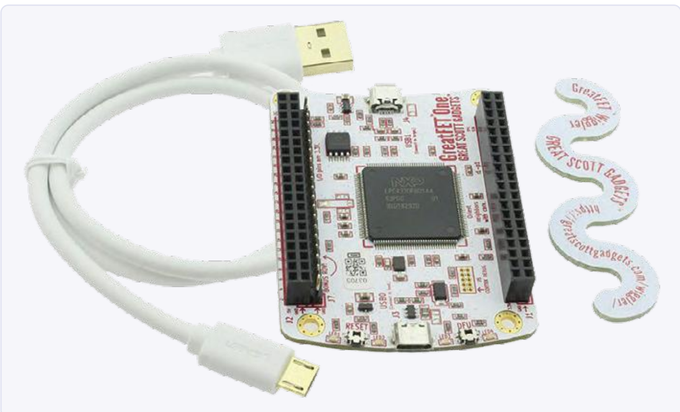


*Figure 1: The GreatFET One...out of the box!*

## What's in the box?

The GreatFET One is designed as open source hardware — a description of the software can be found in a wiki [1], the hardware files are useable by KiCad are also available for download on Github [2]. If you buy the GreatFET One, you get the package shown in **Figure 1** — in addition to the (ready-to-use and equipped with all headers) board, you also get a USB cable and a snake-like Wiggler tool for safely separating external plug-on expansion boards. Documentation for the main interaction API called *LibGreat* can be found under [3].
You may be a little confused by the board's name — no... there is no massive Field Effect Transistor anywhere that I could see — FET in this case stands for Flash Emulation Tools which gives a clue of this original idea behind its design. This was to make some USB device that could communicate with a PC and exchange signals acting like it was a flash memory stick. Once you achieve that you can of course use this emulator to talk with the PC and pretend it is almost any USB device... sounds like a hackers best friend!

## Quick setup!

The Python programming environment is usually a good choice for a system developer who wants to quickly cobble together some code to test hardware. The author put together this quick demonstration in the following steps using Ubuntu 18.04 LTS. If instead you want to work in a Windows environment, you can find the

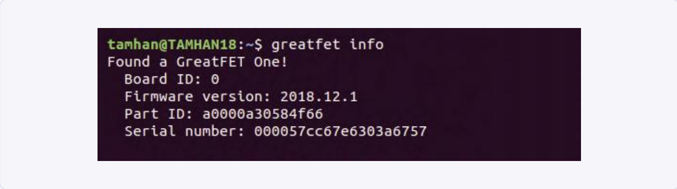



*Figure 2: The GreatFET One board has been identified.*

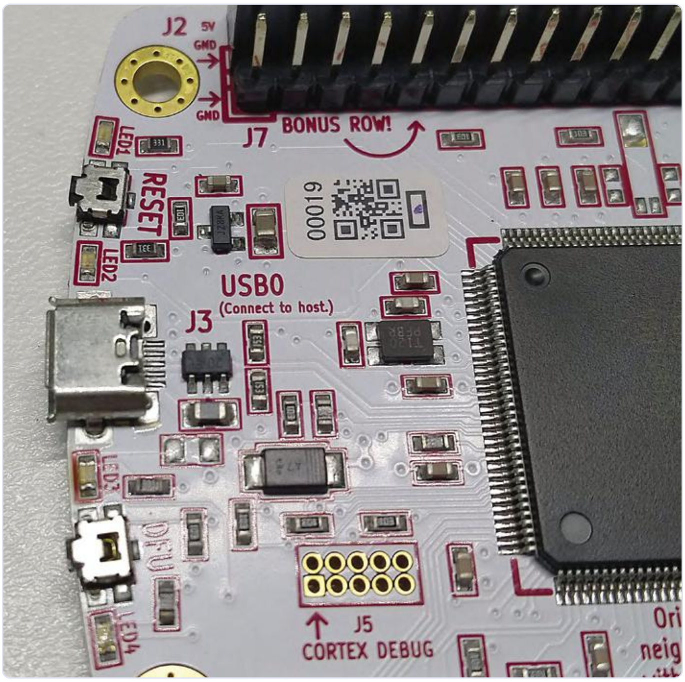



*Figure 3: Clear labelling is a bonus. This USB port connects to the host.*

necessary installation instructions online [4].
Installation of The library begins using the pip3 package manager:

```
tamhan@TAMHAN18:~$ pip3 install --upgrade --user
   greatfet
```

Python has had its own package management for some time now, via which we can load the GreatFET package, including some of its native dependencies. It is important to use *pip3* version - the *pip* intended for Python 2.X does not work. In the next step we check whether our user account is already a member of the PlugDev group:

```
tamhan@TAMHAN18:~$ groups tamhan
tamhan : tamhan adm dialout cdrom sudo dip plugdev
   lpadmin sambashare kvm pico
```

Before connecting the process computer to the computer, we enter the following commands to update the Linux udev-rules:

```
tamhan@TAMHAN18:~$ sudo wget https://raw.
   githubusercontent.com/greatscottgadgets/greatfet/
   master/host/misc/54-greatfet.rules -O /etc/udev/
   rules.d/54-greatfet.rules
tamhan@TAMHAN18:~$ sudo udevadm control -reload-rules
```

It is worth the effort; it can now recognize a connected process computer - as shown in **Figure 2** — by entering the `greatfet info` command.

Whenever possible its always worth checking that any new piece of kit is running the latest firmware version. For the GreatFET we can enter the following command:

```
tamhan@TAMHAN18:~$ gf fw --auto
```

In the following steps, we are using version v2020.1.2 of the firmware.

Combinatorial process computer systems generally suffer from poor bandwidth between individual modules. For a simple test we can write a small routine to toggle a GPIO pin on the board to see how fast we can get things moving. For this we will use a .py file, which you can easily edit in Visual Studio code:

```
tamhan@TAMHAN18:~/greatfetspace$ code worker.py
tamhan@TAMHAN18:~/greatfetspace$ python3 worker.py
```

A nice side aspect of the Microsoft IDE is that the terminal returns immediately after activation - the Python-3 Interpreter can therefore be called up at the same location.

In the next step we can create a GreatFET-Object and create a pin object according to the following scheme:

```
from greatfet import GreatFET

gf = GreatFET()
pin = gf.gpio.get_pin('J1_P4')
pin.set_direction(gf.gpio.DIRECTION_OUT)
```

Another nice feature is the pin assignments; when working with a RPi there can be confusion regarding a pin allocation in software and its physical location on the board.
Great Scott bypasses the problem by assigning a unique label to each header. The string used here then refers, for example, to the pin 4 of header J1 (see **Figure 3**).

This is followed by an endless loop that toggles the pin and outputs a waveform:

```
while 1 == 1:
    pin.write(True)
    pin.write(False)
    pin.write(True)
    pin.write(False)
```

**Figures 4 and 5** shows the output signal generated and we can see that the high and low periods of the square wave are approximately equal. The switching time is not particularly fast nor stable.

When experimenting, please remember the board uses 3.3 V logic; the controller will most probably be destroyed if you try connecting 5 V signals to it.

## Experiments

The GreatFET One comes with an analogue-to-digital converter, which is easiest to activate with one of the many command line helpers. The default input pin to the A/D converter is pin J2_P5:

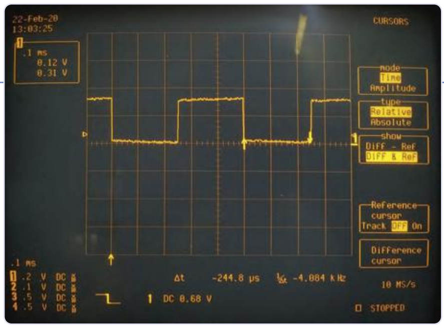


*Figure 4: The output waveform displayed on a scope...*

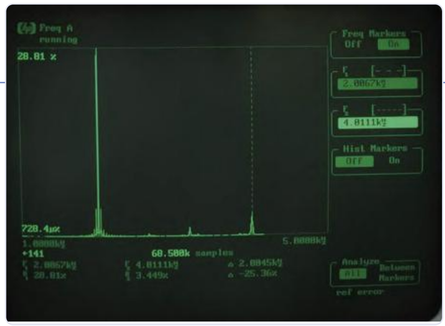


*Figure 5: ... and on a modulation domain analyzer.*

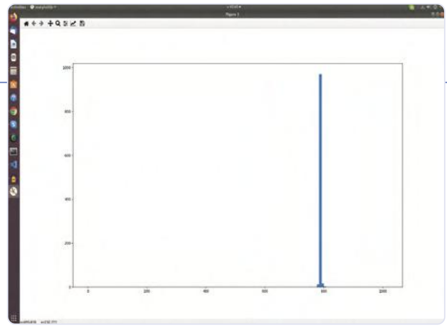


*Figure 6: A simple histogram.*

```
tamhan@TAMHAN18:~/greatfetspace$ greatfet adc
3.18076171875V
```

To test the performance of the Python code running on the hardware, I will therefore need to load both the *NumPy*-Library and *MatPlotLib* at this point - if the two libraries are missing, you need to install them manually. In the next step I will use a temperature controlled voltage reference source. In the meantime I will use the following program.
In the first step, I load some libraries as usual:

```
import numpy as np
from matplotlib import pyplot as plt
from greatfet import GreatFET
```

The implemented ADC reading function is a complex matter. For convenience, I create the NumPy array — using the API is not very efficient:

```
gf = GreatFET()

store = np.array(gf.adc.read_samples(1))
i = 0
while i < 1000:
    store = np.append(store, gf.adc.read_samples(1))
    i=i+1
```

At this point there are two different procedures available. Method number one is the creation of a *Bin-Ranges*, which I then pass to the NumPy `histogram` function:

```
binrange = range(0,1024,8)
hist, bins = np.histogram(store, binrange)
```

The effort of this method is rewarded by returning the array in which the frequency of values in the respective `bins` can be collated. The transfer of `bins` is then just a copy of the `binrange` parameter.

If you only want to display the diagrams directly, you can outsource the histogram display directly to MatPlotLib according to the following scheme:

```
plt.hist(store, bins)
plt.show()
```

The histogram shown in **Figure 6** is worth the effort — please ignore any EMI noise.
Finally, it should be noted that the GreatFET also offers a pattern generator and even a function for exporting Sigrok-Log-Information. Both can be activated according to the following scheme using commands:

```
tamhan@TAMHAN18:~/greatfetspace$ greatfet logic -p
   out.sr -f 2M -n 4
```

The maximum speed attainable depends on the hardware used - there is an interesting discussion of the topic under [5].

## To sum up

The documentation for this device is a little sparse but with all the board's potential as a tool to get under the skin of USB comms and to provide a USB breakout function I imagine it would be a favourite and gain support from the Maker and Hacker community. I was only able to tap into a small fraction of what this board is actually capable of... it has lots of potential.

200124-02



## WEB LINKS


[1] Software: https://github.com/greatscottgadgets/greatfet/wiki

[2] Hardware Description: https://github.com/greatfet-hardware/azalea

[3] LibGreat Docs: https://github.com/greatscottgadgets/libgreat

[4] Installation in Windows: https://greatscottgadgets.github.io/greatfet-tutorials/windows.html

[5] Sample rates: https://github.com/greatscottgadgets/greatfet/issues/286

# Small Circuits Revival – XXL

## Escaped from the Suggestions  Box

Compiled by **Eric Bogers**

During the past six months we've been publishing a Small Circuit every week on our website and in our newsletter. The circuit can be assembled without any special, exotic parts and with a normal soldering iron on breadboard. Now it's a circuit 'from the old days' and then a recent entry from a reader, or an idea from our lab. These circuits are discussed on the website — sometimes controversially — and that' s exactly our aim: to stimulate discussions, experiments and readers' own discoveries! Here is a selection from the first 26 episodes.

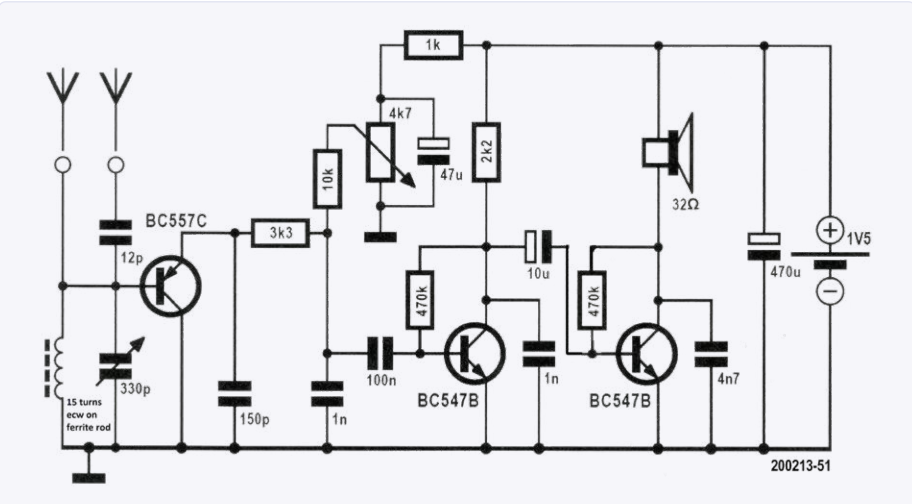



*Figure 1: The diagram of this receiver is not complicated and the reconstruction is far from critical.*

***idea: Burkhard Kainka (Germany)***

Many (beginning) electronics enthusiasts shy away from building radio receiver projects — frequencies are too high, capacitor black magic, scary inductors... The reality of a receiver that actually performs very well using only a handful of standard parts is demonstrated by this circuit.

### Shortwave Audion

Besides the tuning capacitor, the three-stage short-wave receiver described here (**Figure 1**) has an additional regulator for the feedback. In principle, this first stage is an oscillator whose operating point is ajustable, allowing the gain to be reduced. The trick is to adjust the gain such that that all the losses in the oscillator loop are just about compensated for, that is, the whole thing is on the verge of self-oscillation. At this point the receiver exhibits its greatest sensitivity and best selectivity.

The PNP oscillator stage (in common-collector configuration) operates simultaneously as an audion and a demodulator for the RF signal. The two AF stages that follow ensure that there is sufficient audio power so that even a small loudspeaker can be driven.

Whether the short-wave audion can actually be tuned to the point where it is just not oscillating also depends on the damping of the input circuit and the antenna connected. That is why this circuit is provided with two antenna connections. When connected via the 12 pF capacitor, it is loosely coupled and the damping is small. By contrast, the direct connection is (also) suitable for very short antennas mianly because a longer antenna also radiates RF energy which damps the oscillator circuit.

When adjusted correctly, an audion is extremely sensitive; in the early days of radio technology these types of receivers were part of the standard equipment. Even with weak transmitters, distances of up to a thousand miles could be covered.

***idea: Elex Team***

Conceded: twilight switches with lots of bells and whistles are available for next to nothing. But isn't it nevertheless much nicer to build something like that yourself — with literally 10 components?

### Simple Twilight Switch

A twilight switch is an (electronic) switch which turns on a lamp, or such like, when it darkness sets in — with, among other things, the intention of deterring potential burglars by suggesting that someone is at home.

These days you can buy these things for little money from the Far East, but building one yourself is not difficult at all, as the schematic in **Figure 2** proves.

We need, of course, a light sensitive element that will let us know when twilight falls. Here that is LDR R2 (*Light Dependent Resistor*). Its

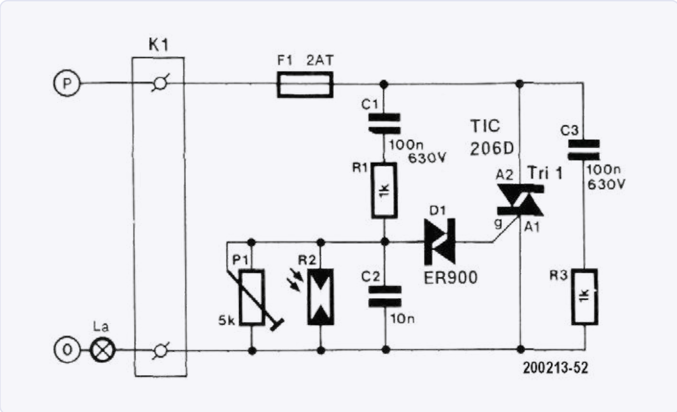



*Figure 2: The most 'exotic' part in this schematic is probably the diac.*

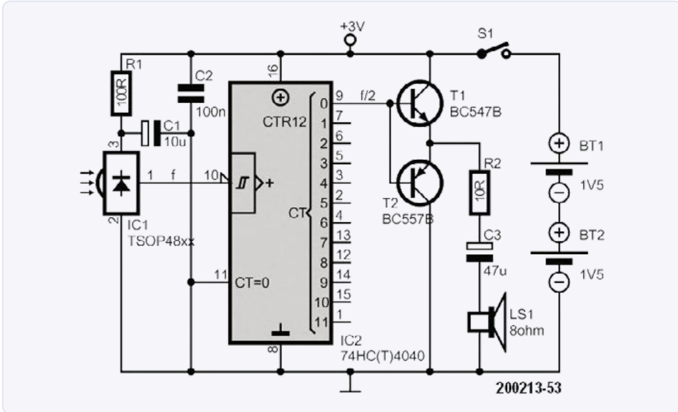



*Figure 3: To make IR remote control signals audible, all you have to do is divide down their frequency.*

resistance increases as the amount of light that falls on it decreases. With that the secret of the circuit has actually been revealed: the combination: C1, R1, C2, R2 and P1 form an light-dependent (AC) resistor. The full AC voltage of 230 V appears across this combination. As it becomes darker outside (which, as we have been let to believe, occurs once a day), the resistance of R2 increases, and with that also the voltage at the junction of R1 and C2.

At a given moment (adjustable with P1), the AC voltage at this junction reaches a value of about 30 V, and diac D1 will conduct. As a result, triac Tri1 will receive a trigger pulse and starts to conduct — the lamp will turn on. Now we have already noted that the entire circuit is connected to the mains voltage; meaning the triac will receive a trigger pulse every half period of the mains voltage, and will block again at the next zero crossing (when the current becomes zero). This whole process repeats itself one hundred times per second (assuming 50 Hz).

We have used a diac here to ensure that the triac receives a well-defined trigger pulse (capacitor C2 lends a hand here by acting as an 'electron reservoir' — since once the triac starts to conduct it will act as a short circuit and the voltage across the light-sensitive voltage divider will reduce to zero.

**Construction**

For the third time: the entire circuit is connected to the mains! The construction therefore has to be done with the required amount of care, and keeping in mind all the applicable guidelines. It is best to build it all into a touch-safe, plug type enclosure. In the case of any doubt, show it to an experienced electrician/electronics technician first!

***idea: Ton Giesberts (Elektor Labs)***

Have you ever wondered what an infrared remote control sounds like? With this small circuit you will be able to find out!

## Acoustic IR Remote Control Tester

In this circuit we use the output signal from a standard IR receiver (IC1 in the schematic of **Figure 3**) to drive a miniature loudspeaker. The IR receiver from Vishay used here is available for various frequencies ranging from 30 to 56 kHz. A large number of IR remote controls operate according to the so-called RC5 protocol that uses a frequency of 36 kHz. In our prototype we used a TSOP4836; and as the part number suggests, it is intended for 36 kHz operation. However, it will also 'work' at other frequencies, if they are not too different.

The signal at the output of the IR receiver is at too high a frequency to generate an appropriate sound signal; that is why we first pass the signal through a divide-by-2 IC (IC2). You can use the signal from any of the outputs, depending on your personal preference (we used output '0' (pin 9) — the input signal divided by two).

We use this signal to drive a very simple push-pull output stage (T1/T2), that nevertheless has enough power to drive a small loudspeaker.

The circuit is powered from 3 V (two AA-batteries); the current consumption amounts to about 13 mA maximum (0.66 mA when idle). R1 and C1 decouple the power supply voltage for the IR receiver to prevent any potential interference from the output stage.

Using this circuit (which you can build on a breadboard or prototyping board) you cannot only check whether an IR remote control is still functional, but you can also compare different brands and models with each other 'by ear'. Admittedly, the usefulness of this tester is limited, but the circuit is good for a Sunday afternoon of crafting and can serve as a basis for other developments.

***idea: Elex Team***

Legions of dog whistles are available from pet shops and by mail order. But with only a few parts you can easily build one that yourself!

## Electronic Dog Whistle

Ultrasonic dog whistles are used in situations where the owner would like to give commands to the dog, and that will only be heard by the dog – for example for training purposes.

The circuit described here can, however, also be used in 'reverse': to scare off aggressive, unleashed dogs — this is because the circuit generates a very loud sound that nearby dogs definitely find unpleasant.

The circuit is simplicity itself, see **Figure 4**. Transistors T3 and T4 together form a classical astable multivibrator (call it a free-running oscillator) with a frequency of about 21 kHz (with the component values as indicated). Because of the completely symmetric design, the circuit generates a nice symmetric square wave. Diodes D1 and D2 prevent the circuit from generating a lot of audible noise in addition to the desired ultrasonic square wave. The 21-kHz square wave is substantially amplified by the two push-pull output stages (T1/T2 and T5/T6) and drives the piezo tweeter LS1. The original design used a Monacor type KSN1001A (or equivalent) for the tweeter; if you look around on the Internet you will find plenty of 'modern' variants.

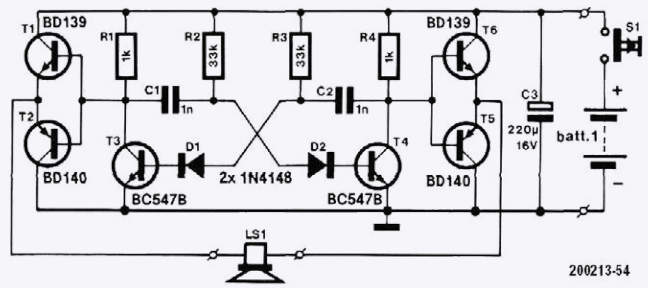



Figure 4: Actually, this is nothing more than a discretely constructed astable multivibrator with power output stages.

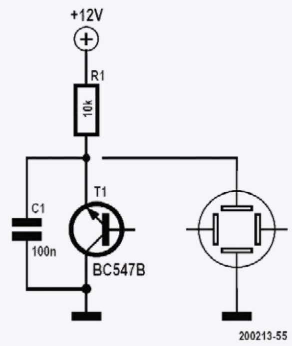



Figure 5: The 'original' relaxation oscillator and its transistor equivalent.

The circuit was designed to be powered from a 9V 'block' battery (PP3 / 6LR22). Finally, a word of warning: because of the large sound volume produced you should not use the circuit in the close proximity of sensitive dog ears — unless you are attacked, of course.

And naturally you can experiment to your heart's content with this circuit — with different component values and a normal loudspeaker you could also build a bicycle siren or whatnot!

One of our readers suggested that after a few modifications this circuit could even serve as a loitering youths repeller in porticoes and the like...

***idea: Burkhard Kainka (Getmany)***

In the past, a sawtooth was often generated using no more than a small neon lamp and a capacitor, and the resuting circuit was called 'relaxation oscillator.'

## Battery Keep-Alive

Less well known, the effect can also be achieved with an ordinary NPN small signal transistor like a BC547 as sketched in **Figure 5**. It is too much to explain here in detail how this can work — let it be sufficient to state that it works and we refer you to the original publication in Elektor's P*roject Generator Compendium 2011* (page 80) for the details.

Lead-acid gel batteries have the undesirable characteristic that they go 'high impedance' i.e. lose their capacity if they haven't been used for a (very) long time. And while it is possible to revive such a 'dead' battery, this is a tedious and lengthy process. It is, of course, much better to avoid this state in the first place.

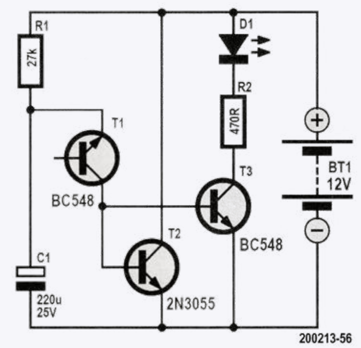



Figure 6: The battery 'keep-alive' contains remarkably few parts.

A tried-and-tested method to prevent the above is to load the battery continuously with very short but reasonably large current pulses. This is achieved with the circuit of **Figure 6**;

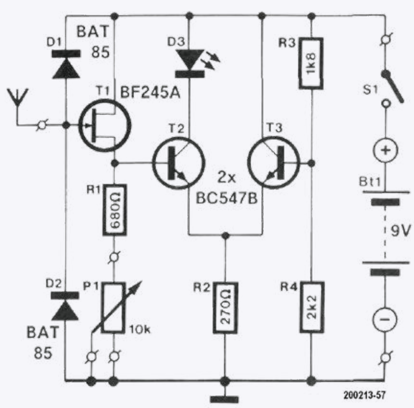



Figure 7: Not as refined and versatile as a modern conduit searcher, but on the other hand, this model can be built in a snap.

the alert reader will recognise the transistor relaxation oscillator from Figure 5.

The oscillator around T1 drives the base of power transistor T2, which will discharge the battery with 1-amp pulses having a duration of about 2 ms when using the component values shown here. This corresponds to an average current of about 1 mA, which is of the same order of magnitude as the typical self-discharge current of a lead-acid gel battery.

The combination of T3, D1 and R2 gives a rough visual indication of the state of charge of the battery; the less frequently the LED lights up, the lower the battery voltage.

***idea: Elex Team***

Apart from the risk of a (dangerous) electric shock and the near certainty of a very massive short circuit that is the consequence of drilling into a cable, the aftermath of this mishap is no fun either.

## Cable & Conduit Locator

Apart from the risk of a (dangerous) electric shock and the near certainty of a very massive short circuit that is the consequence of drilling into a cable, the aftermath of this mishap is no fun either. In any case, the damaged cable has to be replaced – and that is already not nice when these are running neatly through conduit, but in other countries where the wires are often tidily hidden under the plaster work, this is a small catastrophe.

Prevention is better than cure, and that is why the better hobbyist makes use of a cable locator. But what to do on a rainy Sunday afternoon, when it is absolutely essential that a hole is drilled, but the DIY store is closed and you cannot quickly buy a cable locator?

Then you build one yourself in a few minutes! Complicated it isn't, as the schematic of **Figure 7** proves.
Three transistors and a few other bits — that's all. An "antenna" of sorts is connected to the gate of FET T1 (a solid piece of copper wire about 20 cm long); this picks up the ubiquitous 50 Hz or 60 Hz 'mains hum.' The antenna is not loaded because of the very high input impedance of the FET. Diodes D1 and D2 (Schottky types BAT85) prevent the voltage at the gate from going too high.
The buffered antenna voltage is then available at the source of T1; the DC voltage level can be adjusted with P1. The source voltage goes to the base of transistor T2, which acts as a kind of comparator: it compares the voltage at its base with that at the emitter. And this latter voltage is generated by T3. With the component values shown here, the voltage at the top of R2 is about 4.25 V, and tconsequently T2 will start to conduct when its base voltage is higher than 4.95 V. LED D3 will then also light up.
P1 is used to adjust the circuit so that in the idle state the LED just remains off. The circuit can be built on a piece of prototyping strip board in a jiffy. And of course, this circuit also perfectly lends itself to doing your own experiments!

***idea: Kees van der Geer (Netherlands)***

Simply forgot to water it — and then you can wave goodbye to that expensive bonsai. A pity about all the work that went into it… But this dry soil alarm will save you from these disasters!

## Dry Soil Alarm

The idea behind this circuit is actually simplicity itself: as the soil around the roots of the plant dries, its (electrical) resistance increases. Measure this resistance in one way or another; and once it becomes too high (and therefore the soil in the pot is too dry) raise the alarm. This idea is realised in the schematic of **Figure 8**. The circuit essentially consists of the two comparators in a TLC3702. The first is configured as an oscillator (astable multivibrator); the period is about 2 seconds with the component values as indicated.
Brief positive- and negative-going pulses arrive via capacitors C2 and C3 at the inverting input of the second comparator, which is configured as a 'one-shot' (C2 and R5 plus both the electrodes effectively function as a differentiator).
Depending on whether the soil is more or less moist, a larger or smaller amount of the pulse is diverted via the pair of electrodes to ground. Or expressed differently: as the soil in the pot dries out, an increasingly less-attenuated pulse arrives at the second comparator. The positive-going pulses have no further effect, but because of the alternating positive- and negative-going pulses, no corrosion of the electrodes will occur — a hidden advantage of this design.
When the negative-going pulse at the inverting input of the second comparator becomes large enough, the one-shot will be triggered and the LED will light up for about 100 ms; at the same time the buzzer will generate an urgent beep. This will continue until the thirsty plant has been watered.
In itself this circuit is not really amazing; the nice feature, however, is that it is extremely energy efficient. Even when the LED and buzzer are continually activated every couple of seconds, the battery (a 9-V PP3 block) will last for months. This is a benefit of the IC used here in combination with the extremely high resistance values.
Note: at the moment we are working on an update of this circuit; we will come back to that in the next issue!

***idea: Friedrich Lischeck (Germany)***

In a not yet dim past we started this series with an 'energy-efficient relay.' Here is a variation on that topic, and a super-simple one at that.

## Energy-efficient Relay

The big disadvantage of the energy-efficient relay that was described in SMR Episode 1 is that it requires pushbuttons with a normally closed (n.c.) contact. These are sometimes difficult to obtain, and most hobbyists are not likely to have these on hand.
The small but clever circuit that we will take a look at now employs normal pushbuttons (i.e. n.o. , normallly open with a momentary contact) and a standard 5-V monostable relay with two (changeover) contacts — see **Figure 9**.
The nice feature of this circuit is that the original monostable relay (that is, a relay of which the normally-open contact is closed only for as long as the coil is energized; without a coil current the contact returnes to the rest state) is used here as a bistable relay: after pushing 'ON' the normally-open contact remains closed until 'OFF' is pushed.
To make this possible, one of the pair of contacts in the relay is used for this; it is

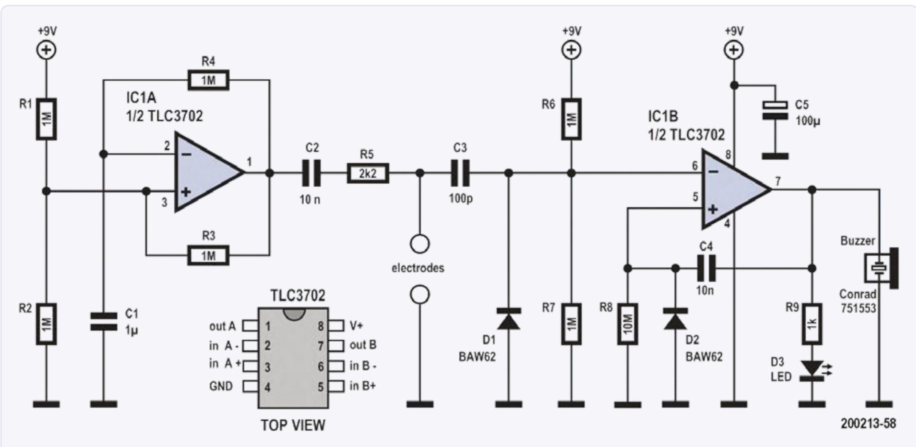



*Figure 8: Two comparators form the heart of the dry soil alarm circuit.*

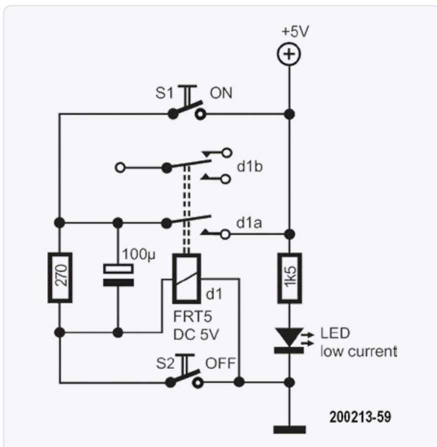



*Figure 9: The originally monostable relay works here as a bistable relay.*

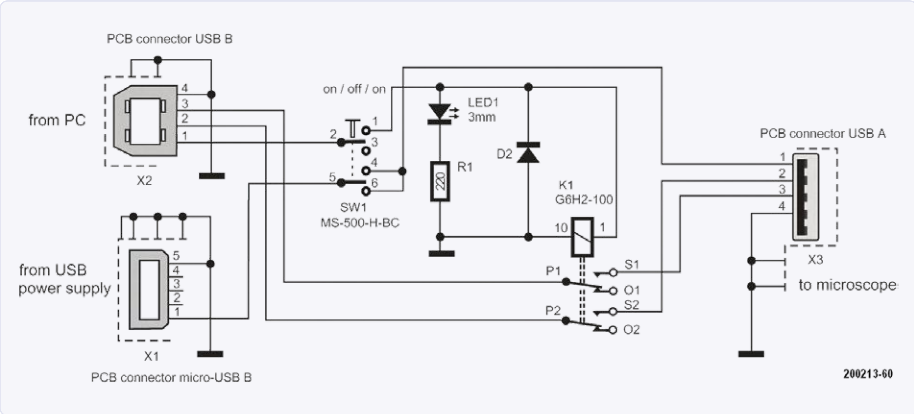



*Figure 10: Attention: the switch must be of the 'centre-off (on/off/on) type!*

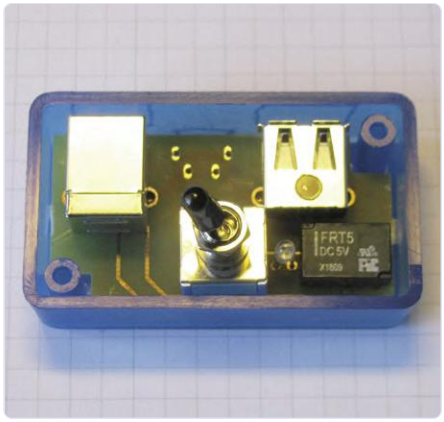


*Figure 11: The author designed a circuit board of which the files can be downloaded from the project page with this article.*

not difficult to figure out the operation of the circuit. The relay coil is energized after pushing S1; both contacts pull in – and the relay coil remains energized via one of the relay contacts and a 270 Ω resistor. Because of the big electrolytic capacitor, which acts as a short circuit at the moment of switch on, the initial energizing current through the relay exceeds the holding current in the energized state.
To switch the relay off it is sufficient to push S2: the relay coil is short-circuited; the relay drops out and the rest state is restored.
The LED indicates whether the circuit is ready to operate. A nice detail is that the relay does not close the instant that the power supply voltage is applied; this does require an actual push on S1. And should someone come up with the idea of pushing both buttons at the same time: not a problem, nothing happens because S2 has priority.

***idea: Dr Martin Oppermann (Germany)***

The Andonstar ADSM302 USB microscope is a popular aid for the 'more fiddly work' in the electronics lab (as is its 'bigger brother', the AD407). It does however have a couple of idiosyncrasies that can be mitigated by the small circuit described here.

## Andonstar ADSM302 Enhancement

The ADSM302 can be used as a stand-alone microscope or used in combination with a PC. In the first case you see 'live' images on the display of the microscope itself; and the second, the image is displayed on the screen of a PC.
When operating with a PC, the microscope is also powered from the USB port of the PC, and here is where the first problem manifests itself: a USB 2.0 port can supply a maximum current of 500 mA, which is insufficient (even with the illumination of the microscope switched off). A USB 3.0 port is acutely required.
To switch between stand-alone operation and PC operation, the cable between the microscope and PC has to be repeatedly connected and disconnected — something that is not really conducive to the longevity of the plugs and sockets.
The small circuit in **Figure 10** solves both problems in one fell swoop. Firstly, it is provided with a micro-USB socket allowing a USB mains adapter to be used to power the microscope.
And with the help of an on/off/on switch it's now possible to switch between stand-alone operation and PC operation. The two switching contacts of the miniature relay connect and disconnect the USB data lines from the PC to the microscope. The type of switch is important: the switching order is On-Off-On. When switching between the operating modes, the supply to the microscope is switched off so that it resets and the electronics can recognize the correct operating mode (stand-alone or PC). The LED lights up in the PC mode.
The circuit can be built on a small piece of prototyping board; for the USB sockets you could, if desired, 'sacrifice' a couple of cheap USB cables. Or you can (as the author did) design a real circuit printed wiring board, use real PCB-mount USB sockets, and build everything neatly into a small enclosure as shown in **Figure 11** and using the the EAGLE PCB files found as a download at [1].

200213-03



## WEB LINK


[1] **Project page**: www.elektormagazine.com/200213-03

# Poorest Man's External 2.4 GHz Wi-Fi Antenna

By **Hans-Henrik Skovgaard** (Denmark)

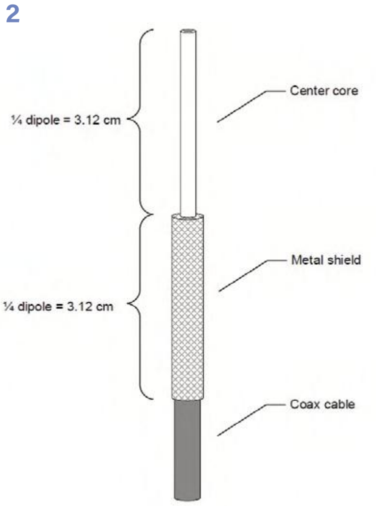
You can rightly argue: "why make your own Wi-Fi antenna, with all these cheap ready-made antennas available". Simply because I wanted to see how difficult it was. I also need to emphasise that this design is based on what was available and could be improved in many ways.
Before we start, note that you need to be very accurate and careful with your work in order to carry out a comparison against an available antenna, for example, of the ceramic type, or other ready-made antennas.
In order to have cables to use for the experiment, I bought 20 "IPX IPEX u.fl Female 1.13mm Connector Cable Single-head Adapter Connector 15cm IPX 1.13 Cable" (*sic*) from aliexpress.com (**Figure 1**).
When designing and building an antenna you need to know the wavelength of the signal the antenna is expected to radiate and/or pick up. The wavelength of a radio signal, λ, can be calculated with the following formula:

$$\lambda = c / f \, [\mathrm{m}]$$

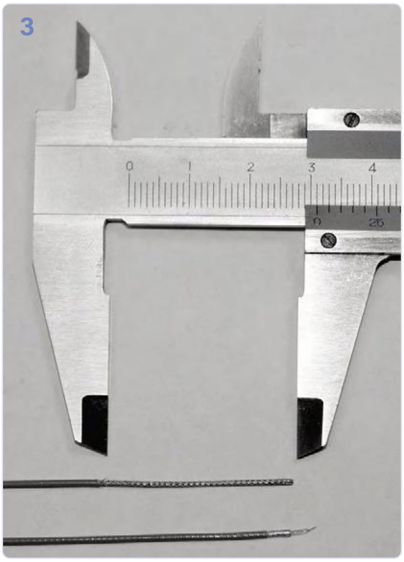
Where *c* is the speed of light (in m/s), and *f* is the frequency (in Hz). As we all know, *c* equals 299,792,458 m / s. For 2.4 GHz Wi-Fi this gives a wavelength of 0.1249 m = 12.49 cm. This value is important to know because the antennas are built to sizes which are certain fractions of the wavelength. For example, 1/4 of the wavelength (3.12 cm) or 1/2 wavelength (6.25 cm). With these figures we can prepare to cut and rework the aliexpress cable as shown in **Figure 2**.
First, cut off 3.12 cm of the plastic shielding as shown in **Figure 3**. The metal shield ('braid') is now exposed. This needs to be carefully untwisted from the inner plastic of the cable and twisted around the outer plastic as shown in **Figure 4**. You now hopefully have a simple ¼-wave 'monopole' antenna.
As you experiment, a utility like WifiScan may be used to see if there are any improvements over a Wi-Fi router's internal antenna. The easiest way is to place the 'antenna' at a fixed location and then see how many WiFi access points can be seen. Do the same with, for example, a Wemos PCB using the internal antenna.
After several uses of the WifiScan program and careful adjustments to the metal shield, I saw an improvement in the RSSI (Received Signal Strength Indication). RSSI is a term used to measure the relative quality of a received signal to a client device,in my case, an ESP8266) but has no absolute value. The RSSI value also varies greatly between chipset manufacturers. Therefore, it will only indicate if your design is heading in the right direction.

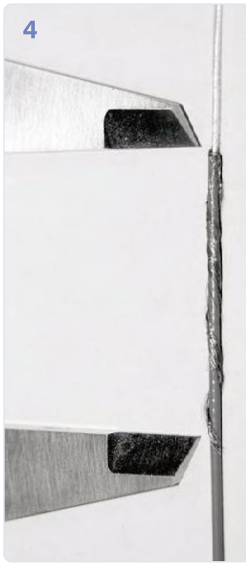
200207-01


1

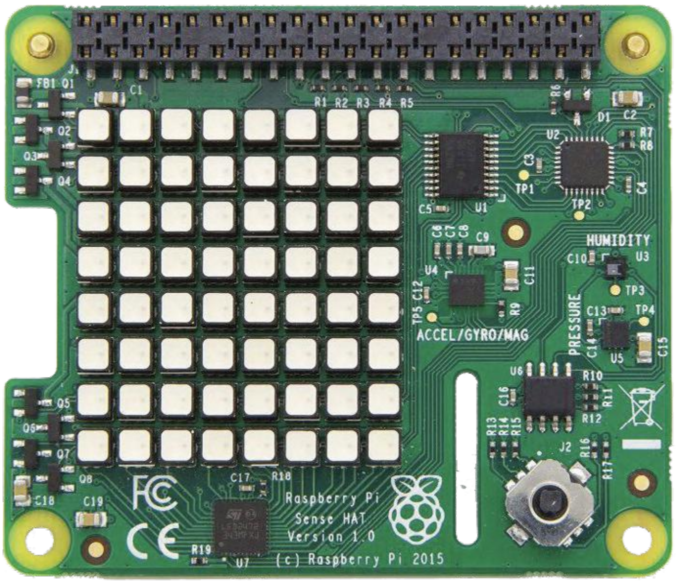


2


3


4

# Simple On-Off Temperature Controller with Raspberry Pi HAT



By **Dogan Ibrahim** (United Kingdom)

The Raspberry Pi Sense HAT board is normally plugged in to the 40-way connector of the Raspberry Pi. In order to interface external components to the Raspberry Pi in addition to the Sense HAT board, we need to connect the Sense HAT to the Raspberry Pi using either a ribbon cable or jumper wires so that other pins of the Raspberry Pi can be accessed. Therefore, we need to know which pins of the Sense HAT board are used by Raspberry Pi, and which pins of Raspberry Pi are free.

The Sense HAT board comprises seven main components and an LED matrix. The components on the board are controlled via the I²C bus interface. These main components on the board are:

| Component | I²C bus address | Function |
|---|---|---|
| HTS221 | 0x5F | Humidity sensor |
| LPS254H | 0x5C | Pressure/temperature sensor |
| LSM9DS1 | 0x1C,0x6A | Accelerometer+ magnetometer |
| SKRHABE010 | – | Joystick |
| LED2472G | 0x46 | LED matrix controller |
| LED matrix | – | – |
| ATTINY88 | – | Atmel microcontroller |

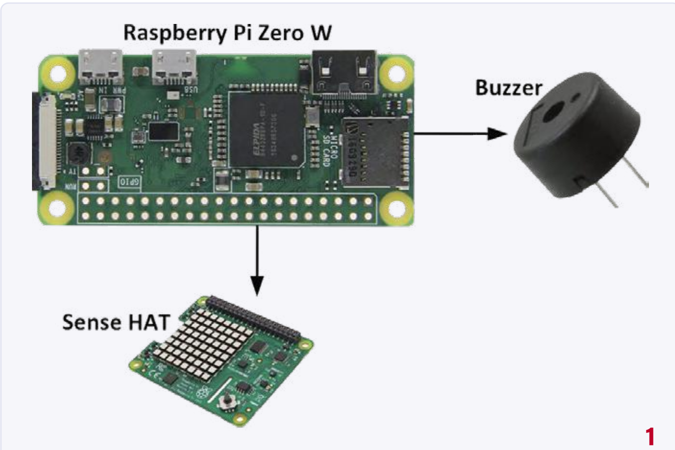
In addition to the I²C control lines, the ATTINY88 microcontroller on the board can be programmed via the SPI bus control lines (MOSI, MISO, SCK, CE0) provided on the board.

The Sense HAT board can be connected to the Raspberry Pi using only the following 9 pins of the 40-way connectors:

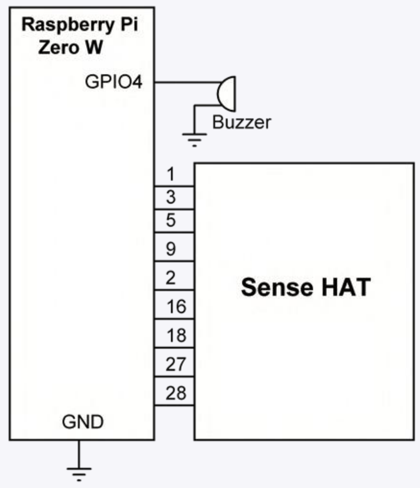
| Sense HAT pin | Raspberry Pi Pin | Function |
|---|---|---|
| 3 | 3 (GPIO2) | SDA (I²C) |
| 5 | 5 (GPIO3) | SCL (I²C) |
| 1 | 1 (+3.3V) | power |
| 9 | 9 (GND) | power ground |
| 2 | 2 (+5V) | power |
| 16 | 16 (GPIO23) | joystick |
| 18 | 18 (GPIO24) | joystick |
| 27 | 27 (ID_SD) | EEPROM |
| 28 | 28 (ID_SC) | EEPROM |

As you can easily figure out, that leaves a large number of Raspberry Pi pins free to use with external devices. Our get-u-going 'project' is an on-off temperature controller involving the The Sense HAT connected to Raspberry Pi Zero W to measure the ambient temperature. Additionally, a small buzzer is connected to one of the Raspberry Pi ports. The set temperature value is hardcoded in the program. If the ambient temperature is lower than the set temperature, the buzzer is activated and the LED matrix displays the ambient temperature in red. If on the other hand the ambient temperature is higher than the set temperature value, then the buzzer is deactivated and the ambient temperature is displayed in blue. The buzzer in this project can easily be replaced with a relay which can be connected to control a heater. The heater will then turn on if the ambient temperature is lower than the set value. **Figure 1** shows the functional diagram of the project, **Figure 2**, the 'circuit diagram.' The buzzer is connected to port pin GPIO4 of the Raspberry Pi. Both the buzzer and the Sense HAT board are connected to the Raspberry Pi using jumper wires.

The code to use for the control program *tempcont.py* is given in **Listing 1**. It's also a free download at [1]. At the beginning of the program the required modules get imported. The buzzer is assigned to number 4 which will correspond to GPIO4. The set temperature value is stored in variable SetTemperature and is hardcoded as 24



1



2

in his example. The buzzer is turned OFF at the beginning of the program. The remainder of the program runs in an endless loop. Inside this loop the ambient temperature is read from the Sense HAT and this temperature is compared with the set point value. If the ambient temperature is less than the set value, then the buzzer is turned On and the ambient temperature is displayed in red as non-scrolling. If on the other hand, the ambient temperature exceeds the set value, the buzzer is turned Off and the ambient temperature is displayed in blue.

The buzzer used in this project can easily be replaced with a relay, and a heater to the relay contact(s). This must be done in an electrically safe and responsible way as it may involve AC line voltage. The room temperature will then be controlled by the program.

200191-01

Source: Book: *Raspberry Pi Sense Hat*, By Dogan Ibrahim. To be released shortly.

**Listing 1: tempcontr.py**

```
#-----------------------------------------------------------------
#   ON-OFF TEMPERATURE CONTROLLER
#   -----------------------------
#
# This is an ON-OFF temperature control project. In this project
# a buzzer is connected to port pin GPIO4 of the Raspberry Pi in
# addition to the Sense HAT. The Sense HAT is connected using
# jumper wires. The buzzer is turned ON if the ambient temperature
# is below the setpoint temperature. At the same time, the ambient
# temperature is displayed in red colour. If on the other hand the
# ambient temperature is higher than the setpoint value then the
# buzzer is turned OFF and the display is in blue colour.
#
# The buzzer in this program can be replaced with a relay for
# example to control a heater
#
# Author: Dogan Ibrahim
# Date  : March 2020
# File  : tempcont.py
#-----------------------------------------------------------------
from display import Disp          # import Disp
from sense_hat import SenseHat    # import Sense HAT
sense=SenseHat()
import time                       # import time
import RPi.GPIO as GPIO           # import GPIO

GPIO.setwarnings(False)           # disable warnings
GPIO.setmode(GPIO.BCM)            # set GPIO mode

Buzzer = 4                        # Buzzer at GPIO4
SetTemperature = 24               # setpoint temp
red = (255, 0 ,0)                 # red colour
blue = (0, 0, 255)                # blue colour

GPIO.setup(Buzzer, GPIO.OUT)      # Buzzer is output
GPIO.output(Buzzer, 0)            # Buzzer OFF

while True:
   T = int(sense.get_temperature_from_humidity())
                                  # get temperature
   if(T < SetTemperature):        # T < setpoint?
      Disp(T, red, 0)             # display in red
      GPIO.output(Buzzer, 1)      # Buzzer ON
   else:
      Disp(T, blue, 0)            # display in blue
      GPIO.output(Buzzer, 0)      # Buzzer OFF

   time.sleep(5)                  # wait 5 secs
```



## WEB LINK


[1] tempcontrol.py download: www.elektormagazine.com/200191-01

# How to Take (Successful) Photos of Electronics

## Showing your projects in their best light

By **Mariline Thiebaut-Brodier, Jean-Paul Brodier †** (France), and **Elektor Graphics Department.**
Photo credits: **Scheimpflug**

Whether they're writing one article or a whole book, the authors of projects published by Elektor are invariably required to photograph their boards and components. Here are a few tips for providing good photos of your projects or preparing screenshots.

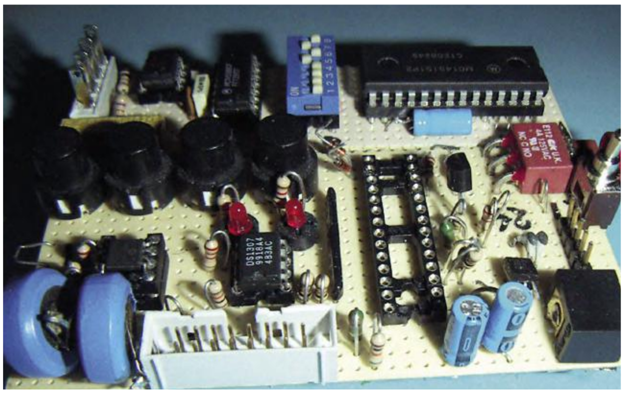


*Photo 1: BEFORE taking our advice.*

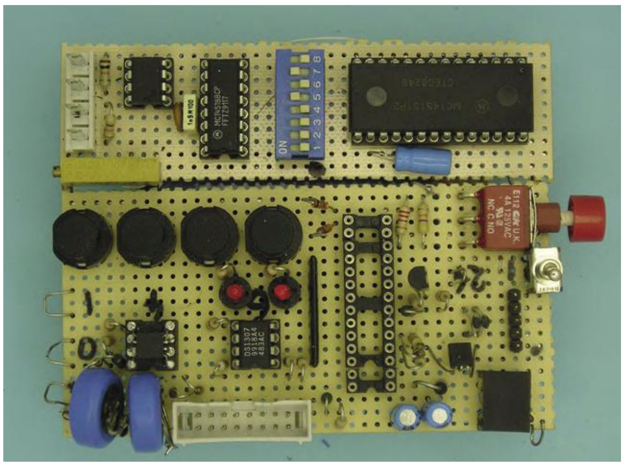


*Photo 2: AFTER taking our advice. ;-)*

A good photo of a component or a populated circuit board must be sharp and have good contrast; the lighting must show the relief, but without creating deep shadows. **Photo 1** shows what you get using an amateur camera with its built-in flash. The shortcomings are obvious: the lighting is inadequate at the rear, the sharpness no better, and the front of the board appears curved... The following paragraphs contain vital tips that will enable you to produce results like **Photo 2**.

### LIGHTING

The most blatant lighting errors are easy to put right. Never use your camera's built-in flash. This always creates 'hard' shadows. If you use several light sources producing a softer light, less of a point source and so more diffuse (desk lamp, small spots), you have better control over the incident light and over the shadows too.

**Photo 3** shows one simple solution for obtaining even lighting: the fluorescent tube in a bench magnifier. The magnifier itself is removed so the camera can be pointed through the hole in the support. Even though the magnifier would offer a degree of enlargement, it also introduces its own optical distortions that we can do without.

To avoid strong shadows around the board, all you need do is to separate it further from its background (here, the sheet of green paper). Any small, flat object will do — here, we've used an eraser that just happened to be handy. Other everyday supports can be used, for example children's' modelling clay. It's ideal, in fact, as it's easy to adapt the quantity and shape.

And lastly, to control the distribution of the light on the object being photographed, we add a reflector — here, two DIN-A3 sheets of paper (297 x 420 mm) stapled together and formed into a drum using an office clip (**Photo 4**). If you don't have two sheets of A3, you can staple four sheets of DIN-A4 (210 x 297 mm) together.

This continuous light source leads to longer exposure times, as it is less powerful than a flash: so it's a good idea to fix the camera on a tripod or to find some other solution to keep it perfectly still. What's more, using a tripod lets you frame your shots more carefully (hardly possible hand-held) and keep the framing the same over successive shots.

The camera can be mounted on a boom head using a swivel joint that allows you to shoot vertically (**Photo 5**). In the absence of this professional equipment, you can improvise — something like the bracket using perforated metal angle in Photo 3. Depending on the equipment you have and how much you like mechanical work, brazing, etc., you can construct set-ups that are more or less sophisticated and more or less permanent. The drill case and clamp are a quick, temporary lash-up solution, as is the sheet of fibreboard (drilling board) used to prop the case up. Mounting the camera onto the bracket presents a risk of scratching the bottom of the body. You can avoid this by gluing a simple piece of card onto the metal angle (**Photo 6**). The tripod thread is 0.25» diameter [6.35 mm] with 20 tpi (turns per inch; UNC). Rather than cutting a screw to the length required, it's more practical to use a wingnut as a locknut.

In case you do not have a 0.25» UNC wingnut, you'll have to make one yourself! A normal 0.25» UNC nut, an M10 washer cut into with tin snips or a junior hacksaw, which will hold in place while being brazed with the blowlamp, and two cuts with a grinder to finish off (Photo 7).

And lastly, don't forget to set your camera's white balance to 'artificial light' tungsten, otherwise your photos will have an unpleasant yellow cast.

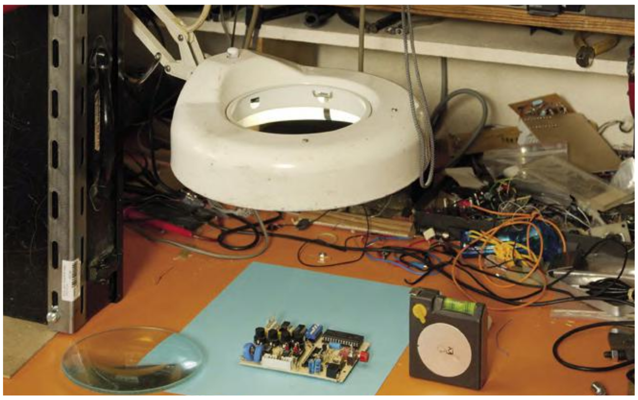
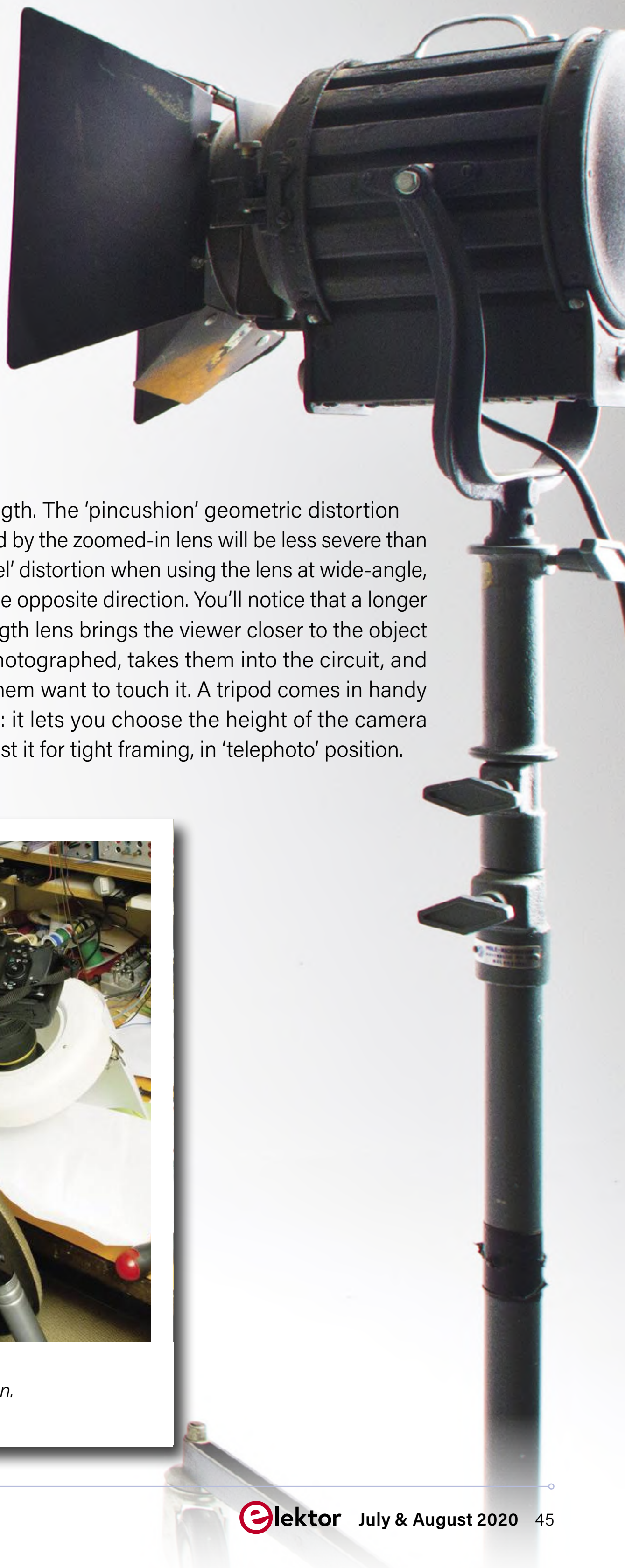


*Photo 3: The magnifier in this lamp has been removed to allow the camera lens to shoot through.*



### FOCAL LENGTH AND APERTURE

In Photo 1 (the bad example), the breadboard appears slightly curved. You need to try as far as possible to avoid this sort of geometric distortion, by adjusting the focal length of the lens used. To do this, press the zoom control towards 'telephoto' (zoom in) to use the longest focal length. The 'pincushion' geometric distortion produced by the zoomed-in lens will be less severe than the 'barrel' distortion when using the lens at wide-angle, and in the opposite direction. You'll notice that a longer focal length lens brings the viewer closer to the object being photographed, takes them into the circuit, and makes them want to touch it. A tripod comes in handy here too: it lets you choose the height of the camera and adjust it for tight framing, in 'telephoto' position.

*Photo 4: A few sheets of paper to spread the light around better.*

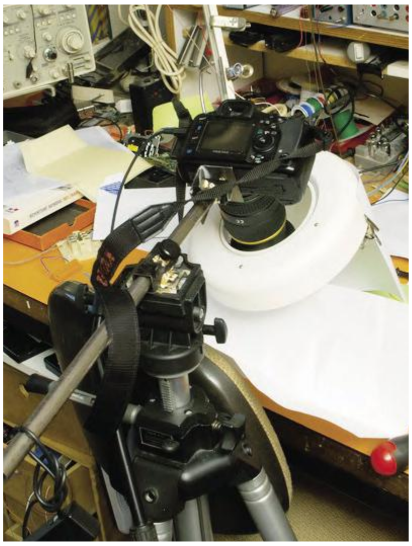


*Photo 5: Boom arm in action.*


*Photo 6: Don't forget the (self-adhesive) card to avoid scratching.*

*Photo 7: Home-made wingnut.*

*Photo 8: To avoid the arm's collapsing, without needing to tighten the M6 nut excessively or requiring a complicated mechanical assembly to build and adjust each time it is moved, make two saw-cuts in the metal angle and bend up a little tab.*

As far as distortion is concerned, we have now gone from pronounced barrel to slight pincushion. That's just the way it is with zoom lenses, you're always between The Devil and The Deep Blue Sea. We're choosing the lesser of the two evils: a long focal length and slight pincushion distortion.

The sharpness to the rear is still not good. It is possible to obtain sharpness all the way from front to back, or something pretty close, but using, at best, professional equipment, or at least manual aperture control. For overall sharpness, stop down (*f*/5.6 to *f*/8) to achieve greater depth of field. However, if you want only one detail of the object to be sharp, with the rest remaining out of focus, you can open the aperture (e.g. *f*/2.8).

### COMPOSITION

Frame your shot in such a way that the whole of the object being photographed is visible — or at least, the most important part of it. Make sure you leave enough 'white space' around it, both at the sides and top and bottom.

## SIZE OF DIGITAL PHOTOS

You often ask us how many megapixels are needed to allow proper reproduction of a digital photo. The size at which a photo can be reproduced depends on the quality of the camera used.

The larger the size at which the photo is to be reproduced, the greater the number of megapixels required. The table below gives a certain number of examples to illustrate the relationship between the number of megapixels a camera has and the maximum reproduction size (in this instance, referred to the 300 dpi (dots per inch) needed for reproduction in a magazine like Elektor).

| Camera resolution (pixels) | Maximum reproduction size @ 300 dpi resolution |
|---|---|
| 640 × 480 | 5.42 cm × 4.06 cm |
| 800 × 600 | 6.77 cm × 5.08 cm |
| 1024 × 768 | 8.67 cm × 6.5 cm |
| 1280 × 960(1,3 megapixel) | 10.84 cm × 8.13 cm |
| 1600 × 1200 (2,1 megapixel) | 13.55 cm × 10.16 cm |
| 1800 × 1200 (2,3 megapixel) | 15.24 cm × 10.16 cm |
| 2048 × 1536 (3 megapixel) | 17.34 cm × 13 cm |
| 2400 × 1600 (4 mégapixel) | 20.32 cm × 13.55 cm |

## SCREENSHOTS

Here are some tips for screenshots:

- Make sure the window to be reproduced is as large as possible
- Take a screen grab (under Windows: Print Screen for the whole screen and Alt + Print Screen for the active window; alternatively, use Win 10 Snip & Sketch)
- Paste the screen grab onto a blank canvas in an image processing program (e.g. Paint) and save the image in .BMP, .PNG or .TIF format, giving it an identifiable name. Do not use the jpeg (.jpg) format (too many compression artefacts).
- Do not modify the colour settings
- Do not change the aspect ratio
- Do not touch the resolution
- Do not do any cropping on the original image

**In other words, don't process the image in any way, simply send it to us in its original ('raw') state.**

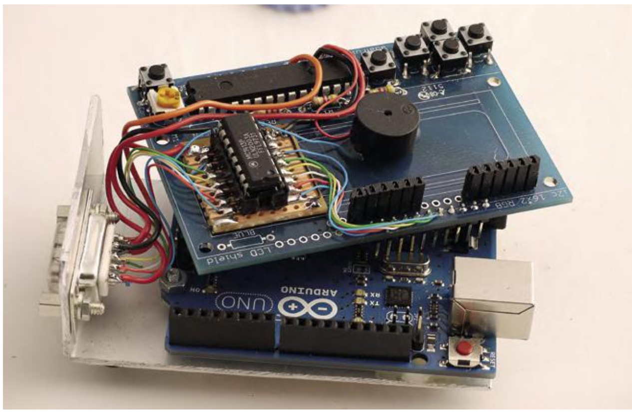


Photo 9: The plan view has given way to a perspective view to reveal the way the boards are stacked.


Photo 10: Comparing the size of two transistors. Tip: to avoid reflections from metal surfaces, you can grease them lightly.

## SENDING FILES

Many authors submit a text file (MsWord, OpenOffice) with all the illustrations already incorporated. This sort of document gives an overall idea of the content. However, for the page layout in Elektor magazine, you must provide separate files for the text and for all the supporting material (photos, PDF, software, etc.). The best thing is to zip up all the supporting elements together, identifying them carefully (don't call them just "circuits" or "PCB", but number them properly).

If the article has already been assigned a number, use this number when naming all the documents relating to it, adding appropriate indications to identify the contents. Don't forget to clearly mark the version. If no number has been assigned yet, identify the project using an abbreviated name and add your initials.

For photos, never try to process them yourself! Send them as they come out of your camera, without any retouching at all. If the files are very (too) big, don't compress them – for example, by saving as JPG. This always entails a certain loss of quality! In this situation, it is preferable to upload the photos in a different way. To do so, please contact your appointed editor.

For purely technical illustrations, photograph the object 'straight on' with no perspective (a plan view, as in Photo 2).
In other cases, it may be useful and more pleasing to shoot the subject from some particular angle (**Photo 9**).
And lastly, it's sometimes helpful to give an idea of the scale of the object photographed. You can do this by including an everyday object (matchstick, ruler, coin, etc.) in the composition for comparison (Photo 10).

### BACKGROUND

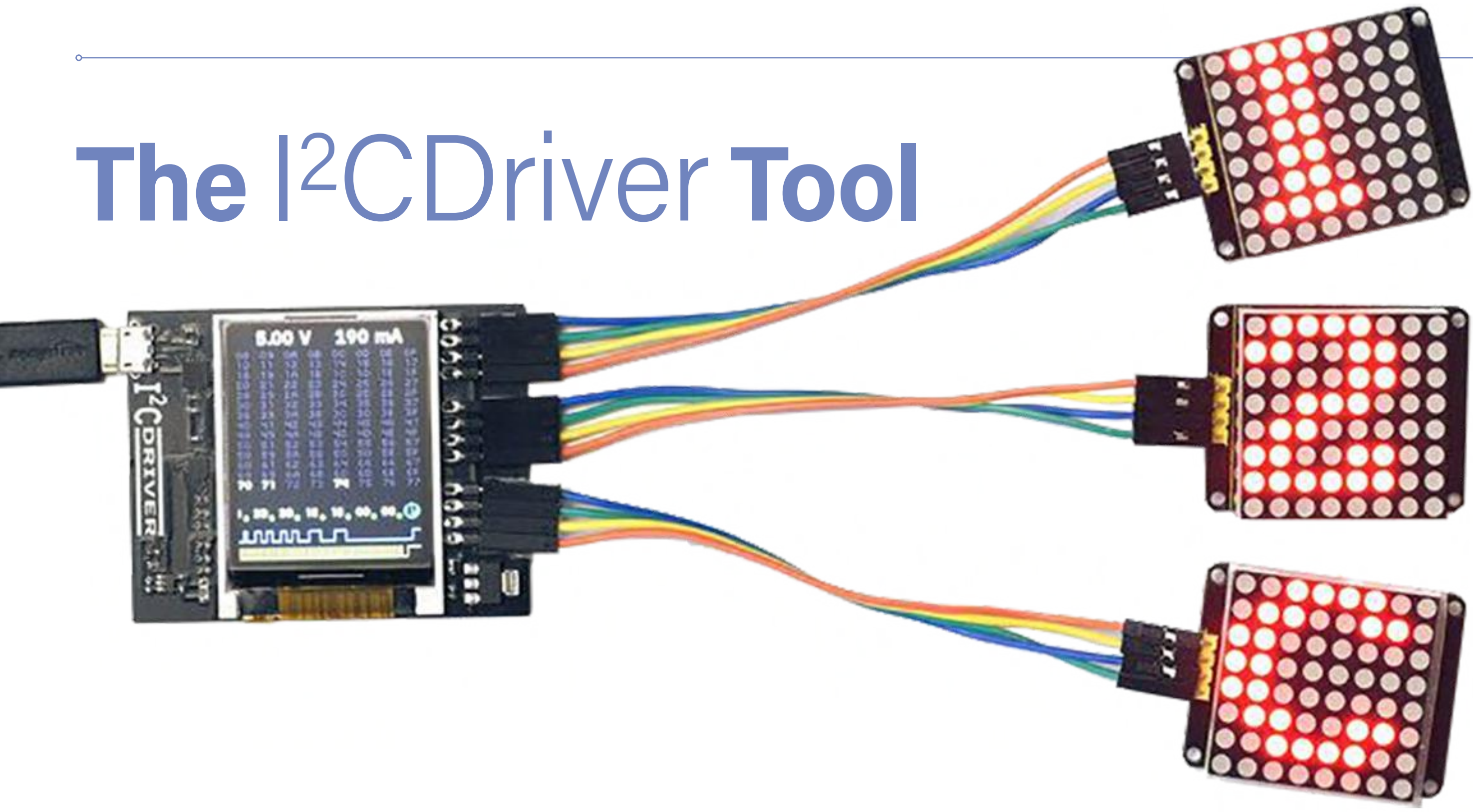
The object should always be shot against a totally neutral background of a suitable colour that contrasts with the object itself. Use uniformly light or dark backgrounds (e.g. a sheet of paper), with no visible structure, no perspective lines or untidy compositions that will distract the eye from the main subject and make it harder to incorporate the image into its context. Attention: if you use your camera's auto-exposure facility, avoid setting the objects against a white background — this will confuse the automatic system, which will be trying to compensate for what it assumes is a back-light situation. Never shoot your object against a pretty tablecloth, carpet, or floor covering (wooden floor, lino), to mention just a few examples of 'banned' backgrounds.
Since the object itself is the only thing we are interested in, you must make sure that it is the only thing in the photo (don't show it surrounded by half the equipment on your workbench or some of your home furnishings!).

### CONCLUSION

Photos are vital in making readers want to find out more about a project. Using a little imagination, some elbow-grease, and salvaged materials, you can really improve the quality of your photos of your projects. Do share your hints and tips for taking photos of electronics with other *Elektor Magazine* readers — send them to editor@elektor.com.

200186-02 | 160463-EN

# The I²CDriver Tool




By **Tam Hanna**

The I²C bus standard is used in a wide variety of embedded applications. For testing and development purposes, Excamera Labs has developed the I²CDriver. This is an open source tool for controlling I²C devices over USB. It runs on Windows, Mac and Linux systems and has a built-in colour screen that shows a live 'dashboard' of all I²C activity on the bus.

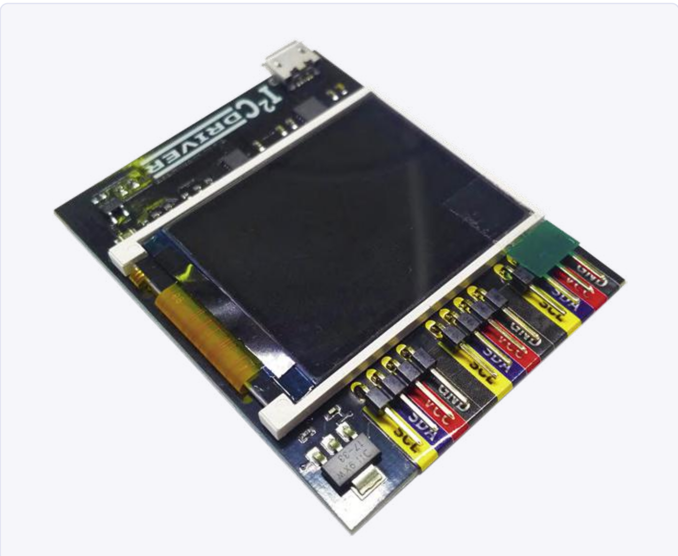



*Figure 1: The I²CDriver board is quite compact.*

After unboxing the contents of this kit can be seen in **Figure 1** — included is a circuit board with a Micro USB port and three groups of pin headers to hook up three I²C peripherals. On the underside four rubber feet ensure that the board is raised from the bench surface and does not slide around too freely.
For the I²CDriver Core Kit, Excamera Labs includes three 100-mm long cables terminated in Dupont-type female headers to provide the I²C signals to the peripheral devices. The voltage regulator on board the I²CDriver can deliver a total of 470 mA maximum output current at 3.3 V to power three I²C peripherals and signals are 5V tolerant. The board features an 8-bit microcontroller from the Silicon Labs EFM8 Laser Bee family of devices. The waveforms shown in **Figures 2**, **3, and 4** indicate the ripple level you can expect on this 3.3 V supply rail at various levels of peripheral loading. There is DC continuity between the PC USB port and peripheral devices, i.e. no galvanic isolation.

## Hook up some hardware

The author has recently been working on the 'HygroSage' humidity sensor which uses an I²C bus for communication with a processor. This seemed like a good opportunity to simply insert the I²CDriver between the PC and the sensor (see **Figure 5**). This sensor's has relatively low power requirements so we can power it directly from the PC via the I²CDriver board. The HygroSage unit started without any problems and its current consumption is shown along the top line of the I²CDriver screen. The I²CDriver display shows a grid of all the nodes but communication activity with any of the nodes is only shown once the software is installed and running on the PC. This software is described in more detail in the next section, it was noted that the software sometimes did not boot up correctly. In the case of a successful start, the display showing a grid of all the node numbers and the volume of traffic passing through each node indicated by the colour of the node. This forms a sort of 'heatmap' **Figure 6** of all active network nodes. In any I²C network with multiple devices, you will be able to see at a glance which ones are the most active.

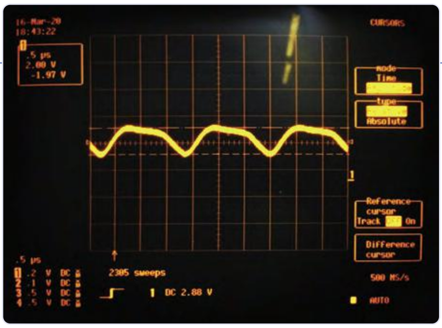


Figure 2: Power supply ripple with I²C peripherals drawing 500 mA...

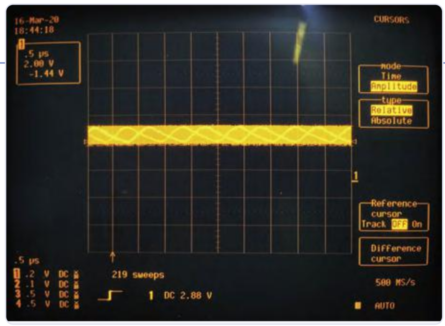


Figure 3: ... now 200 mA ...

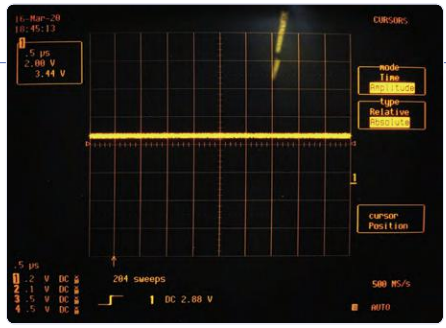


Figure 4: ... and now with no load.

The colour display has quite a narrow viewing angle so that the screen information such as differences in the 'heatmap' shading are barely visible from certain angles.

## Installation

Under [1] you will find the *i2cdriver-installer.exe* file under the *Resources* tab which will install the Windows software.
After the download, you have to click on it on the right and mark it in the settings dialogue as coming from the local computer before the operating system allows the installer to be processed. Developers working with Linux or Mac OS can also find the necessary setup instructions on the above website.
After the installation is finished in Windows we open the folder *C:\Program Files (x86)\Excamera Labs\I2CDriver*, where there will be both a command-line and a GUI version of the product
If you start the software with a connected I²CDriver and select *Monitor Mode* you will be able to see the last communication transaction as shown in **Figure 7**.
In practice, however, the use of the analysis function is limited, because in almost all cases more than one packet of information is passed at any one time. In this case, select *Capture Mode* instead. The button then appears activated, as shown in **Figure 8.**
The Capture function was tested by invoking a system error and then viewing the captured communication exchanges. Events leading up to an intermittent system failure can thereby be captured and examined in much the same way that hardware signals leading up to such an event can be recorded on a digital storage scope.

## Hardware interaction

Some time ago the author was required to implement a relatively complex algorithm on a controller. The most convenient way turned out to be to initially get the procedure running on a PC and then port it to the controller.
A similar procedure can be used when working with more complex sensors in an embedded environment. In the command prompt, *i2ccl* is a program that you can use to send commands to the I²CDriver using the following example:

```
C:\Program Files (x86)\Excamera Labs\I2CDriver>i2ccl.exe
Usage: i2ccl
```

Of particular interest here is the ability to write or read information in individual registers of connected devices. This not only helps when working with novel sensors, but can also be used to read out information during (automated) test runs.
If you don't want to program in the shell, you can rely on a Python API. The manufacturer demonstrates the use of a group of ready-made sample drivers - the following snippet can be used to collect information from an LM75B digital temperature sensor:

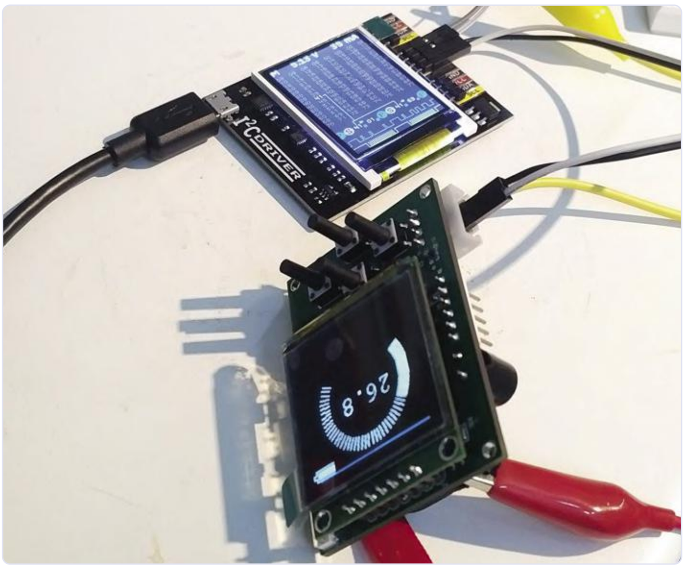


Figure 5: The I²CDriver connected to the author's sensor board.

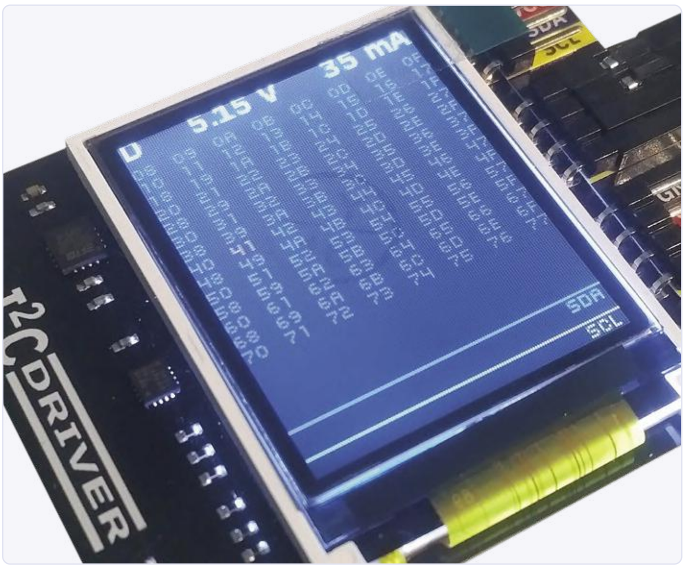



Figure 6: The highlighted address in the display indicates just one peripheral is connected here.

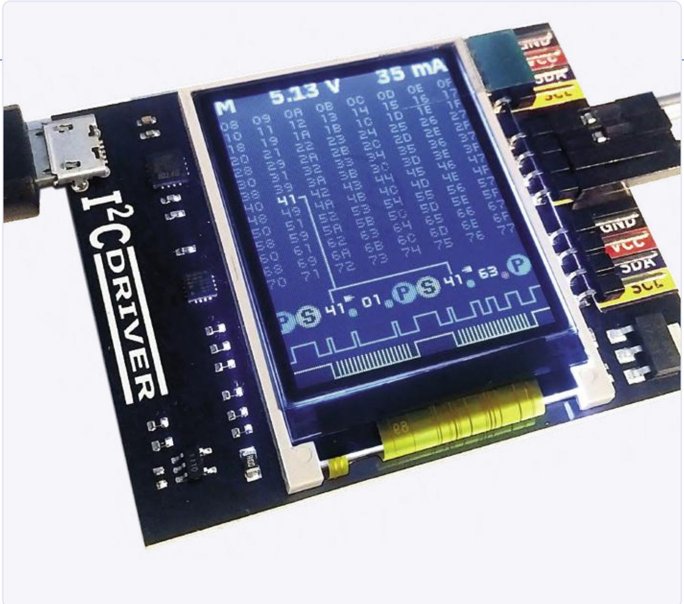



*Figure 7: Register operations appear on the I²CDriver display.*

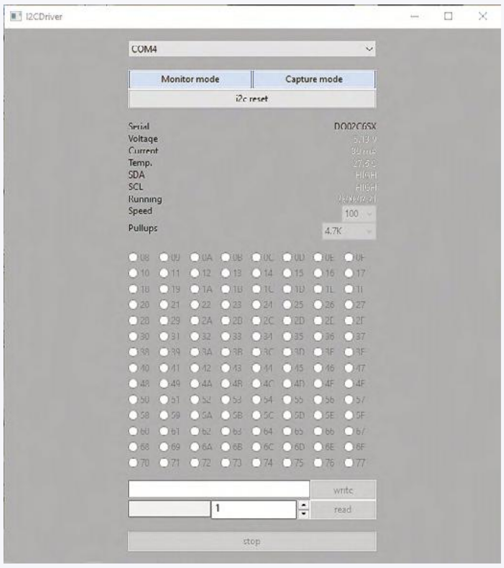



*Figure 8: The I²CDriver desktop software.*

```
    import i2cdriver
    i2c = i2cdriver.I2CDriver("/dev/ttyUSB0")
    d=i2cdriver.EDS.Temp(i2c)
    d.read()
17.875
    d.read()
18.0
```

The actual control API is simple and is available to view at GitHub [2]:

```
class LM75B:
    def __init__(self, i2, a = 0x48):
        self.i2 = i2
        self.a = a
```

Excamera Labs implements the hardware drivers using the Python OOP API. In this snippet `self` is a driver required by the language specification, while `i2` is an I²C- driver object. Finally, `a` defines the address at which the sensor can be accessed.
Information held in the register can then be read in using the following commands:

```
    def reg(self, r):

        return self.i2.regrd(self.a, r, ">h")

    def read(self):
        return (self.reg(0) >> 5) * 0.125
```

It should be noted that the Scan command used in *i2cdetect* is called from the Python command line and performs the scan function known from OrangePi and Co.
Finally, reference is made to the documentation available under [3]. It explains both the I²C-API and the physical communication protocol — if you are already familiar with the FTDI-API you can also address the I²CDriver directly.

## Conclusion

For anyone developing applications that rely on I²C bus protocols the I²CDriver tool can provide a whole raft of data to help identify and diagnose any system problems. Regardless of whether it is a quick analysis of the activity of an I²C network or the integration of a new sensor — the board provides valuable help.
The price is reasonable considering the time saved and convenience it brings to speed up hardware development. It has the ability to quickly capture and store information that would otherwise require the setup of a whole range of sophisticated test equipment. If it could also be used in stand-alone mode it would undoubtedly extend the usefulness of this device further.

200148-02



## LINKS & LITERATURE

[1] Software: https://i2cdriver.com/
[2] I²C driver API: https://github.com/jamesbowman/i2cdriver/blob/master/python/lm75b.py
[3] Documentation: https://i2cdriver.com/i2cdriver.pdf

# JOY-iT Joy-View
# Portable Touchscreen

By **Harry Baggen**

It's not easy to find a small, high-quality monitor that is conveniently portable – the options are fairly limited. The Joy-View 13 makes the choice a lot easier, and thanks to the presence of both HDMI and USB C ports, it can be connected to virtually all modern devices. We tested this small 13-inch monitor with several types of computers.

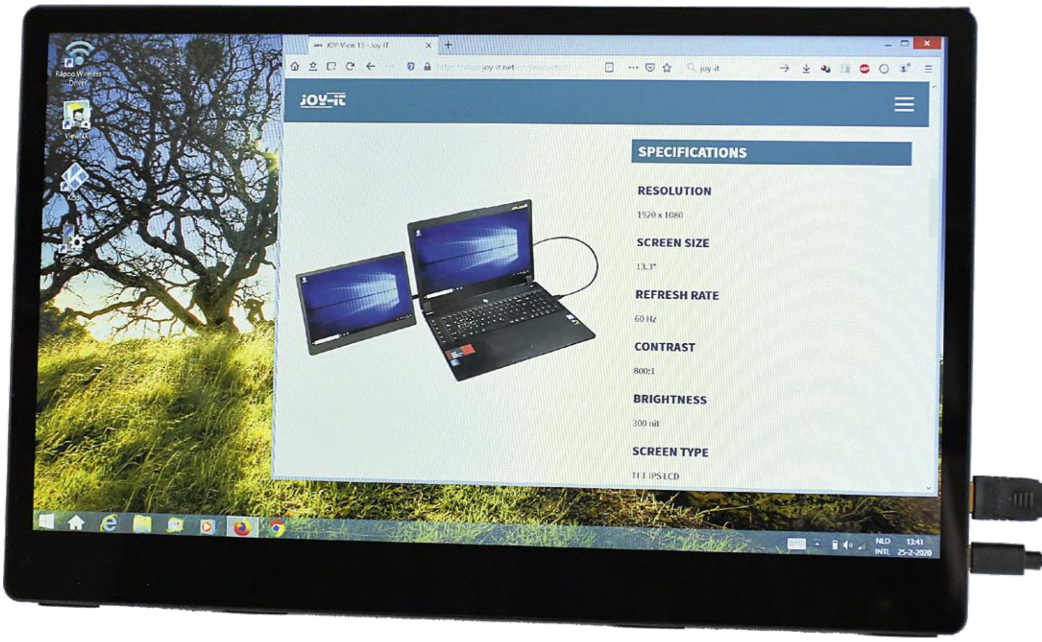



*Figure 1: The Joy-View 13 portable touchscreen.*

A second screen for your laptop can be very handy in some situations. Small monitors are not expensive and there are lots of them to choose from, but they are not designed to be portable. What you want is something that is compact and touch-sensitive, and which preferably can operate independently of AC mains power. These are precisely the features offered by the Joy-View monitor. (**Figure 1**). It is thin, sturdy, and can be powered from the AC mains, a laptop or a powerbank, and thanks to its USB-C port it can also be connected to the latest laptops and smartphones.


*Figure 2: The connectors on the right side (top) and left side (bottom) of the Joy-View 13.*

## Unpacking

When you unpack the monitor, it feels a lot like a large tablet computer. The screen housing is made of black anodised aluminium and is nicely finished. Most of the monitor is only 5 mm thick. The bottom part, where the connectors are located (and probably most of the electronics), is a bit thicker at 9 mm (**Figure 2**). The monitor comes with a matching plastic 'Smart Case' with magnets that hold it fairly firmly to both sides of the screen (**Figures 3** and **4**). The case also serves as a stand – you can attach it to the back of the monitor (also with magnets) and unfold the bottom part (**Figure 5**). To prevent the monitor from sliding over the support surface, there are four small but sticky feet underneath.

An on/off button, a toggle button for menu navigation, and a USB-C power connector are located on the left side of the thicker bottom portion of the monitor. On the right side there is a mini HDMI connector, a USB-C connector that can be used for both power and data, and a 3.5 mm jack socket for headphones. The base also has some small openings for the two built-in speakers. The monitor weighs a bit more than 1 kg, including the case. That's not especially lightweight, but the monitor does make a sturdy impression.

The Joy-View 13 comes with a large number of cables – an HDMI to mini HDMI cable, a USB-C to USB-C cable and a USB-A to USB-C cable – as well as an AC mains adapter and a cleaning cloth for the screen.

*Figure 3: The back of the housing. The part with the diamond pattern is the magnetic Smart Case.*

*Figure 4: The screen with the Smart Case folded out.*

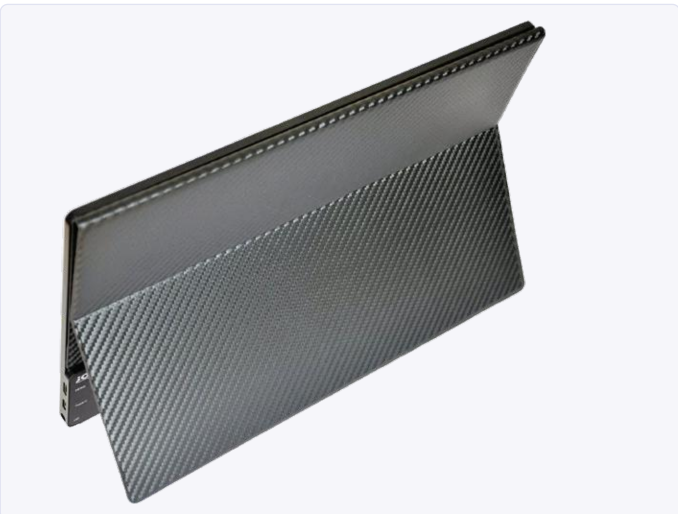


*Figure 5: The screen with the Smart Case folded out. 5. The Smart Case folded down at the back to act as a stand for the monitor.*



## Video

The actual screen has a diagonal dimension of 13.3 inches. The resolution is Full HD, which means 1920 × 1080 pixels. The front surface is fairly reflective, but that is usually unavoidable with a touchscreen. The viewing angle of the IPS screen is good in all directions. If you look at the screen from an angle, the image is a bit darker but the colours remain reasonably good. The image contrast is good and the maximum brightness is an impressive 300 Cd/m², comparable to my standard monitor and the screen of my laptop. You can even work in a brightly lit room or outdoors in the shadow, but I wouldn't try to use it in full sunlight. The colour rendition is good, although the colours look a bit too saturated to me, but that can be adjusted somewhat with the menu settings. The on-screen menu offers quite a few options, but using it is not especially easy. To activate the menu, you have to press the menu rocker button, which requires pressing the bump on the rocker button fairly deep. That's also true for selection of the menu items, with the result that I sometimes ended up with the wrong menu item. Fortunately, you don't need the menu very often. Once everything is configured the way you want, you only rarely need to use the menu.

## Connection options

With its two USB-C ports and a mini HDMI port, the Joy-View 13 can be connected to a large number of devices, from PCs to smartphones. I tried a variety of combinations. But before describing them, I would first like to mention the options for powering this portable monitor. This can be done through both USB-C ports. The port on the left side is only suitable for power, while the port on the right side is suitable for both power and data. Either the included AC mains adaptor or a powerbank can be used as the power source. One of the included cables can be used to connect the monitor to the power source. If the monitor is connected to a USB-3 port, it can provide communication with the screen as well as power, as you would expect.

There is also a consideration regarding the built-in touchscreen. In order to use it, the monitor needs to be connected to a modern smartphone or tablet through a USB-C port. The manufacturer provides a list of compatible devices, which can also be found on [1]. Another option is to connect the monitor to a PC, laptop or Raspberry Pi board through two cables: HDMI for video and USB for touchscreen control. This is possible with Windows 8 or 10, and with Raspbian on a Raspberry Pi.

First I connected the screen to my desktop PC running Windows 10. This worked perfectly with the HDMI cable, with power being supplied by the included AC mains adapter. In this case I did not connect the touchscreen, because it would be a bit confusing in a setup where the desktop is divided over two screens and only one of them is touch sensitive.

The next combination was the Joy-View 13 with my Windows 8 laptop. That worked very well. After connecting the HDMI cable and the USB-C cable to the USB-3 port of my laptop, I immediately had video with touchscreen functionality. This worked very nicely – the screen responded directly to touch input, and Windows 8 (just like Windows 10) is well prepared for touchscreen use.

The final combination was a Raspberry Pi 3B running a current version of Raspbian. Here as well, the RPi was connected to the Joy-View 13 through two cables. With this arrangement, the touchscreen also worked right away, with no need to adjust any settings in Raspbian. However, I had some problems with supplying power

to the screen. With a reduced resolution setting, the RPi was just able to supply enough current to the monitor through the USB port, but with full resolution things went wrong and I had to connect the AC mains adapter. Aside from that, everything worked well.
I also tried to connect the screen to a fairly recent tablet with a USB-C connector, but unfortunately this did not work. However, the laptop model was not in the Joy-iT list of compatible devices.

## Conclusion

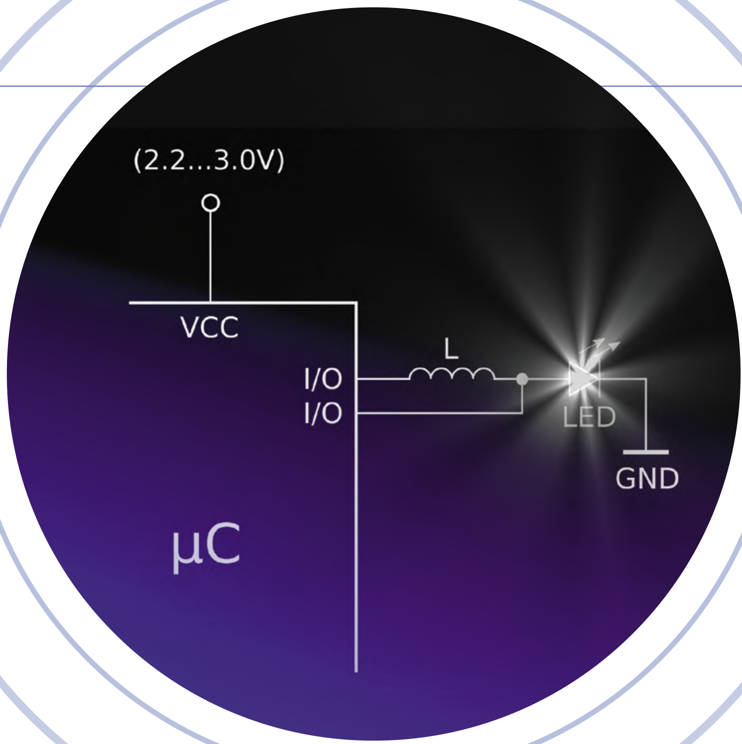
The Joy-View 13 portable touchscreen monitor is a nicely finished unit that provides excellent video and can be used in many applications and with many different devices thanks to the built-in capacitive touchscreen, the versatile connection options and the variety of power supply options. If you are looking for a monitor that is compact and easily portable, the Joy-View 13 is an excellent choice. And if you would like to have something a bit larger, you can also opt for the 15.6-inch version of this monitor, the Joy-View 15.

200169-04

*Figure 6: Here the Joy-View 13 is connected to a Raspberry Pi 3B running Raspbian.*

### WEB LINKS

[1] **List of compatible devices**: http://joy-it.net/files/files/Produkte/JT-View13/JT-View13-Ger%C3%A4te.pdf




# LED Booster for Microcontrollers

## With just one component

By **Martin Heine**

There's many a time when you want to connect a white LED to a microcontroller operating from a 3 V supply voltage. Unfortunately, this doesn't work and your nice white LED only lights up feebly or not at all.

Why does it work perfectly with red and green LEDs, but not with white? A bit of data sheet research reveals the reason: white LEDs have a forward voltage of 3.2 V, so a 3 V supply is simply not enough to let them light up properly. The advice you often see in online

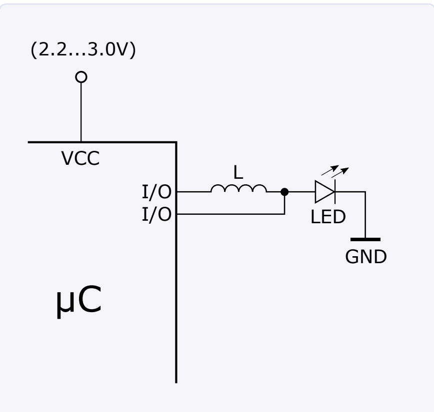



*Figure 1: Circuit diagram of the LED booster.*

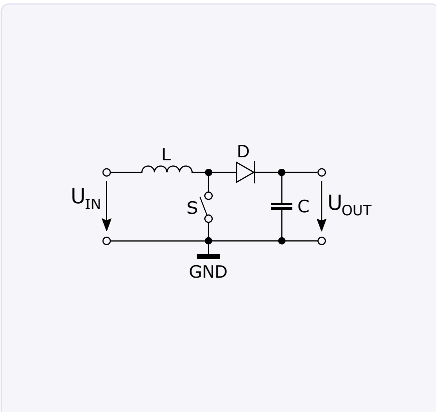



*Figure 2: Basic circuit of a boost converter.*

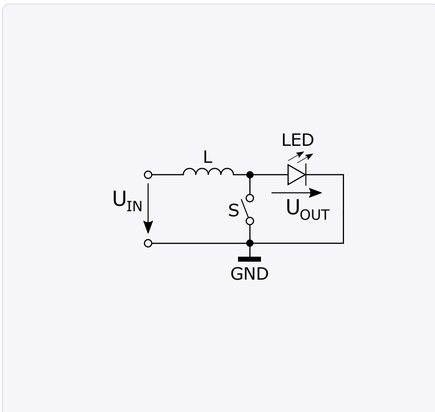



*Figure 3: Diode and capacitor replaced by a LED.*

forums is to use a boost converter to generate a higher voltage, along with a transistor switch to control the LED. For just a single LED, this seems like a lot of overhead.

The good news is that there's an easier way. And it only needs one inexpensive component: an inductor, which costs next to nothing. If you wire it up right and drive it the right way, your white LED will light up nicely — and this even works with a microcontroller supply voltage as low as 2.5 V. Magic? Not at all.
The circuit shown in **Figure 1** is a boost converter. It also goes by other names, such as step-up converter. But how does this minimalistic boost converter work?

## Operating principle of a boost converter

Take an inductor L (a coil) and connect one end to the input voltage $U_{IN}$ and the other end to a switch *S* tied to ground. When the switch S is closed, a gradually rising current flows through the inductor L, creating a magnetic field. When the switch S is opened a bit later the magnetic field collapses, generating an inductive voltage over the coil (**Figure 2**), the same as an ignition coil in a car.
This voltage adds to the supply voltage, so the voltage $U_{OUT}$ at the diode D (which charges the capacitor C) is higher than the supply voltage. When the switch S is closed again, this process repeats. A LED is basically just a diode, so the diode D can be replaced by a LED (**Figure 3**). And the LED can be connected directly to ground.

Now you're probably wondering how this LED boost converter can be implemented using a microcontroller with only one additional component, in the form of an inductor. For this we take advantage of the different I/O pin modes. A microcontroller I/O pin can be set to push-pull mode or to open-drain mode. Push-pull means that the pin is switched to $V_{CC}$ for a logic 1 (high) or switched to GND for a logic 0 (low), as shown in **Figure 4**. With the open-drain setting, by contrast, the output is open for a logic 1 and switched to GND for a logic 0. That's exactly the same as the switch S in the circuit described above.

The value of the inductor depends on the switching frequency, the current, the input voltage and the output voltage. The following approximate formula for determining the inductor value can be found in most data sheets for boost switching regulators:

$$L = \frac{U_{IN} \bullet (U_{OUT} - U_{IN})}{(\Delta I_L \, f_s \times U_{OUT})}$$

*where L* = inductance [H], $U_{OUT}$ = output voltage [V], $U_{IN}$ = input voltage [V], $I_{OUT}$ = output current [A], $f_s$ = switching frequency [Hz], $\Delta I_L$ = inductor ripple current [A]

The inductor ripple current $\Delta I_L$ is the difference between the minimum and maximum current through the coil. In other words, it is the peak-to-peak waveform of the average (DC) current through the coil (**Figure 5**).
The current through the coil always differs from the output current by a few percent. This is why most data sheets from boost switching regulator manufacturers also provide the following approximate formula, which is fully sufficient for this LED booster circuit:

$$\Delta I_L = 0.2 \times I_{OUT}(\max) \times \left(\frac{U_{OUT}}{U_{IN}}\right)$$

The maximum output current $I_{OUT(max)}$ is the maximum allowable current through the LED.

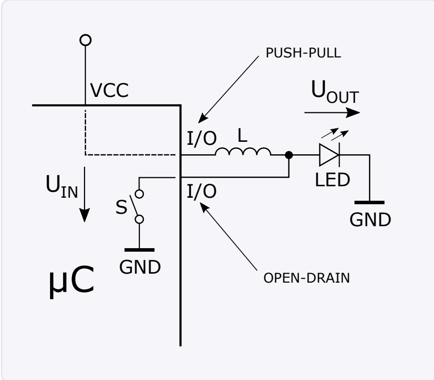



*Figure 4: Configuring the microcontroller pins.*

This is specified as 30 mA in the data sheet for the selected white LED. If we set the output current $I_{OUT(max)}$ to 30 mA, the output voltage $U_{OUT}$ to 4 V and the input voltage $U_{IN}$ to 2.5 V in the above formula, the resulting inductor ripple current $\Delta I_L$ is 9.6 mA. Using the previous formula, we can then calculate the inductance value *L* as 97.66 µH for a switching frequency $f_s$ of 1 MHz.

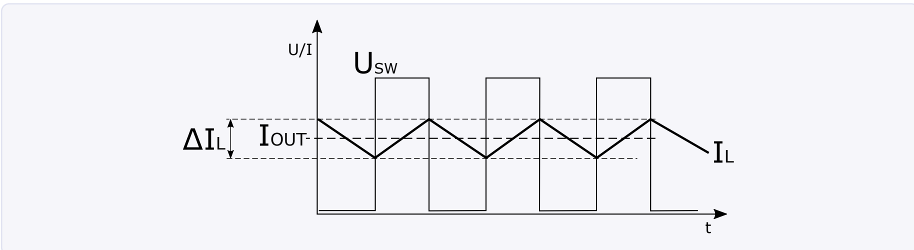



*Figure 5: Voltage and current waveforms.*

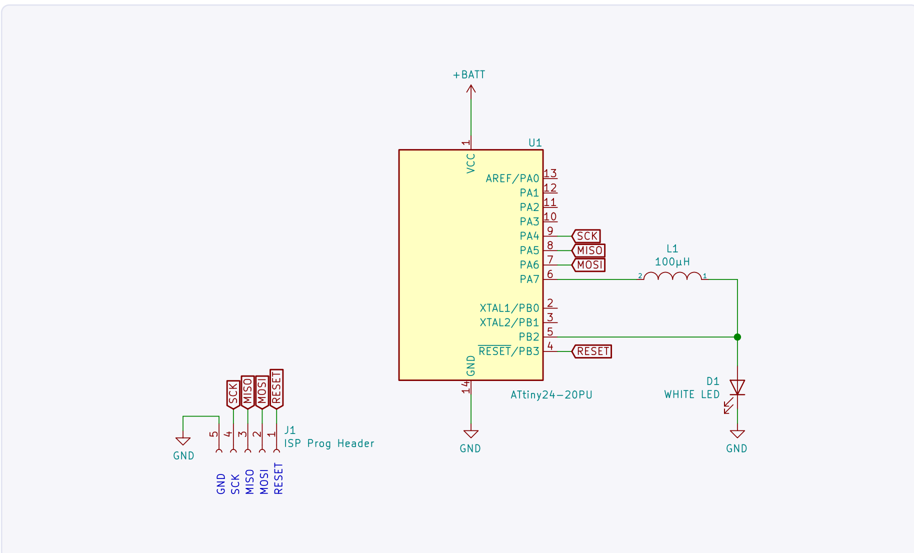



*Figure 6: Circuit of the LED booster test setup.*

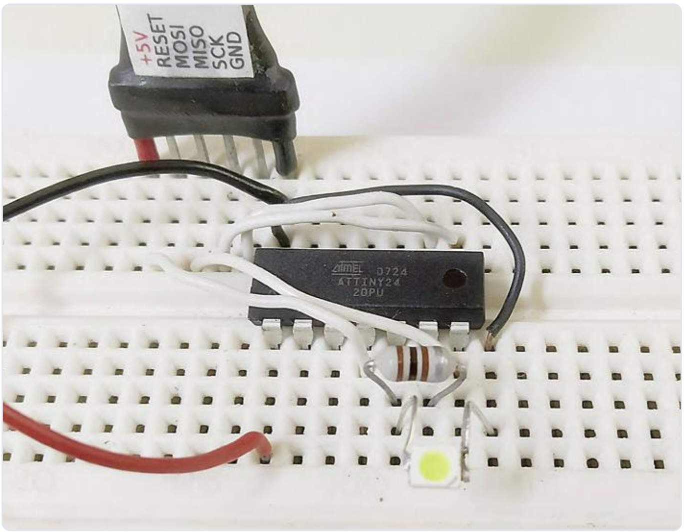



*Figure 7: LED booster test setup.*

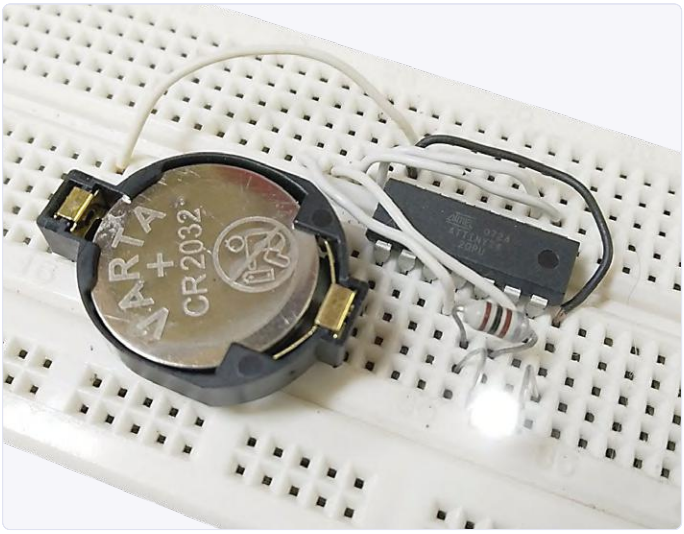



*Figure 8: A white LED blinker powered by a 3 V button cell.*

There's a virtually unlimited choice of 100 µH inductors, available in a variety of shapes and sizes. An inductor in an SMD 0805 package is suitable for a small white LED with a typical maximum current of 30 mA.

## Test setup

We took a white LED in a PLCC-4 package with a forward voltage of 3.2 V and a forward current of 30 mA, soldered two wires to it, and pugged them into a breadboard. Then we added an ATiny24 AVR microcontroller from our parts bin and a 100 µH inductor (**Figures 6** and **7**). We connected the Reset, MOSI, MISO, SCK and GND pins of the microcontroller to our programmer. Finally, we used an adjustable lab supply to power the circuit with a supply voltage of 2 V to 3 V.

## Programming the LED booster

Once all the required connections are in place, it's time to program the ATiny24. There are various ways to output square-wave signals from a microcontroller. One option is pulse width modulation (PWM), and another is toggling the I/O pins. For the sake of simplicity, we opted for the latter approach. For our boost converter we need a switch that toggles to ground, which means the output concerned must alternately be set to open-drain and pull-down. This is done directly in a loop, without any delays. With an ATiny24 microcontroller clocked at 8 MHz, this gives an output frequency of approximately 1 MHz.
To change the default internal 1 MHz clock frequency to 8 MHz, you have to disable the *CKDIV8* fuse (which is set by default). This can be done with the AVRdude program as follows:

```
avrdude -U lfuse:w:0xEA:m
```

If we now embed the booster driver loop (open-drain/pull-down loop) in a second loop that toggles the input voltage ($U_{IN}$) pin, we get a classic blinker with a 'boosted' white LED (**Figure 8**).

190361-03

**LED booster source code**

```
#define F_CPU 8000000UL              // 8 Mhz

 #include <avr/io.h>
 #include <util/delay.h>

 int main(void)
 {
   unsigned long int i;

   while (1)
   {
     PORTA |= (1<<PA7);
     DDRA |= (1<<DDA7);              // LED ON
     for(i=0; i<=400000; i++)        // Booster Loop
     {
       DDRB |= (1<<DDB2);            // Open Drain
       DDRB &= ~(1<<DDB2);           // Pull down
     }
     PORTA &= ~(1<<PA7);             // LED OFF
     _delay_ms(500);
   }
 }
```

# Experimental **Ultrasonic Clothes Washer**

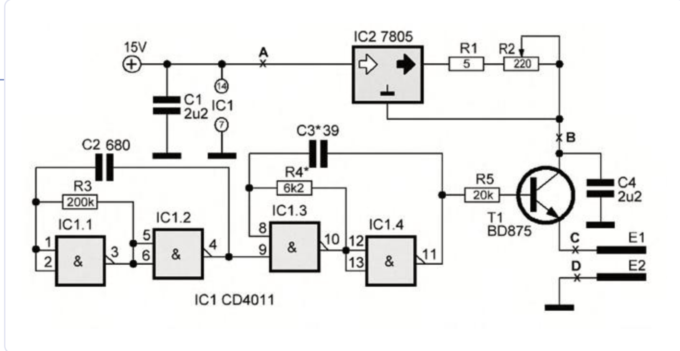



*Figure 1: Schematic of an experimental ultrasonic washing device.*

By **Andrey M. Shustov** (Russia)
and **Michael A. Shustov** (Germany)

The adoption of advanced energy-saving technologies has allowed an ultrasonic washing device to be developed for household use. Ultrasonic washing is conducted by compressional and rarefactional waves which are periodically formed in the volume of liquid and appear in a practically incompressible medium (water). Clothes and other objects that require cleaning placed in such a liquid are subjected to intensive hydro acoustic effects. Hydro-acoustic waves initiate the occurrence of ultramicroscopic gas bubbles which contribute to the separation of dirt microparticles from the clothes so 'washed', or in general, the object to be cleaned. The formation and subsequent collapse (destruction) of gas bubbles leads to the formation of ozone which is known to sterilize clothes.

The advantage of ultrasonic washing is that clothes cannot be deformed and torn; even woollen clothes and thin linen can be washed safely. In addition to the washing and disinfection of clothes, it is possible to treat vegetables and fruits for canning, as well as disinfect water.

The circuity for an experimental medium-power ultrasonic washing device as shown in **Figure 1** consists of a power supply source (IC2), two interconnected oscillators operating at a frequency of 10 kHz and 1 MHz (IC1), an output stage based on power transistor T1 and an ultrasonic emitter connected to points C and D.

The power supply source in the prototype is not regulated and designed for a maximum power consumption of about 3 watts, which should be enough to wash clothes in a liquid volume of 10 to 25 litres. It is useful to provide an ultrasonic washing device, even this experimental one, with a control to give smooth adjustment of output power. The schematic shows a constant-current source that is adjustable within 25 to 1000 mA and connected in the gap between points A and B. **Figure 2** shows an alternative for it: an adjustable constant-voltage source with a range of 5 V to about 13 V.

The pulse packet oscillator is constructed using a standard CMOS IC type CD4011 and has no special characteristics. To match the oscillator frequency to the resonant frequency of the ultrasonic emitter, it is necessary to tweak the values of the *R-C* elements in the high-frequency oscillator i.e. IC1.3/IC1.4. Both regulator IC2 and transistor T1 should be fitted with a heatsink.

Be sure to select an ultrasonic emitter ('activator') that can be waterproofed to achieve the maximum energy release of ultrasonic oscillations into the environment (liquid). Piezoelectric ceramics (barium titanate and strontium titanate), ferrite- or permalloy-core emitters, piezoelectric plates, etc. are usually used as ultrasonic emitters

The project is open to a wide field for experiments. For example, consider obtaining ultrasonic oscillations by passing pulses of electric current through water using a pair of closely spaced electrodes connected to points A and B of the device. Periodically passing current pulses between the electrodes will cause acoustic, electro stimulated modulation of a solution. Aluminum is recommended for use as the electrodes.

The circuit, even if experimental, must provide reliable isolation from the mains power. The washing container should be located at a distance away from grounded items, and installed on a dry floor.

The method of washing with an ultrasonic washing device are as follows:

- washing powder is put into a washing solution using the same dose as recommended for hand washing;
- the water temperature should be about 65 °C;
- clothes should float freely in the solution and should be occasionally stirred with a wooden forceps.

Very dirty areas of clothing are recommended to be additionally lathered. The washing process lasts 30 to 40 minutes or more depending on the power and efficiency of the ultrasonic activator.

Clothes can also be rinsed using an ultrasonic washing device. Optimal use of the device is achieved after several washings.

200237-01

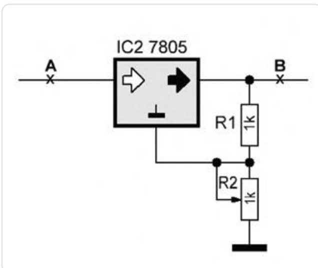



*Figure 2. Adjustable voltage source for the ultrasonic washing device.*

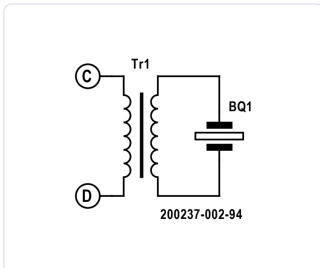
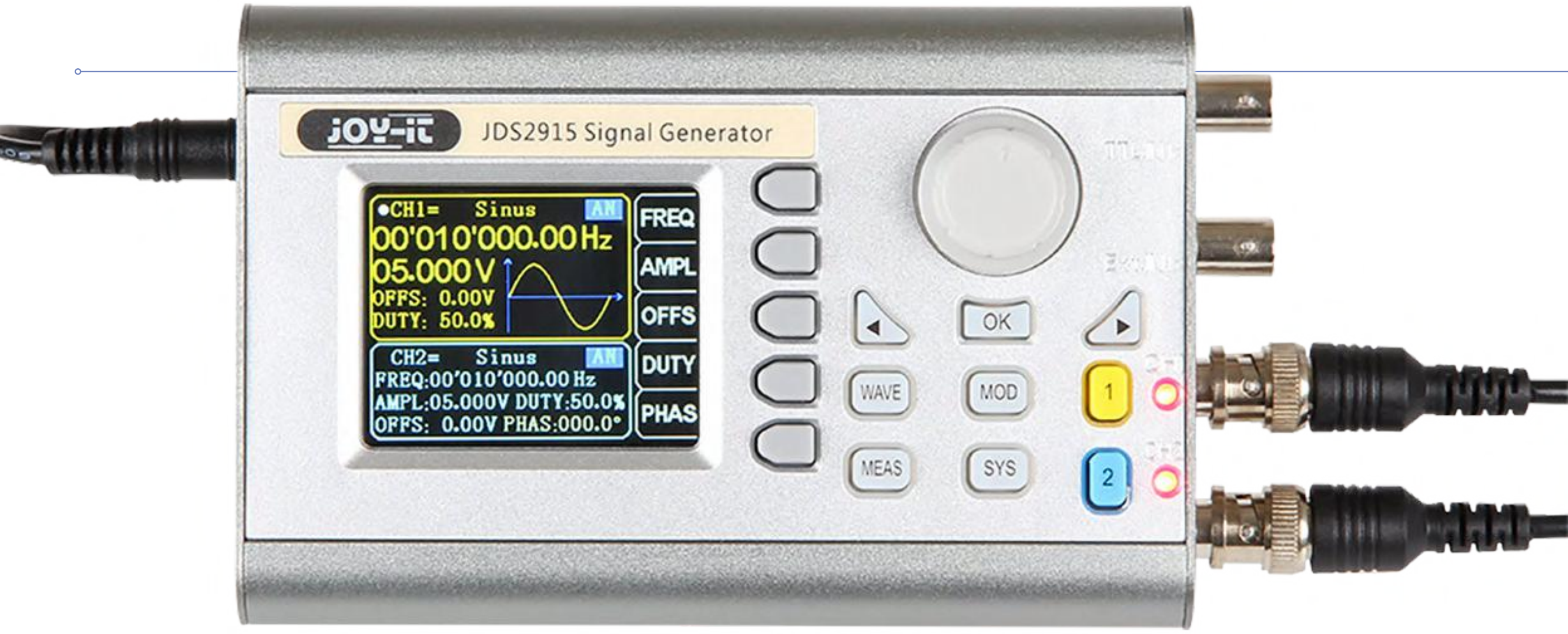



*Figure 3. Circuit of the ultrasonic emitter. Duly water-proofing the device and the connections is a must.*




# The **Joy-iT JDS2915** Signal Generator

## A 2-Channel Signal Generator and Frequency Counter

By **Dr. Thomas Scherer** (Germany)

Joy-iT have introduced the JDS2915 two-channel signal generator. This model can output various signal waveforms at frequencies up to 15 MHz and costs less than €100. It is a cheaper version of their JDS6600 signal generator which can output signals up to 60 MHz. For many applications this lower-cost 15 MHz version may be all you need.

Last spring my co-worker Harry Baggen reviewed the Joy-iT JDS6600 signal generator which is the big brother of the JDS2915. In his review he felt that the unit was let down by its cheap-looking plastic case. I'm sure he will be delighted that this smaller, cheaper, lower-frequency version from Joy-iT is housed in a more robust aluminium profile case. The model JDS6600 he reviewed [1] has the same features as the JDS2915 on review here (except for its different frequency bandwidth and different case) so I will refer to his review and try to avoid repetition of common features by concentrating to other aspects of the unit.

### Unboxing

**Figure 1** shows what comes in the box: In addition to the actual signal generator in a tough metal housing, there are two BNC cables with crocodile clips for the two signal outputs, a short cable with BNC plugs at both ends for connecting to other devices, and a USB 2.0 cable with a USB-A and USB plug for controlling the signal generator via a PC, a mains power adapter with a lead terminated in a barrel plug (5 V/2 A) and a thin DIN A5 sized manual. *A manual? Yes finally… a real manual!* I spoke too soon; the tri-lingual (German, French and English) information booklet consists of just two pages for each language. A more comprehensive user manual can be found on the Joy-iT website [2] (in German and English). You can also download a datasheet and the PC software for remotely controlling the signal generator (including some installation instructions).
The thin DIN A5 booklet only indicates (**Figure 2**) which buttons do what. The thing I am not so keen on is the layout of the input and output connectors on the right side panel of the aluminium case and not on the front panel like the JDS6600. From an operators point of view there is practically no difference between the two models. The 'Control buttons' are mounted slightly lower on the JDS2915 and the shaft encoder is directly above them. On the JDS6600, this knob has been moved to the right.

Even before trying it out, I had concerns that the unit is quite lightweight and would need to be held down whenever I wanted to operate a pushbutton on the front panel. Indeed this turned out to be the case; there are no rubber feet on the underside of the case to prevent sliding.

## Key Features of the JDS2915:

- Supply: Euro-style mains power supply rated 5 V / 2 A
- Two channel signal outputs with 50-Ω impedance
- Output signal ≤10 MHz: 0 to 20 $V_{PP}$, in 1 mV steps
- Output signal >10 MHz: 0 to 10 $V_{PP}$, in 1 mV steps
- Offset: -10 to + 10 V in 10 mV steps
- Output waveforms: sine, square, triangle, pulse, arbitrary etc.
- Frequency signals: 0 to 15 MHz, in 10 mHz steps
- Duty cycle (for pulse and triangular waveforms): 0.0 to 99.9%
- Special functions: sweep for sine and pulse waveforms
- Frequency counter: 0 to 100 MHz
- Frequency accuracy: ±22 ppm
- Frequency stability: ±1 ppm/3 h
- Amplitude stability: ±5%/5 h
- Digital signal resolution: 14 bit
- Signal sampling rate: 266 MS/s
- Dimensions: 145 x 95 x 55 mm (WxHxD)
- Weight: 450 g (without power supply)
- Current consumption at 5 V: max. 850 mA (measured)
- Display: 2.4" colour LCD
- Operation: keypad, encoder, Wi-Fi or remote via USB and PC app

In contrast to information given in the data sheet, the signal generator can also deliver pulses with a repetition rate up to 15 MHz. The output stage cannot supply a signal much greater than ± 10 V. If you increase the output DC offset to the maximum positive level of + 9.99 V, the output signal is limited to 200 $mV_{PP}$. With a DC offset of -4 V for example, the available output swing is 12 $V_{PP}$ maximum.

## To the bench...

If you have used a signal generator before, the operation of the JDS2915 (apart from the arbitrary signals) is fairly self-explanatory. I was able to produce any waveform and adjust the output without recourse to the PDF manual [3]. The software manual [4] for the JDS2915, is currently only available in German, and actually shows a picture of the JDS6600 signal generator in its plastic housing and makes reference only to the JDS6600 in the text. This makes me think that the JDS6600 series and the JDS2915 are, except for the casing, technically largely identical devices. The JDS2915 software documentation certainly looks like it was originally written for the JDS6600 model.

The software itself is a program created with LabView (**Figure 3**). This '4th generation programming language' from National Instruments is more often used in a professional testing, measurement and control environment, to perform automated production testing. Although the drivers for linking signal generators together


*Figure 1: The complete kit of parts of the JDS2915 signal generator.*

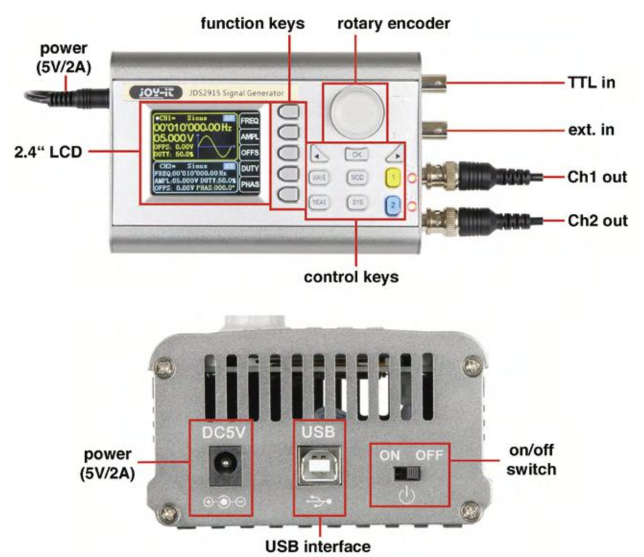



*Figure 2: Layout of the JDS2915 sockets and controls.*

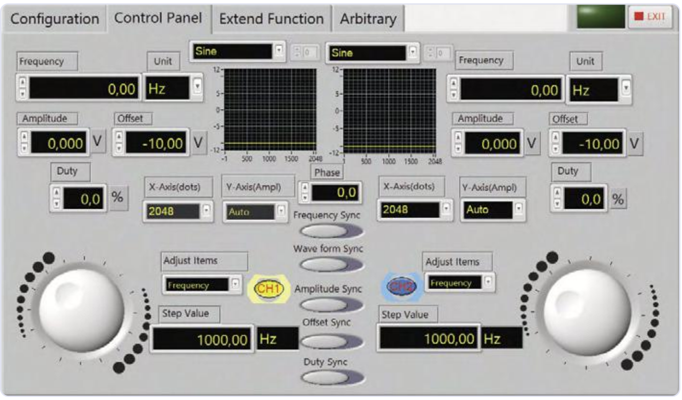



*Figure 3: The PC program for remote control of the JDS2915 uses LabView.*


*Figure 4: The display in three different modes: sinewave (top), pulse (middle) and frequency measurement (bottom).*

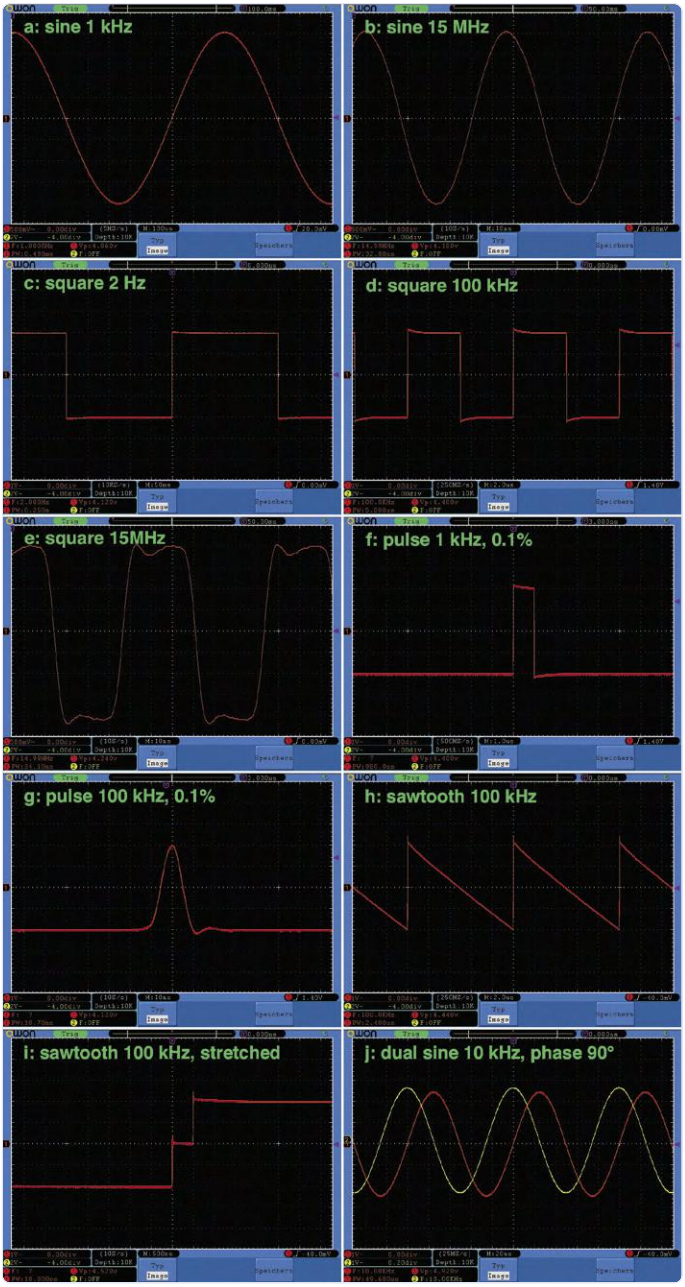



*Figure 5: Waveforms at 4 $V_{PP}$ out. 5a: sinewave at 1 kHz; 5b: sinewave at 15 MHz; 5c: squarewave at 2 Hz; 5d: squarewave 100 kHz; 5e: squarewave at 15 MHz; Pulses with a duty cycle of 0.1%: 5f: 1 kHz and 5g: 100 kHz; 5h: sawtooth or triangle wave at 1 kHz and a duty cycle of 0.1%; 5i: previous signal with the timebase expanded; 5j: two 10 kHz sinewaves with a phase shift of 90 °.*

exist, they are not made available for downloading from Joy-iT (or the software producers). Consequently you are not able to integrate the signal generator into an automated test environment under LabView. That is a real shame — it would be a really cool feature.

## Operation and signals

**Figure 4** shows the display in three different modes (there are more). Above is the setting for a sinewave signal at 4 $V_{PP}$ and 1 kHz. The offset is +1.0 V and 'DUTY' or duty cycle is not operational here or for the squarewave output function. The blue channel 2 is at 10 kHz at 5 $V_{PP}$ and 0 V offset. The parameter 'PHAS' = phase is interesting here, because here you can set a phase shift between the two signals from 0 to 360 °.

The pulse duty factor is useful for the pulse output function and is also interesting when applied to the triangle waveform generator (centre). The output signal is triangular when the duty cycle is set to 50%. Adjusting the duty cycle changes the waveform to produce a sawtooth.

The frequency counter (below) appears when the 'MEAS' button is pressed. For testing I just measured the 1 kHz calibration signal output from my scope. The displayed value wanders between 999 Hz and 1 kHz, indicating that the signal may not be completely symmetrical.

The most important thing about a signal generator is the quality and the range of different waveforms generated. **Figure 5** shows a variety of output signals produced by the JDS2915. The signals have a set amplitude of 4 $V_{PP}$ and were measured using a 100 MHz scope which provides a single-channel sampling rate of 1 GS/s.

### Sinewave

The sinewave in **Figure 5a** looks very clean. If you look more closely however, small steps in the waveform are evident. This turned out to be a feature of the input 8-bit A/D converter of my scope rather than the signal generator output signal. The 14-bit D/A of the generator offers much finer resolution. I checked this by expanding the X and Y axes: the steps only show up when amplified 100 times! Even at 15 MHz, the signal from **Figure 5b** still looks relatively pure with only slight visible distortion. The output signal amplitude remains relatively constant even at this frequency.

### Squarewave

**Figure 5c** shows a squarewave signal at 2 Hz (DC-coupled). The value of 4.12 V displayed is due to the uncalibrated state of my scope's input amplifier. In fact, it's 4.03 V! At 100 kHz, there are slight overshoots at the rising and falling edges shown in **Figure 5d**

— not nice, but acceptable. At 15 MHz in **Figure 5e** limitations in the bandwidth of the analogue output amplifier are noticeable — but the signal still roughly approximates to a square wave.

### Pulse

A pulse with a duty cycle of 0.1% at 1 kHz is shown in **Figure 5f**. The pulse has a width of 980 ns which represents an error of 2%. At frequencies above 100 kHz, the shortest pulse generated with a duty cycle of 0.1% is no longer a step but has a characteristic pulse shape with all its higher frequency components filtered out by the amplifier bandwidth. The pulsewidth is about 10 ns measured at half the peak output voltage level (**Figure 5g**). At frequencies above 1 MHz, the output signal amplitude becomes lower and drops to around 2 V at frequencies over 10 MHz. This is not surprising, since the generator's sampling rate of 266 MHz gives a resolution of just under 4 ns. In contrast to information on the data sheet, you can generate pulses at a repetition rate of up to 15 MHz.

### Triangular

**Figure 5h** shows a 100 kHz triangular output signal that with its duty cycle 'DUTY' adjusted to 0.1% so that effectively a sawtooth waveform is output. Slight overshoot is evident on the rising edges. If you expand the horizontal timebase by five (**Figure 5i**), you can see steps in the rising edge. With a duty cycle of 0.0%, these steps disappear completely to give a clean sawtooth waveform.

### Phase shifting

It's useful to have two independent output signals, locked together but with an adjustable phase shift. The waveforms shown in **Figure 5j** are of two 10 kHz waveforms (at 5 $V_{PP}$) phase-shifted by 90 ° relative to one another. If your scope has X-Y inputs you can use the phase-shifted signals to draw Lissajous patterns on the screen.

## Conclusion

The one aspect of this unit's design which niggles me slightly is the case. An enclosure made of (sufficiently thick) aluminum profile has got to be more robust and offer better electrical shielding and heat dissipation than one made of plastic. For me the unit, at 450 g, is however just too light, especially when compared to my old function generator (**Figure 6**), which weighs in at 2.5 kg (thanks mostly to its power transformer). It doesn't slide across the bench top when I press any of the buttons. As far as the layout goes I would prefer to have the BNC sockets mounted on the front panel — but maybe your needs are different and the socket positions are an advantage.

Technically there is little to complain about with this signal generator with its built-in frequency counter. The signals are surprisingly clean — a big improvement on what I get out of my old generator. A unit offering a wider bandwidth will always have an advantage but in practice I personally hardly ever need a signal frequency higher than 10 MHz and remote control via a PC is a bonus. In this price bracket you wouldn't really expect a LabView driver function to be included. In general: for just under €100 (for Elektor members) this function generator delivers an excellent price/performance ratio.

200143-02


*Figure 6: For comparison: My faithful old (partially analogue) function generator, maybe an XR2206 lurks in there somewhere...*



## WEB LINKS

[1] **Review of the JDS6600 Signal Generator**: www.elektormagazine.com/magazine/elektor-110/51155

[2] **Joy-iT Web site**: https://joy-it.net/en/products/JT-JDS2915

[3] **JT-JDS2915 Manual** (PDF): https://joy-it.net/files/files/Produkte/JT-JDS2915/JT-JDS2915-Anleitung-07.02.20.pdf

[4] **NI VISA setup instructions** (PDF file in German): https://joy-it.net/files/files/Produkte/JT-JD6600/JT-JDS-Software-Anleitung.pdf

# Traffic Lights Programmed in PIC Assembly Code

By **Andrew Pratt**
(United Kingdom)


This little project demonstrates the programming of a set of traffic lights consisting of six LEDs to represent the two sets of red, yellow, and green lights (**Figure 1**). The lights are to imitate the type used at roadworks to control the flow of single-file traffic. The programming is done in PIC assembly code rather than a higher-level language.
On power up, both sets of lights will be at red and remain so for 10 s. Then set A will change to red and yellow for 2 s then to green. After a further 20 seconds Set A will change to yellow for 5 s and then to red. Both sets will remain at red for 10 s before set B goes through the same changes. This cycle will repeat until power is removed. In addition, there is a signal that enables the light sequence — if this signal goes low the sequence stops at the next double red. The Enable signal must be high for the entire 10-s period when both sets are at red prior to starting sequence.
From the written description we need to identify the different states that the system can be in, as follows:

| State | Lights A | Lights B |
|---|---|---|
| 0 | Red | Red |
| 1 | Red - Yellow | Red |
| 2 | Green | Red |
| 3 | Yellow | Red |
| 4 | Red | Red |
| 5 | Red | Red-Yellow |
| 6 | Red | Green |
| 7 | Red | Yellow |

Having identified the eight states that are needed the next thing to do is to work out the allowable transitions and the input conditions to trigger these conditions. This is a very simple example in that the states follow round one after another with only a single allowable transition to the next state. The resulting diagram using the Moore Machine format is in **Figure 2**. The outputs only depend on the current state and are shown on the diagram inside the box for each state.
The transition to the next state is determined by the elapsed time in a particular state with the exception of leaving states 0 and 4 where there are two reds, in these states the enable signal must be present for 10 s. This will cover our requirement for the sequence to continue to the next two reds state if the enable signal goes low. In the circuit

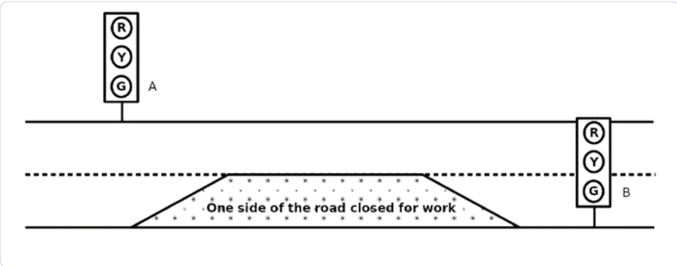



*Figure 1: Roadworks traffic control. Note: traffic direction is left-hand lane i.e. United Kingdom.*

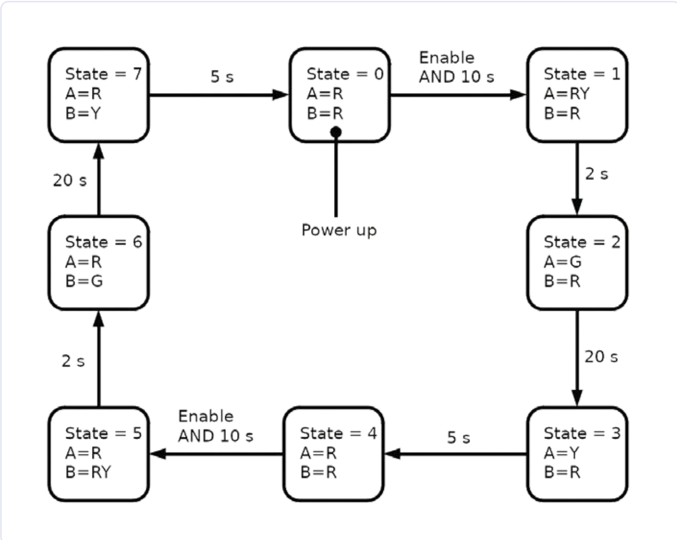



*Figure 2: Roadworks state chart.*

**Listing 1: Complete program for the traffic lights**

```
;PROG_8_03.asm
LIST     P=16F1823
#INCLUDE <p16f1823.inc>
#INCLUDE <fsm_macros.inc>

    RADIX DEC               ; Default numbers are to base 10.
    BOOK_CONFIGURATION      ; See macro in fsm_macros.inc.

    CBLOCK 0x70
    TICKS                   ; Used to count 8.2ms ticks.
    SECONDS                 ; Used to hold the seconds count.
    ENDC

    ORG 0X00
    GOTO START

    ORG 0X04                ; Interrupt vector.
    BCF INTCON, TMR0IF      ; Clear the tmr0 overflow interrupt flag.
    DECF TICKS, F           ; Decrement by one the 8.2ms ticks.
    BTFSS  STATUS, Z        ; Test to see if TICKS has reached zero.
    GOTO $+4                ; If TICKS has not reached zero jump to return from interrupt.
    INCF SECONDS, F         ; If TICKS has reached zero increment the seconds count.
    MOVLW 122
    MOVWF TICKS             ; Restore TICKS to 122 (122 x 8.2 ms 1 second).
    RETFIE                  ; Return from interrupt.

START
    MOVLB 1                 ; Bank 1 required for the following macros, and TRISC.
    SET_FREQ_32MHZ          ; See file fsm_macros.inc.
    SET_TMR0_CASE1          ; Gives 8.2 ms ticks. See fsm_macros.inc.
    CLRF TRISC              ; Set all PORTC as outputs.
    MOVLB 0                 ; Select bank 0 for PORTA and PORTC.
;===============================================================================================
S0
    MOVLW b'00100100'
    MOVWF PORTC                    ; Turn Red A (RC5) and Red B (RC2) on others off.
SS0_0
    CLRF SECONDS                   ; Clear the seconds counter.
SS0_1
    BTFSS PORTA, 5
    GOTO SS0_0                     ; If the enable signal is low return to sub-state 0.
    MOVLW 10
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS0_1                     ; If less than 10 s have elapsed stay in substate 1.
;===============================================================================================
S1
    MOVLW b'00110100'
    MOVWF PORTC              ; Turn Red A (RC5) and Yellow A (RC4) and Red B (RC2) on others off.
SS1_0
    CLRF SECONDS             ; Clear the seconds counter.
SS1_1
    MOVLW 2
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS1_1
S2
    MOVLW b'00001100'
    MOVWF PORTC                 ; Turn Green A (RC3) and Red B (RC2) on others off.
SS2_0
    CLRF SECONDS
SS2_1
    MOVLW 20
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS2_1                  ; If less than 20 s have elapsed stay in substate 1.
```

*continued overleaf...*

```
;=================================================================================
S3
    MOVLW b'00010100'
    MOVWF PORTC                      ; Turn Yellow A (RC4) and Red B (RC2) on others off.
SS3_0
    CLRF SECONDS                     ; Clear the seconds counter.
SS3_1
    MOVLW 5
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS3_1                       ; If less than 5 s have elapsed stay in substate 1.
;=================================================================================
S4
    MOVLW b'00100100'
    MOVWF PORTC                      ; Turn Red A (RC5) and Red B (RC2) on others off.
SS4_0
    CLRF SECONDS                     ; Clear the seconds counter.
SS4_1
    BTFSS PORTA, 5
    GOTO SS4_0                       ; If the enable signal is low return to sub-state 0.
    MOVLW 10
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS4_1                       ; If less than 10 s have elapsed stay in substate 1
;=================================================================================
S5
    MOVLW b'00100110'
    MOVWF PORTC              ; Turn Red A (RC5) and Red B (RC2) and Yellow B (RC1)on others off.
SS5_0
    CLRF SECONDS                ; Clear the seconds counter.
SS5_1
    MOVLW 2
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS5_1                  ; If less than 2 s have elapsed stay in substate 1 .
;=================================================================================
S6
    MOVLW b'00100001'
    MOVWF PORTC                   ; Turn Red A (RC5) and and Green B (RC0) on others off.
SS6_0
    CLRF SECONDS                  ; Clear the seconds counter.
SS6_1
    MOVLW 20
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS6_1                    ; If less than 20 s have elapsed stay in substate 1.
;=================================================================================
S7
    MOVLW b'00100010'
    MOVWF PORTC                 ; Turn Red A (RC5) Yellow B (RC1) on others off.
SS7_0
    CLRF SECONDS                ; Clear the seconds counter.
SS7_1
    MOVLW 5
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS7_1                  ; If less than 5 s have elapsed stay in substate 1.
    GOTO S0

    END
```

diagram given in **Figure 3,** there are six LEDs connected through current limiting resistors. The value of these resistors is not critical, 1 kΩ will give about 3 mA. A PIC type 16F823 is used and it is programmed through an FTDI USB/serial bridge.

Prior to producing the code that runs the above state machine the PIC needs to have certain configurations done such as the configuration file settings, the oscillator frequency and the ports configured. It is best to progress in small parts and test as you go. This testing takes the form of adding debug code to prove that each bit you add actually works and can be depended on — writing a complete program and then trying to get it to work is not the way to go.

The top-level state machine of the assembly code program is given in **Figure 4**, and the actual PIC assembly code, in **Listing 1**. The program listings and associated discussions of the components in the program

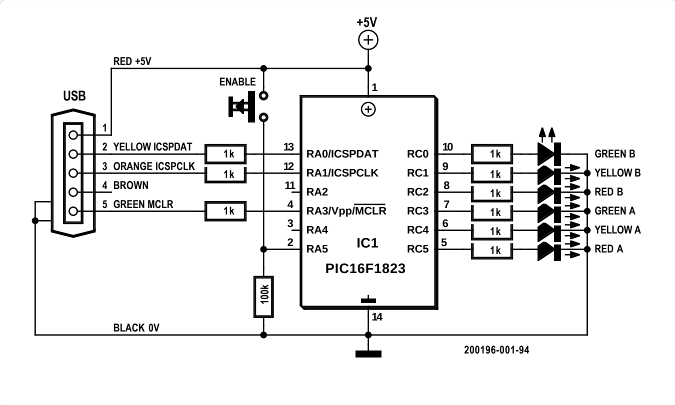



*Figure 3: Circuit diagram of the (virtual) traffic lights control system.*

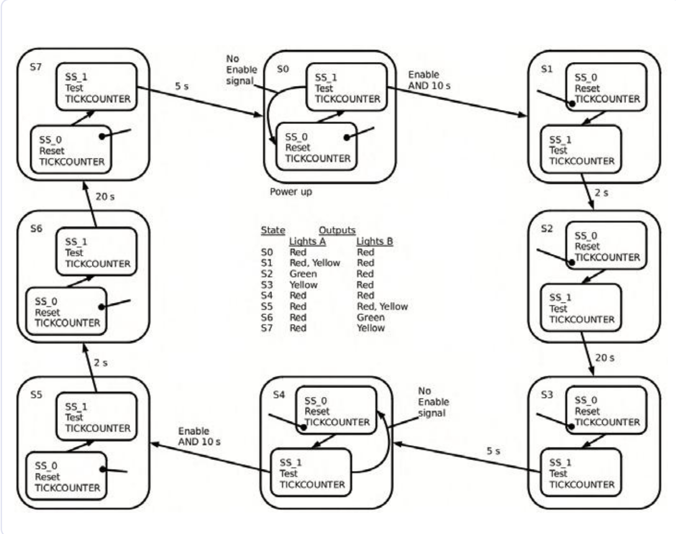



*Figure 4: Complete state chart for the traffic lights.*

arranging the timekeeping and input/output checking are unfortunately beyond the scope of this article, and the same applies to a special test-as-you-go program and a mini debugger for the internal seconds counter. The latter two are helpful to build the final program in a step-by-step, educational way. All (sub)routines and the associated state diagrams may be found at [1] though, for free downloading.

200196-01

## WEB LINK

[1] Article resources and support page: www.elektormagazine.com/200196-01



# E-E-e-ternal Blinker

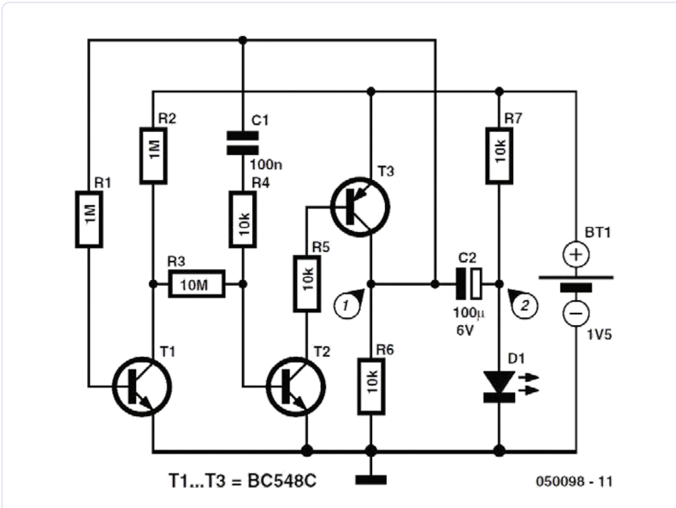
By **Burkhard Kainka** (Germany)

You occasionally see advertising signs in shops with a blinking LED that seems to blink forever while operating from a single battery cell. That's naturally an irresistible challenge for a true electronicist.

And here's the circuit. It consists of an astable multivibrator with special properties. A 100-µF electrolytic capacitor is charged relatively slowly at a low current and then discharged via the LED with a short pulse. The circuit also provides the necessary voltage boosting, since 1.5 V is certainly too low for an LED.

The two oscillograms aid in illustrating how the circuit works. The voltage on the collector of the PNP transistor jumps to approximately 1.5 V after the electrolytic capacitor has been discharged to close to 0.3 V at this point via a 10-kΩ resistor. It is charged to approximately 1.2 V on the other side. The difference voltage across the electrolytic capacitor is thus 0.9 V when the blink pulse appears. This voltage adds to the battery voltage of 1.5 V to enable the amplitude of the pulse on the LED to be as high as 2.4 V. However, the voltage is actually limited to approximately 1.8 V by the LED, as shown by the second oscillogram. The voltage across the LED automatically matches the voltage of the LED that is used. It can theoretically be as high as 3 V.

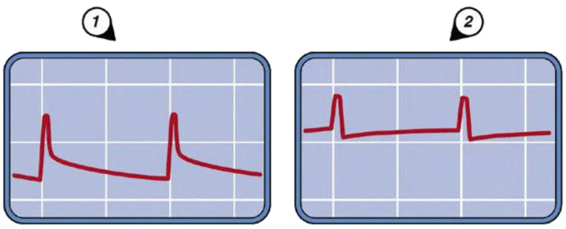
The circuit has been optimised for low-power operation. That is why the actual flip-flop is built using an NPN transistor and a PNP transistor, which avoids wasting control current.




The two transistors only conduct during the brief interval when the LED blinks. To ensure stable operating conditions and reliable oscillation, an additional stage with negative DC feedback is included. Here again, especially high resistance values are used to minimize current consumption.

The current consumption can be estimated based on the charging current of the electrolytic capacitor. The average voltage across the two 10-kΩ charging resistors is 1 V in total. That means that the average charging current is 50 µA. Exactly the same amount of charge is also drawn from the battery during the LED pulse. The average current is thus around 100 µA. If we assume a battery capacity of 2500 mAh, the battery should last for around 25,000 hours. That is more than two years, which is nearly an eternity. As the current decreases slightly as the batter voltage drops, causing the LED to blink less brightly, the actual useful life could be even longer. Feel challenged? Let us know on www.elektormagazine/labs.

200200-01





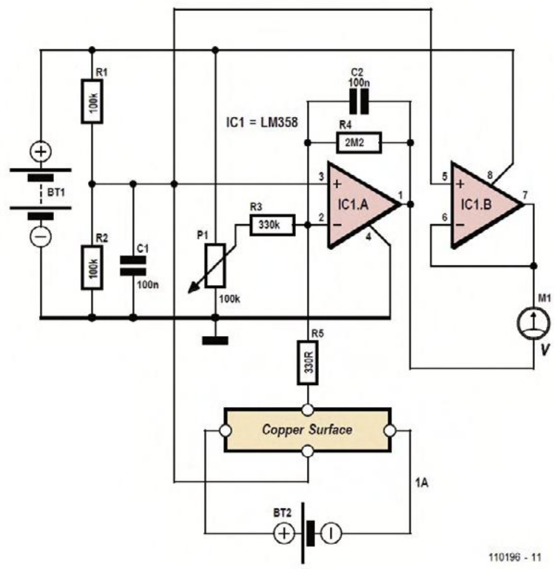

# Experimental **Hall Sensor**

By **Burkhard Kainka** (Germany)

Hall sensors can of course be purchased but making them yourself is far more interesting (and satisfying)! According to the theory the crucial thing is to use a touch layer that's as thin as possible; the length and width are unimportant. An 'obvious' starting point for our trials would be copper, which in the form of printed circuit board material is easy to find and handle. Copperclad board may be obvious but not ideal, because it has a very weak Hall constant. Nevertheless, we should be able to use it to demonstrate the Hall effect by using very powerful magnets in our sensor.

To achieve detection, we need the highest possible level of amplification. In the circuit shown here the voltage amplification is set by the relationship of the two feedback resistors of the first op-amp. With the values given (2.2 MΩ and 330 Ω) produce a gain of 6,667. This also creates a convenient bridge connection for taking measurements. The trimmer potentiometer allows fine adjustment. With zero setting that's accurate to within millivolts we could use this test point to measure Hall voltages of well below a microvolt. Finally, in this way we could also measure the flux density of a magnet.

Copper has a Hall constant of




$A_H = -5.3 \times 10^{-11}$ m³ / C

The thickness of the copper layer is $d$ = 35 µm. The Hall voltage then amounts to:

$V_H = A_H \times I \times B / d$

With fieldstrength $B$ = 1 T and current $I$ = 1 A, a Hall voltage of $V_H$ = 1.5 µV is produced. The 6,667-fold gain then achieves a value of 10 mV. The circuit thus has a sensitivity of 10 mV per tesla. That said, adjusting the zero point with P1 is not particularly easy. The amplifier has a separate power supply in the form of a 9-V battery (BT1). To take measurements we connect a lab power supply with adjustable output current (BT2) to the Hall sensor (the copper surface) and set the current flowing through the sensor to exactly 1 ampère. Then the zero point must be adjusted afresh.

Next, we place a strong Neodymium magnet below the sensor. The output voltage of the circuit should now vary effectively by several millivolts. Note that there are several effects that can influence the measurements we take. Every displacement of the magnet will produce an induction voltage in the power feed wires that is significantly greater than the Hall voltage itself. Every time you move the magnet you must wait a while to give the measurements time to stabilize. With such small voltage measurements problems can also arise with thermal voltages due to temperature variations. It's best not to move and inch — and to hold your breath as long as possible!

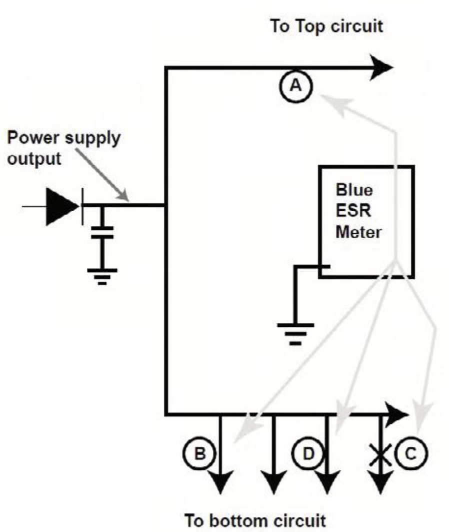
200198-01

# **Tracing Shorts** with the Milli-Ohm or ESR Meter

By **Paul Hetrelezis**

Because the ESR meter is essentially a low ohmage (or resistance) meter, it can check shorting between tracks, particularly if a schematic of the PCB is available as a reference — by making a reference point with one probe on a PCB track and tracing the other probe towards and away from that reference point.

A good example would be making a supply line as a reference point then the probe at the ground (GND) points. The illustration shows two power supply lines branching from the power supply diode and filtering capacitor. Those lines are at point A and the other is the supply line with shown points B , D and C in that order. Suppose there is a short circuit contained in the circuit in either these points lines. That is either at supply line where point A is located or at the alternate supply line where points B, D or C are located. The non-ground probe is moved slowly towards point A and a noted increase in resistance is occurring.




This indicates that you are on the wrong track as you would expect this with the probes being separated further causing higher resistance track readings. Now you reverse the probe movement and move to points B, D and C in that order. You see the reading slowly decreasing with the lowest reading occurring at point C. This indicates a short in the circuit that area C is located. Note that this method only works for single sided soldered PCBs.

## Tips

- When performing a junction test with a DMM in circuit across a semiconductor junction with a paralled capacitor, the reading will be false due to the charging and discharging effects of the paralled capacitor. It is best here to measure these components, particularly suspect ones that are suspect out of circuit.
- When measuring a board component for resistance, it is wise to first to take the reading from the component side of the board where possible. This will ensure the component is not showing an open circuit reading due to actually soldering of the checked component being a dry joint. Also ensure there are no parallel paths of conduction with the component being measured.

200212-01


# Elektor
# SDR Hands-on Kit

By **Harry Baggen** (The Netherlands)

Software defined radio (SDR) allows receiving and decoding of radio signals using relatively simple hardware and a computer. The 'SDR Hands-on kit' from Elektor contains all the necessary parts to make a start with this modern form of radio reception. Here we give an impression of the features of this affordable kit and our initial experience with SDR.

Elektor has already given much attention to SDR (software defined radio) over the years. Not just the technology, but also in the form of practical and easy-to-build hardware. Already in 2007 there appeared an SDR receiver in the form of a shield for an Arduino Uno, where the Uno took care of the tuning of the oscillator on the SDR board. The audio-out signal from the circuit board was sent to a PC, which used special SDR software to distil the received audio signals. This way, signals from 150 kHz to 30 MHz could be received. This popular SDR receiver has since been followed by two successors, the most recent dates from 2016. For a few tenners you can buy the ready assembled board of the Elektor SDR Shield 2.0 and it offers a great opportunity for an introduction to the SDR phenomenon.

## Complete kit

For some time now, Elektor has been offering a so-called SDR Hands-on kit, which contains a hands-on book, an assembled SDR


*Figure 1: Contents of the Elektor SDR Hands-on Bundle.*

*Figure 2: This you have to supply yourself: an Arduino Uno, USB cable, audio jack cable and a piece of wire.*

shield, a number of headers, two special toroidal cores (**Figure 1**) and a length of thin coaxial cable. The only thing that you have to provide yourself is an Arduino Uno (but most of us will have one of those already — **Figure 2**)

The book in this kit is the *SDR Hands-on Book* written by the developer of the SDR shield, Burkhard Kainka. He explains in a clear way how to make a start with the SDR shield. He also shows how you can use different SDR software programs and pays a great deal of attention to antennas, an essential element for obtaining good reception. Of course, the software for the Arduino is also described in detail. And finally he offers a few bigger projects based on the SDR shield: a stand-alone receiver, a WSPR transmitter and a QRP transceiver.

A review of the book has already been carried out by Elektor RF specialist and Retronics Conservator, Jan Buiting, PE1CSI. His review [1] is pretty extensive and I encourage you read it. Speaking for myself, I am more familiar with audio and measurement technology and as a non RF specialist I thought it would be interesting what my experience with this kit would be.

## The first steps

The first step was soldering the supplied pinheaders into the circuit board. The headers are supplied separately because of practical considerations. I also fitted a 2-way pin header into the connection for the antenna input. This is very convenient when connecting an antenna. There should have been a few more of those pinheaders in the box, because I can imagine that not everyone will have these in their parts collection.

The circuit board was now plugged into an Arduino and the Uno could be connected to a PC or laptop. This requires a USB cable (USB-A to USB-B) for the data connection and a 3.5-mm stereo jack cable for the audio connection. Then it was time to grab the book. In the first few pages the author briefly describes the hardware on the SDR shield and it then immediately turns practical with the introduction of the G8JCFSDR software. Downloading and installing the software was done quickly. A big advantage of this software is that the Elektor SDR shield is fully supported by G8JCFSDR (**Figure 3**). After selecting the shield in the configuration menu, the software offers the option of loading the appropriate frequency-control software into the Arduino, without the need to install the Arduino IDE. This also went quite quickly. Note that in G8JCFSDR, at the heading Display, you have to select the option "full", otherwise the Config button is not visible. After I hooked a piece of wire to the antenna input, I could go on the quest of finding a transmitting station. Before you realise it, the afternoon is gone and you have tried all kinds of options and settings in the software. I have in the past played a little with a simple short-wave receiver, but this is much more fun and offers an infinite number of possibilities! After gaining this first experience I could continue with installing

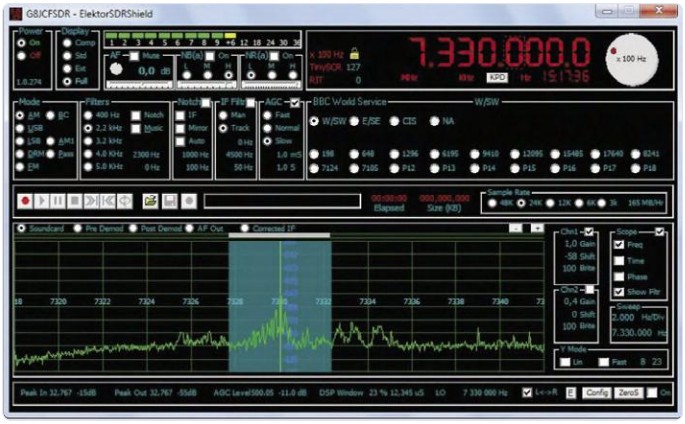



*Figure 3: The G8JCFSDR software is ideal to begin with, the SDR shield is fully supported.*

the separate PC tuning software for the oscillator on the SDR shield (all software written by the author of this book is available from the Elektor website [2]). This is necessary when using other SDR software, such as SDRsharp [3], the next software program that is described in the book. This offers an entirely different feature set compared to the first program.

## Conclusion

To be honest, I didn't progress any further than what I described above, and that covers only the first 40 pages of the book. The next chapter is about antennas, and in my experiments I already quickly discovered how important the antenna is. I will certainly continue with the book and try different antenna types and connection options. The two supplied toroidal cores will be very useful for these experiments. I will also work through the remaining chapters, but I think that I will be quite busy for a while with the antenna experiments. This is a very nice kit that doesn't cost much and will gain you a great deal of experience in the field of Software Defined Radio.

200167-04



## WEB LINKS

[1] **Book review**: www.elektormagazine.com/news/review-sdr-hands-on-book

[2] **Software**: www.elektor.com/sdr-hands-on-book

[3] **SDRsharp**: http://airspy.com/download/

# Elektor Labs Pipeline

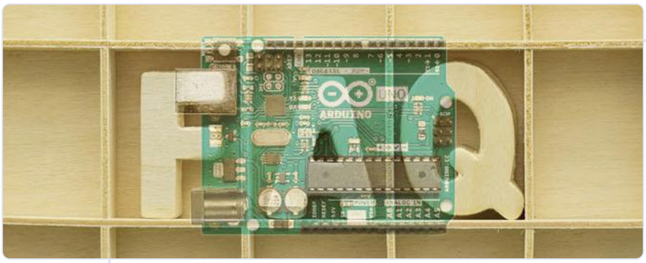
By **Clemens Valens** (Elektor Labs)

Elektor Labs is an open platform where anyone can post projects about anything as long as it's electronics related. The results is a rich collection of solutions for all sorts of problems and issues that you didn't even know existed. Here are a few.

## Build a **Portable** PM2.5 Particles Monitor

Microscopic particles of solid or liquid matter suspended in the air impact human health through climate changes and direct inhalation. Also known as *particulate matter* (PM) the fine dust particles are commonly classified by size. PM10 designates particles with a diameter from 2.5 to 10 µm; PM2.5 is the category of fine particles with a diameter of 2.5 µm or less. PM2.5 pollution is considered particularly deadly as it can cause lung cancer. Did we scare you? Then this project is for you.

www.elektormagazine.com/labs/3969




## Please Add the Elektor Labs **Arduino FAQ** to Your Favourites

We get many questions about Arduino. Most of them have been asked before, and some even several times. There is of course a countless number of webpages out there which treat most of these questions in some way, but it means digging through forums and blogs to find the answers. As this can be quite time consuming, Elektor Labs decided to compile their own Frequently Asked Questions (FAQ) for Arduino and write down the answers once and for all. So, before asking us, check our Arduino FAQ to see if the answer to your question is already there.

www.elektormagazine.com/labs/1876

## Request and Parse **JSON Data** with a Microcontroller

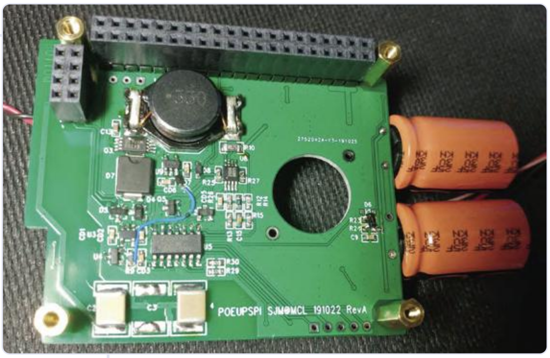
Although the example used in this project is probably only useful for people travelling through the Netherlands (you know, the capital of Denmark), it does show how to use a microcontroller to extract data from a website. In this case it's an ESP32 requesting and parsing JSON data and then displaying the information of interest on a small OLED display.

www.elektormagazine.com/labs/3965

## Power Your Always-On Raspberry Pi **over Ethernet**

How do you power a highly reliable LoRa gateway that can be trusted not to trash its SD card and can be remotely located and managed? With this Power over Ethernet (PoE) powered uninterruptible power supply (UPS). A couple of 25-farad supercaps provide enough backup time to cover ten seconds of power outage plus another fifteen seconds to cleanly shut down the Pi if necessary.

www.elektormagazine.com/labs/3948



## Music **Stimulates** All Living Beings

Recent studies suggest that exposing living creatures of all sorts, animal or vegetal, to music has positive effects on their development and well-being. Inspired by this idea, the author developed a system to stimulate the growth of tomatoes inside a small greenhouse.

www.elektormagazine.com/labs/3910

## Model Your Body **ESD-Wise**

Electrostatic discharges (ESD) can destroy electronic devices or cause system malfunctions. Professional ESD simulators are expensive, but a cheap high-voltage generator combined with a small microcontroller can be a good alternative. The circuit presented here conforms to the Human-Body model and will work more reliably than your finger(s).

www.elektormagazine.com/labs/3909

## **Multilingual** Speaking Clock

As clocks go, this one too displays time and date, and it does so on a small TFT screen. However, this clock also announces the time & date at predefined moments or at the press of a pushbutton. Because the user is expected to record his/her own time messages, the clock is truly multilingual (and, in fact, can say anything you like).

www.elektormagazine.com/labs/3875

## Build an LED **Gravity Balance**

The effect of gravitational acceleration on an object is demonstrated with the help of a virtual ball rolling and bouncing on a Neopixel-based LED bar. This project has no practical use whatsoever but then, who cares?

www.elektormagazine.com/labs/1942

180571-G-01

**70% Funded**

NEW FEATURE

# ELEKTOR ~~KICKSTARTER...~~

## Well, you get the idea.

By **Erik Jansen**

Elektor has always supported entrepreneurial spirit. For years, we have been offering the opportunity to give innovative ideas a stage on the online LAB, and we regularly select projects that we publish — for a fee — in our media and even produce the resulting products in our online store.


Come up with a name and **win** a voucher for the Elektor Store

elektormagazine.com/kickstarter

In addition, together with Messe Munich, under the label *Fast Forward Awards*, we provide young start-ups with an opportunity to present their innovations and business at the largest electronics trade fair in the world. We offer them our platform, our network, and our advice to grow faster and get in touch with the right people.

We are proud of this. And it's therefore not for nothing that we changed our motto last year from "Learn > Design > Share" to

design > share > sell electronics

where we expressly mean that the Elektor Store is also at the service of our community.

### Kickstarter

In order to reinforce the latter, we have expanded the online Elektor LAB with a new add-on which we fondly call "Elektor Kickstarter." The reason for this is quite obvious: we first test the viability of a project as a product in the Elektor Store with our own supporters.

A project that is suitable for this purpose can be registered as "Elektor Kickstarter" by its creator. If the project is approved by the Elektor review team, we will discuss a price and the minimum amount we should be able to sell for the project to be viable. We then see if there are enough people to "back" the project to start production.

We do all this without any actual money flowing back and forth. It is a commitment that can simply be waived. However, only the backers get the first chance to purchase the product at a nice discount as soon as it appears in our shop.

### Impossible name

No matter how nicely it covers what we are trying to achieve (and with great respect to the actual "Kickstarter" concept), we obviously can't use the name Elektor Kickstarter. We also thought of "The Elektor Pledge," but that name didn't make it either. That is why we thought it would be nice to ask our own community to think along with us in order to come up with a good name for this new feature.

In addition to the eternal fame, the creator of the final name will receive a nice voucher worth 100 euros ready to spend in the Elektor Store.

Send your suggestion to our LABS community manager Clemens Valens and he will, completely impartially, collect the best suggestions. Of course, we will also give this election some more attention in our upcoming editions!

## How It Works

### How can your project participate?

**A** After a project has been completed, the owner of that project can initiate the request for it to be produced. To do so, only one button needs to be clicked. By the way, you can also do this for your already published projects. This new feature is not only available for new projects.

**B** If a user applies, Elektor's own LAB engineers will decide whether the project fits. The considerations here are based on subjective issues such as the level of innovation and uniqueness as well as our past experience. It is difficult to make this a more transparent process.

**C** If we believe in a good chance of success, we will cooperate to estimate how much the final product will cost, how much budget is needed for a first production run, and make it available to all our readers to support it.

### What does it mean to "back" a project?

**A** When a visitor to the online Elektor LAB "backs" a project, they commit to buy the project at the estimated cost as soon as it is available in our shop. This support is **not binding**, and there is no need to pay yet.

**B** As soon as the project is "fully funded," Elektor will start the production process. Once finished, the product will be put in the Elektor online stores.

**C** Backers will be the first to receive an email with a personal coupon code with which they can buy the product at a nice discount.

**D** Backing can be done in two ways:
- If you are already registered with Elektor, all you must do is click the "Back This Project" button.
- If you are not registered yet, you can also participate by first filling in your email address.

**E** In all cases, you will receive an email to confirm that you want to support the project.

**F** If a project receives funding, or if a project doesn't make it within the set time, we will also let you know.

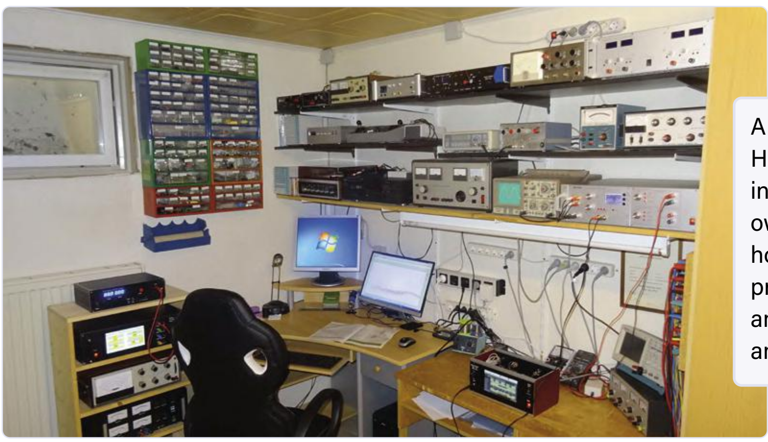
# Do You Really Need **All That Stuff???**

## This is where I spend many hours working on electronics projects...

...but also where I find the peace & quiet to learn, write, study, surf, dream and even meditate without family members who keep asking me where the d^#n car keys are.

Compiled by **Clemens Valens** (Elektor Labs)

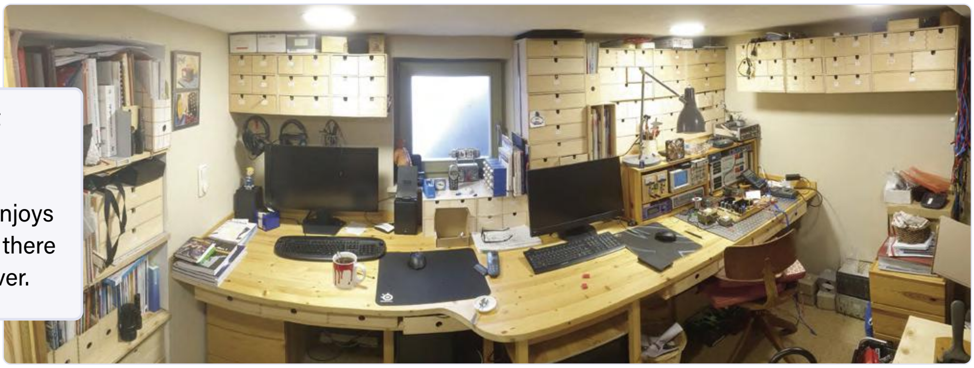
**Tam Hanna's** electronics workspace is located in a historic building in Budapest, Hungary and loaded with a wide variety of tools for design and testing, including an LCR meter, impedance analyzer, curve tracer, and drill press. The Iwatsu DS-6612A digital storage oscilloscope still makes a lot of hours. Pegboards and Stanley boxes are used for storing equipment and keep the space tidy.



A mixture of vintage instruments (mostly Heathkit and Radio RIM) as well as instruments designed and built by the owner himself equip **Rainer Schuster's** homelab. On the workbench, work-in-progress: a Raspberry-Pi-4-based guitar amplifier running 'Guitarix' virtual guitar amplifier software for Linux.

This is just part of the impressive homelab owned by **Philippe Demerliac** (a.k.a. **Cyrob**, www.cyrob.org). Philippe loves electronics instruments and has been collecting them for more than 40 years. The collection includes over 200 Tektronix TM500 series plugins, most of which are operational but, due to time constraints, not calibrated.

What once was a children's room is now **Jos Verstraten's** home electronics lab. This is where he enjoys testing inexpensive kits and modules. He currently stores a variety of handy electronics test and measurement equipment to the left of his computer and printer. All the necessary connected wires and cables are neatly tucked out of the way under his desk.



This is a 180° view of the homelab that **Matthias Key** built in the basement of his house. Having been an electronics enthusiast since around 1985 he still enjoys his hobby — „*it's like meditation*" — as there are so many things to learn and discover.

**Christian Weber** set up this workshop in his home basement with everything required to do most common household repairs and to keep his tube radios in shape. Christian photographed his homelab while sorting a large amount of parts he received out of an inheritance.

200179-01

## SHOW US YOUR ELECTRONICS WORKSPACE!

Want to share details about your electronics workspace with the Elektor community? Fill out this online form so our editors can contact you! www.elektormagazine.com/pages/workspace-submission

# The Kinetic Clock Maker

## Stargate jigsawed!

By **Eric Bogers** (Elektor Netherlands)

As a result of the small competition held on the Elektor Labs website back in July 2019, we've been 'visiting' some interesting home labs the past few months. However, a number of readers have made the point that while it is fun to see where others are realizing their projects, it's even more fun to see what projects they are working on. We duly act on that request!

*Figure 1: Peter Neufeld's laboratory.*

In this first episode 'new-style' we pay a visit to Peter Neufeld's lab — he's no stranger to many Elektor readers, being the author of the acclaimed article series *BASIC for the ESP32 and ESP8266*. In the last episode [2] it turned out that Mr. Neufeld is a fan of clocks of all shapes and sizes, as shown by the electronic hourglass described as an example.
**Figure 1** shows a view of Mr. Neufeld's homelab. On a bookshelf we see the prototype of the hourglass clock, right next to an Elektor book. Of course, the usual equipment is present, such as an oscilloscope and a multimeter. On the right you can just see a soldering station, and on the workbench some projects currently being worked on. On the left side, we also see a receiver and two transceivers, which suggests that Mr. Neufeld is also a radio amateur.

### Stargate

On the workbench you can also see a hardware/software project that Mr. Neufeld developed for a friend. It is a clock with two NeoPixel rings and a number of moving parts, resulting in a kinetic object that can even show time in the old-fashioned 'analogue' way. **Figure 2** shows this object up close. The hours are displayed on the inner NeoPixel ring, and the minutes and seconds on the outer ring. The large wooden ring on the outside rotates around the object every minute for a few seconds using a miniature motor; with a second motor the holding gear mechanism is moved, but as the author himself says, this is actually only done because it looks interesting. **Figure 3** shows the wooden 'inside' of the clock up close — that's a lot of hours jigsawing...
The electronics (hidden in the base of the kinetic object) don't really amount to much — the LEDs (86 in total) and both motors are controlled by an ESP8266 (in the shape of a WeMos D1 mini Pro). Considering Mr. Neufeld's aforementioned article series, it

*Figure 2: Close-up of the kinetic NeoPixel clock.*

will come as no surprise that the overall code has been developed with Annex WiFi RDS [1]. In **Figure 4** you can see the diagram of the electronics as outlined by the author.
The colours of the 'hands' of the clock can be set via a web interface (**Figure 5**) (and, if desired, can also be saved permanently); various lighting effects are also possible. At [3] you can admire the kinetic object in full glory in a short video. Because the whole thing resembles the movie (and TV series) 'Stargate', the author named his creation accordingly.
The whole project has also been described on Elektor Labs [4].

200208-04


*Figure 3: A masterpiece of jigsawing.*

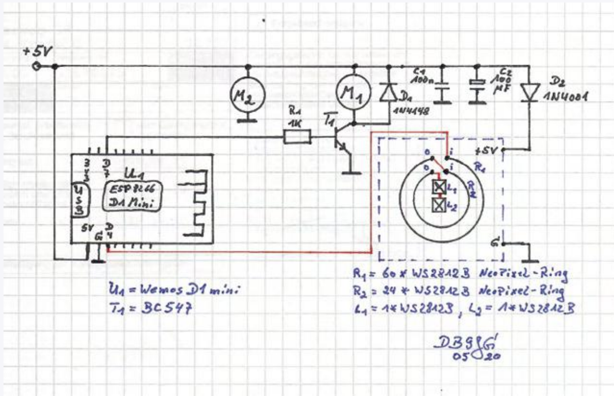



*Figure 4: The electronics aren't particularly complicated.*

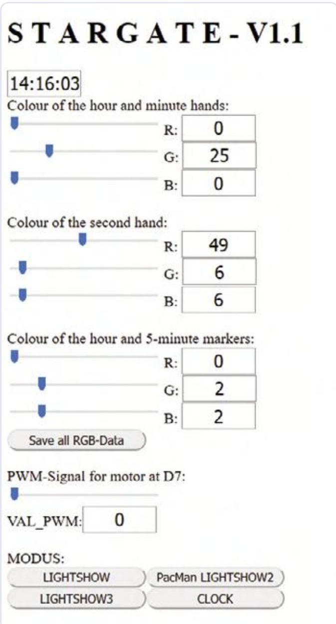



*Figure 5. The web interface of the NeoPixel clock.*

## WEB LINKS


[1] BASIC for the ESP32 and ESP8266 (1): www.elektormagazine.com/190400-03

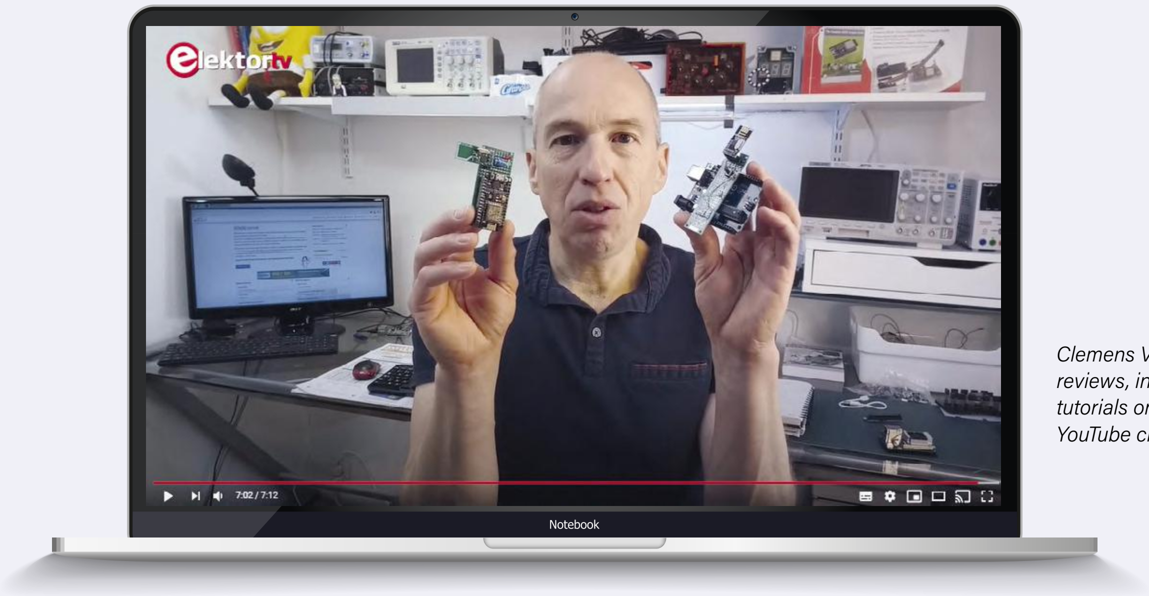
[2] BASIC for the ESP32 and ESP8266 (2): www.elektormagazine.com/190400-B-04

[3] Kinetic Video Object video: https://vimeo.com/387921846

[4] Kinetic Object Clock at Elektor Labs: www.elektormagazine.nl/labs/kinetic-objekt-clock-with-neopixel-ring

# Hello, World!

## We're Elektor, and We're Social




*Clemens Valens posts reviews, interviews, and tutorials on Elektor's YouTube channel.*

Elektor is all about sharing ideas. Since the 1960s, we've been working to nurture the electrical engineering community with projects and design inspiration. Today, we publish in four languages and we use our social media channels to support and collaborate with our members.

Elektor is all about collaboration and sharing ideas. Since the 1960s, we've been working to nurture the electrical engineering community with electronics projects, engineering insights, technical tutorials, and design inspiration. Today, we publish in four languages (English, German, Dutch, and French), and each month we work hard to gain new members — professional engineers, electronics makers, and technical students — from around the globe. Besides using our well-known magazines — *Elektor* (EN/DE/NL/FR) [1][2][3][4], *Elektor Industry* (EN/DE/NL), *MagPi* (FR/NL) [5][6], and *Make:* (NL) [7] — we use the Elektor social media channels to support and collaborate with our members.
According to Hootsuite [8], 3.8 billion people use social media. As for using social media platforms specifically for electronics-related discussions, our marketing team recently found that 40% of Elektor members regularly use social media to keep track of electronics-related news, learn about new products, and to share projects and experiences. And the team believes our social reach will continue to grow, as we engage new audiences, attract more students, and educate existing members about all the benefits of communicating with like-minded innovators via any number of social media platforms.

### Elektor social media channels

There are dozens of active major social network sites. But it doesn't make sense to be everywhere, so we focus our efforts to Facebook, Instagram, Twitter, LinkedIn, and YouTube. Want to know what we're doing on each channel? Read on.
**Facebook**: Our members use Facebook mainly for updates (DIY projects, news, products, and offers) and to interact with our teams (editorial, engineering, and customer service). We currently have 43,000+ followers on the platform. Choose your language!

- Elektor Labs [9]
- Elektor Deutschland [10]
- Elektor France [11]
- Elektor Nederland [12]

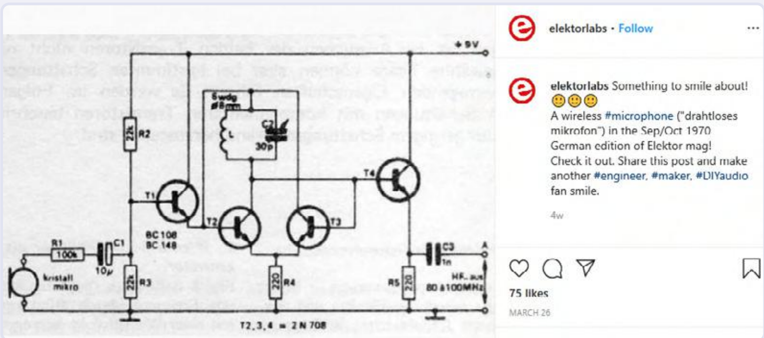
**Instagram**: Instagram is an excellent platform for sharing images and thoughts about products and design projects. When our team posts to Instagram, we try to highlight a project, a useful product, or a member of our community (say, an engineer's workspace). Although post mostly in English, all members of our community find the content useful and engaging.

- ElektorLabs [13]

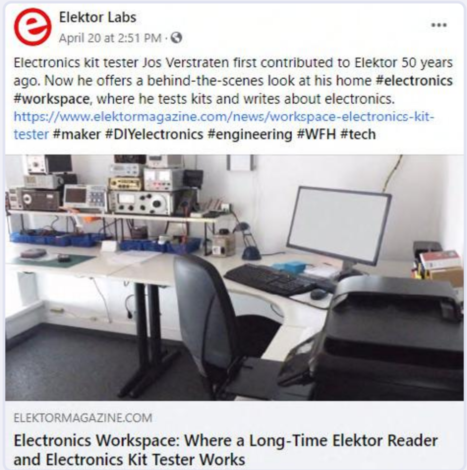
**YouTube** : During the past several months, we've really put a lot of effort into the content we post on the Elektor TV YouTube channel. Thanks to our Technical Manager, Clemens Valens, we are posting a few videos per month. Recently, Clemens created


*Use LinkedIn to learn about electronics-related events and projects. Chat with our Client Team!*


*Follow us for regular news items, DIY projects, web store updates, and more!*




*Our Instagram account is for fun posts as well as new product updates!*




*Use Facebook to interact with our team, read articles, and more.*

videos on topics such as optimizing IoT power consumption; home automation; mousejacking; and how to publish on GitHub. We also publish details product reviews, interviews, and tutorials (e.g. Modern Valve Electronics). Subscribe today so you don't miss a new video!

- ElektorTV [14]

**Twitter**: Well over 10,000 followers use Elektor's Twitter channels short updates on projects, new products, and special offers. It's also a great tool for messaging our editorial, engineering, and customer service teams. Want to publish an article in Elektor? Want to share details about your workspace? Want to sell a product in the Elektor webshop? Reach out on Twitter in the language of your choice.

- @Elektor [15]
- @ElektorNL [16]
- @ElektorFR [17]
- @ElektorDE [18]
- @Clemens_Elektor [19]

**LinkedIn**: Over the past few months, we've received quite a few messages through LinkedIn. Some Elektor members follow the Elektor International Media account. Other are connected with individual Elektor members. One thing is for sure: LinkedIn is a great tool for showcasing your engineering skills and learning about new industry opportunities. Whether you are looking for a new job or interested in networking, we recommend that you follow our corporate account (and our staffers) and check in frequently.

- Elektor International Media [20]

## Ready to get social?

We encourage all our community members to follow at least one Elektor social media account. We share much more than just updates about *Elektor magazine*. We post DIY projects, technical tips, need-to-know electronics industry news, interviews with top engineers, new product announcements, special offers, and fun tidbits to make electronics enthusiasts smile. Whether you need a project idea, want to learn about a new microcontroller, or have technical question, we've got you covered.

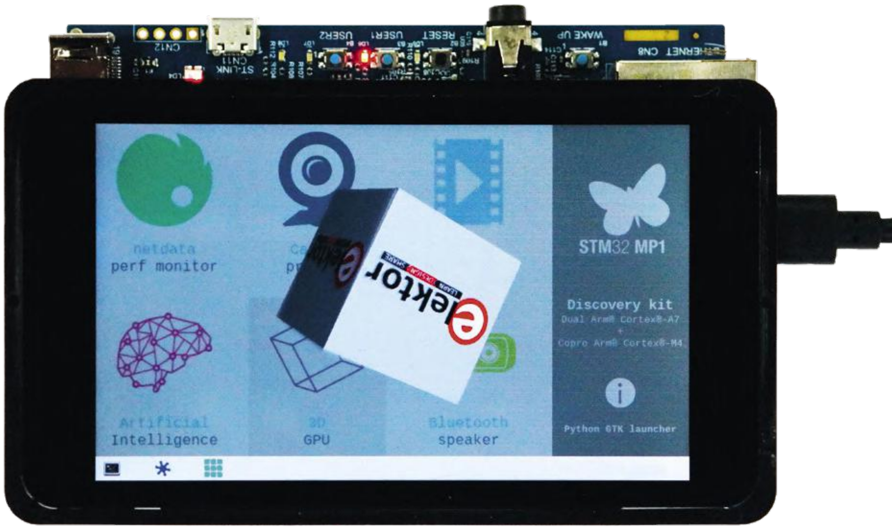
200232-01

## WEB LINKS


[1] www.elektormagazine.com
[2] www.elektormagazine.de
[3] www.elektormagazine.nl
[4] www.elektormagazine.fr
[5] www.magpi.fr
[6] www.magpi.nl
[7] www.makemagazine.nl
[8] blog.hootsuite.com social-media-statistics-for-social-media-managers
[9] www.facebook.com/ElektorLabs
[10] www.facebook.com/ElektorDE
[11] www.facebook.com/ElektorFR
[12] www.facebook.com/ElektorNL
[13] www.instagram.com/elektorlabs
[14] www.youtube.com/ElektorIM
[15] twitter.com/Elektor
[16] twitter.com/ElektorNL
[17] twitter.com/ElektorFR
[18] twitter.com/ElektorDE
[19] twitter.com/Clemens_Elektor
[20] www.linkedin.com/company/elektor-international-media

# M4 + 2x A7 + GPU:
# An Unbalanced Dream Team

## Latest STM32MP1-SoCs can tackle demanding applications


(source: Viacheslav Gromov)

By **Viacheslav Gromov** (Germany) (leser@gromov.de)

When we consider the level of processing demanded by more sophisticated applications such as I-IoT, cryptography and increasingly complex Human-Machine Interfaces (HMI) it becomes clear that a solution based on an ARM Cortex-M7 processor is reaching its limits. To tackle this performance shortfall, many of the major microcontroller manufacturers now have their own range of so-called Heterogeneous Multi-core Processors (HMPs) on the market. These devices combine large and powerful application cores to handle the HMIs and dedicated smaller cores to handle fast real-time applications all in one package. Here we take a closer look at an HMP example from STMicroelectronics; these devices are particularly interesting in terms of unit price and the extensive level of documentation. The main advantages of the MP1 family are their energy efficiency, built-in 3D GPU, Linux and Android support and high level of data security, made possible by effective internal isolation of system components and robust security zones.

This necessarily short article is only intended as an introduction to the main features and capabilities of HMP devices. It is aimed primarily at developers with some experience of ARM Cortex-M systems or who have delved deeper into embedded Linux applications using some of the most advanced features of the Raspberry Pi (RPi).

**Table 1. Characteristics of the MP1 family. The C variant of each type indicates an integrated HW crypto unit while the A variant does not have this feature.**

| Description | Cortex-A7 (650 MHz) | Cortex-M4 (209 MHz) | Vivante 533 MHz GPU with MIPI-DSI | FD-CAN |
|---|---|---|---|---|
| STM32MP151 | 1 | 1 | 0 | 0 |
| STM32MP153 | 2 | 1 | 0 | 2 |
| STM32MP157 | 2 | 1 | 1 | 2 |

### The MP1-family of devices

An overview of the MP1 family of HMP devices is given in **Table 1**. They differ mainly in the core configuration and the integration of a GPU. The most comprehensive version is the *STM32MP157C*. This has all three cores (ARM-Cortex-A7-Siamese twins with shared L2 cache and the ARM-Cortex-M4), a 3D GPU including DSI, a crypto unit and an FD-CAN Interface (**Figure 1**).

Due to the A7 core on the wafer, the entire chip including the M4 cores can only be fabricated using pure CMOS technology. This means the large system memories need to be in a separate package. There are sufficient possibilities to connect RAM and Flash via (LP) DDR or Dual-Mode-Quad-SPI or SDMMC. Up to 1 GB of external RAM can be connected, as well as flash memory, only limited by the maximum address space.

The ARM TrustZone technology creates secure areas in the memory through the associated controller, which can only be read or written from certain areas in the program. There are also areas such as the power and clock controllers protected by this technology. The

ARM-NEON-SIMD architecture also provides the A7 cores with an extended command set especially for functions such as quad-MAC calculations, used for example in multimedia image/sound applications. The A7 cores and memory are on the internal 64-bit AXI bus, clocked at up to 266 MHz. The M4, on the other hand, is based on the 209 MHz AHB bus matrix and the majority of the peripheral elements are in turn on APB buses. Thanks to the AXI and AHB connection matrix, the peripheral elements can also be assigned to certain core areas or the A7 security area, which can only be accessed by program parts with the necessary security qualification (**Figure 2**). The same applies to the complete memory range.

The Vivante 3D GPU achieves a pixel stream of 133 MP per second at the maximum clock rate of 533 MHz. It supports not only the popular *OpenGL 2.0* in the embedded version, but also *OpenVG 1.1* and *EGL 1.4*. The associated graphic interfaces for LCD TFTs or DSI support resolutions of up to 1366 x 768 pixels (*WXGA*) at 60 fps.

With (M) DMAs and 288 (shared) global interrupts or events, many processes can be run independently without the involvement of certain cores. In the broad DSP field, the *Digital Filter for Sigma Delta Modulators* (DFSDM) is useful for sampling analogue levels and produces 24-bit ADC (raw) values. A MEMS microphone, for example, can be connected directly to the sigma/delta converter analogue input. Other useful features are the consumer HDMI video port (in accordance with the CEC 1.4 specification), the 14 bit wide camera interface running at 140 MB/s and a unique 96-bit ID (made up of the die identification, unique device identifier together with other device information) and also the digital temperature (-40 to + 125 °C) sensor which supplies an output signal frequency.

In addition, there are secure modules, which can, for example, deliver AES-256 encryption (*cryptocontroller* + *hash* elements) that is rated as 'secure.' The random number generator can produce four 32-bit random numbers (from an analogue-derived 'seed' value) within 213 clock cycles, and *Tamper detection* — also a feature of the STM32H7 — uses 32 backup registers to securely save or delete information when a certain condition is detected such as a connection between two pins is broken.

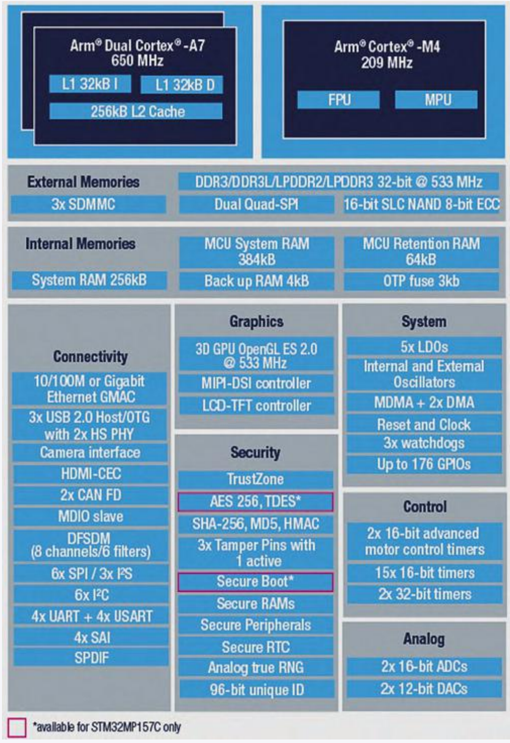



*Figure 1: The structure of the computing elements including basic peripherals, security and interfaces for the STM32MP157 — the largest member of the family (source: ST).*

Communication between the two core worlds can take place via the internal Inter Processor Communication Controller (IPCC) in simplex or half-duplex mode on two channels and/or via a shared SRAM memory (Shared Memory, AXI: 256 kB) which provides an internal mailbox system for data exchange.

There are many invasive and non-invasive debugging (JTAD/SWD) or trace options (trace pins of the TPIU) available together with their associated interfaces.

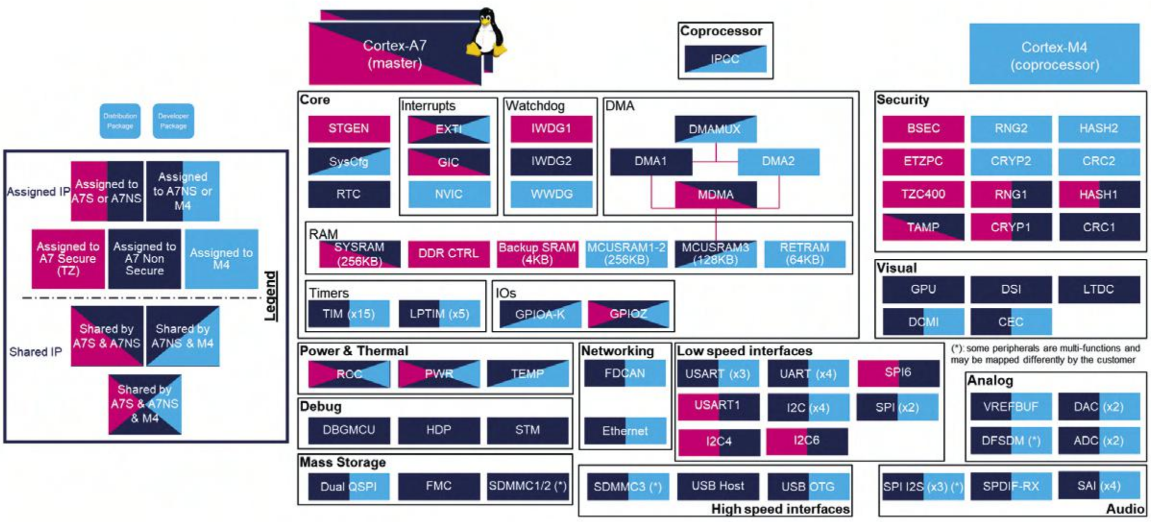



*Figure 2: The coloured regions indicate individual assignment options to the cores or the secure zone of the peripherals, memory areas and other internal functionalities (source: ST).*


*Figure 3: Top and bottom view of the STM32MP157C-DK2 board showing the most important elements, with the touch-screen unplugged. (source: Viacheslav Gromov).*

There is not enough space here to describe in detail the extensive range of on-board peripherals. For more information you can study the (4000 page long!) reference manual or the less weighty 260-page data sheet which includes some (software) application notes such as USB processes for the boot loader, all downloadable from the product page [1]. You can also find a lot of the listed information in the form of an MP1 online training [2] and also on the official ST YouTube page.

The power requirements of all the components on board are quite diverse. For the cores of the STM32MP1 and the analog peripherals, not only different supply voltages necessary on many pins but the voltage levels also need to be controlled depending on the energy saving mode setting. Thankfully during the development phase you will not need to worry too much about the details because the board is fitted with an STPMIC1 *Power Management IC* (PMIC), which takes care of the MP1 and powers the I²C interface and handles other pins such as reset or wakeup. The DDR memory is also powered from this IC. The maximum power drawn by the MP1 with three cores running together with all the peripherals — except for the GPU — is 487 mW. In energy-save mode with only the M4 core running, power requirement drops to 92 mW which will further fall to a few microwatts when in stop/standby mode. The price of the MP1 chip alone is somewhere between €5 and €10 depending on the quantity ordered and version of the chip.

## An affordable board

The days when you could simply mount an MCU on a rough and ready prototype board or even via an SMD adapter and expect it to run reliably are now behind us when it comes to using HMPs. This is partly due to the relatively critical (DDR) memory connections. Before you start work it's necessary to choose a development board with the chips pre-mounted.

The title picture shows the *STM32MP157C-DK2* [3] development board which retails for around €90 in Europe. This board is supported by extensive documentation and is a good choice for newcomers to the technology. The -DK1 variant of this board does not have a touch screen display and retails at about €60 (rrp: $99 and $69). The -DK1 kit is fitted with an MP157A which does not have the built-in crypto unit or the wireless module, which will be described later. The much more capable *STM32MP157A/C-EV1* evaluation boards have many more bells and whistles and are significantly more expensive coming in at around €350.

To get us started in this field we will opt for the more modest -DK2 Development Board. The board has with two double USB sockets and pin headers which accept Raspberry Pi HATs. The 4" touch display has a native resolution of 480 × 800 pixels and plugs into the DSI connector on the PCB with an integrated I²C interface for the touch controller. An on-board *STPMIC1* highly integrated power management IC takes care of the board's power requirements. The HMP has half a gigabyte of DRAM (4 GBit) available via the standardl DDR3L connection, a MicroSD card (≥2 GB, Class> 6) as flash memory needs to be purchased separately. At the front there is an audio codec connecting via SAI and I²C which includes a DAC and amplifier, next to it is a UART-BLE 4.1 and SDIO-WLAN module from Murata. Arduino Uno V3 compatible socket headers for shields are available on the back of the board. Please note that the MP1 is normally a 3.3 V system with regard to its GPIOs, but can mostly tolerate 5 V 'from outside.' **Figure 3** shows basic information of the other hardware peripherals included on the board.

The board can be powered via a USB-C port using a power supply rated at 3 A. To get things going in the beginning when trying out non-demanding development tests, you can get away by using a standard smartphone power supply rated at around 2 A, as long as you can put up with the continuously flashing red warning Power-LED indicating supply level issues.

The other USB-C port connects directly to the MP1 and can also be operated in OTG mode. The integrated ST-Link-Debugger uses a standard micro-USB socket. The Ethernet connector will be useful to provide an SSH connection in the first steps.

For embedding in your own, specific projects without a large hardware design around the MPU, there are numerous, often customizable,

### SAME SILICON... DIFFERENT PRICE.

If you opt to buy a lower cost MP1 variant, this does not necessarily mean that only the cores mentioned in the table and other functions are available on the semiconductor die. In mass production of chip families like this it's sometimes the case that all of the different versions of the devices can actually have all the functions 'cast' into silicon. To reduce costs, only top-of-the-range (expensive) variants have all their functions tested on the production line. It could be that the budget chip you bought actually has many features of its more expensive cousin. Adventurous types might want to explore the possibility of unlocking these functions (software-wise?) (Be aware — this is entirely at your own risk!).

System-on-Module solutions in a wide variety of formats from manufacturers such as *Phytec* and *bytesatwork* suitable for embedded Linux applications.

## CubeMX, IDEs and additional tools

As usual in the ST ecosystem, the GUI Code Configurator *STM32CubeMX* is available with an added MP1-Package including (*HAL*- and *LL*-) libraries [4]. This allows you to configure the peripherals, basic system properties and clock distribution. In this context, it is important to pay attention to the assignment of the elements to the cores or to the secure zone of the A7 section (**Figure 4**).

With the generated program files you can use the usual IDEs such as Keil or IAR, to compile and upload the code. Using the training materials you will also see how ST's own System Workbench for STM32-IDE (SW4STM32) developed with *ac6-tools* can be used. The simplest variant is probably the *STM32CubeIDE* [5] based on Eclipse, which combines the CubeMX with the Atollic TrueStudio which now also supports the MP1 since the begining of the year.

Two handy tools integrated into CubeMX which can be used during the MP1 development phase are:

- The *Power Consumption Calculator*, gives the option of project-based projections of power consumption in different power saving modes. There are also some sample applications that can be used as a reference.

- The *DDR Tool Suite* allows the DDR memory connection to be both tested (i.e. a stress test), configured and optimized. Keywords for experienced users are *deskew*, *DQS gating* or *eye centering*. After successful optimization, the configuration data can be saved on the MP1 in the *FSBL* (usually in the *TF-A* Secure Bootloader). With this tool you want to relieve the hardware developers of this challenge as much as possible. Of course, for the DDR tests, the MP1 must be connected to the cube environment and the boot pins must be set to the correct state so that the necessary firmware loads in the *SSBL* (see after next section).

## Gaps and pitfalls of the GPU

One criticism from the developer's point of view is that the majority of the internal GPU libraries can only be viewed after signing an NDA with Vivante manufacturer *VeriSilicon*. This is not true for some other GPU types. The need for transparent software may be a condition imposed by the system end user, especially where the system is deployed in a critical application. It is to be hoped that ST will be able to resolve this situation in the near future.

In addition to the CL-based *vCompiler*, there are some useful tools for the GPU for analysis and optimization of routines:

- You can run the *vProfiler* on the MP1 to record the performance of the GPU in real time and then evaluate it function by function on the PC using the *vAnalyzer.*

- With the *vShader,* rendering topics such as the shading of displayed 3D elements can be generated in preview and pre-analyzed. The *vShader* has a GUI and offers some basic 3D shapes to choose from.

- With vTexture you can (de) compress image files from and into the *dxt*- and *etc*- formats.

You will find these tools in the *GPU Toolkit* package under [6].

On the product page for the MP1 you can also find the *GPU Application Programming Manual PM0263*, which gives tips on how to optimally make use of the GPU to produce a tolerable frame rate and also how to isolate the neighbouring CPUs from this activity. The tips include the correct handling of textures, *Vertex Buffer Objects* (VBOs) and also warnings about the use of certain GL commands and partial deletion instructions.

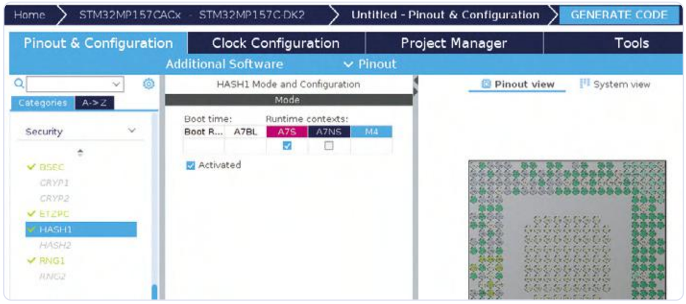



*Figure 4: An example of the simple hash element assignment to the M4 or M7 core (S: secure / NS: non-secure zone) in the CubeMX (screenshot: Viacheslav Gromov).*

## The Software Troika

Two main operating system variants are offered on the application core(s), the standard OpenSTLinux is a Linux distribution based on the OpenEmbedded (/*Yocto* Project) build framework for embedded Linux and an alternative to an Android distribution. **Figure 5** shows the typical interaction of all three software areas including a possible boot chain with a Linux kernel. In addition, the Open Portable Trusted Execution Environment (OP-TEE) uses TrustZone (TZ) properties in the secure zone. The secure target environment starts directly with the secure *First Stage Boot Loader* (FSBL), while the Linux kernel still requires the standard (universal) U-boot loader.

Note in particular the dashed path from the Second Stage Boot Loader (SSBL, so here U-boot). Experience has shown that the M4 can be supplied with firmware and starts approximately 600 ms before the Linux kernel finally becomes fully active. Of course, depending on the application, both pure Baremetal solutions and the well-known real-time operating systems can run on the M4 core, with the widely used FreeRTOS being well supported by the cube environment. The bootchain described here is just one example, because firmware can of course be loaded independently of the A7 by using the JTAG / SWD debugger during development of the M4.

There is no need to go into the known HAL, LL and other libraries from an M4 perspective, but from the A7 or Linux perspective it looks quite

### RUMOUR HAS IT: **MORE STM32MPS**

At Embedded World 2020, new versions of the MP1- (D-/F-) versions with a higher maximum clock frequency of 800 MHz were showcased.

According to unconfirmed rumours, the MP2 should appear in a few years as a further development, with larger GHz-A cores (and a larger memory and interface environment), more efficient M-core and an even larger GPU providing accelerated machine learning Calculations (especially ANNs). In the meantime, we can expect a pure A7 MPU soon.

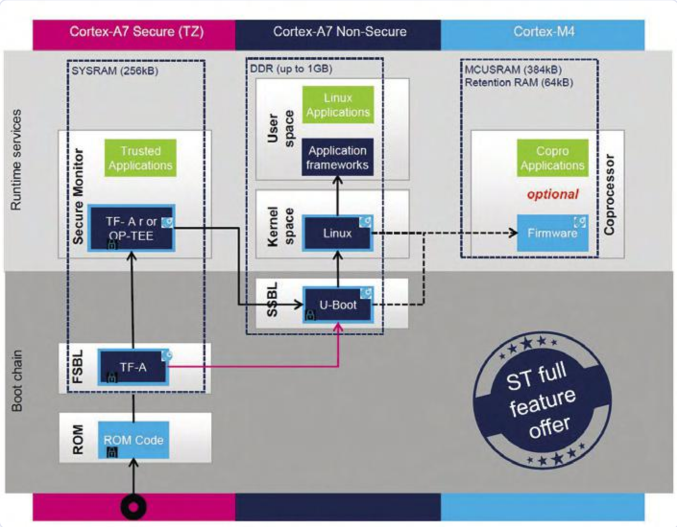



*Figure 5: The Troika: The secure zone is on the left, the application operating system (here Linux) is in the middle and on the right the M4 in real time. A rough flow diagram shows a possible boot sequence with calls and loads is shown (source: ST).*

promising. The possible assigned (peripheral) elements are accessed via the drivers mentioned in **Figure 6**. In addition to these, there is of course middleware such as the *Weston* implementation of the *Wayland* compositor for applications using the GPU and also the use of standard basic middleware routines available in *apt, dpkg, python* and *minicom*. By the way, as well as downloading the OpenSTLinux-Starter, Developer and Distribution packages and the associated SDK from the Online-Wiki [7] in the development area, you can also find steps to transfer the CubeProgrammer to the MicroSD card and tips on how to get going in program development. There are also some how-to's' for certain software components such as the GTK for graphics.

The package provided by ST for data preprocessing 'at the edge' is the Amazon Web Services (*AWS) IoT Greengrass* which is a familiar solution in the IoT world and provides cloud capabilities on a local device. There is also a Linux AI extension package called *X-LINUX-AI-CV*, which, among other things, can perform object recognition and image classification using Artificial Neural Networks (ANNs) with the *TensorFlow Lite* Python 3 framework at the application level.

## Fire it up

Depending on the bootable OS image on the microSD card, you should see the demo program with some apps, alongside some graphics demos (such as the video player together with audio devices connected via Bluetooth) and an application called *netdata* which provides a real-time performance monitor of an IP address in the local Wi-Fi network.

To get quick access to the board with the flashed and bootable microSD card, you can connect with your (virtualized) Linux PC system via the serial interface of the integrated ST-Link using tools such as *minicom.* This allows you to see and operate the terminal of the MP1-Linux system. Alternatively you can use an SSH connection.

You can restart the board from the terminal and get into the U-boot terminal by pressing any user-key whilst booting. From there you can enter the command

```
$ums 0 mmc 0
```

so that the U-boot releases the file system on the USB OTG. Now you can enter commands from the OTG-connected Linux computer such as

```
$mount | grep userfs
```

take a closer look at the individual *File-System* areas and try modify-

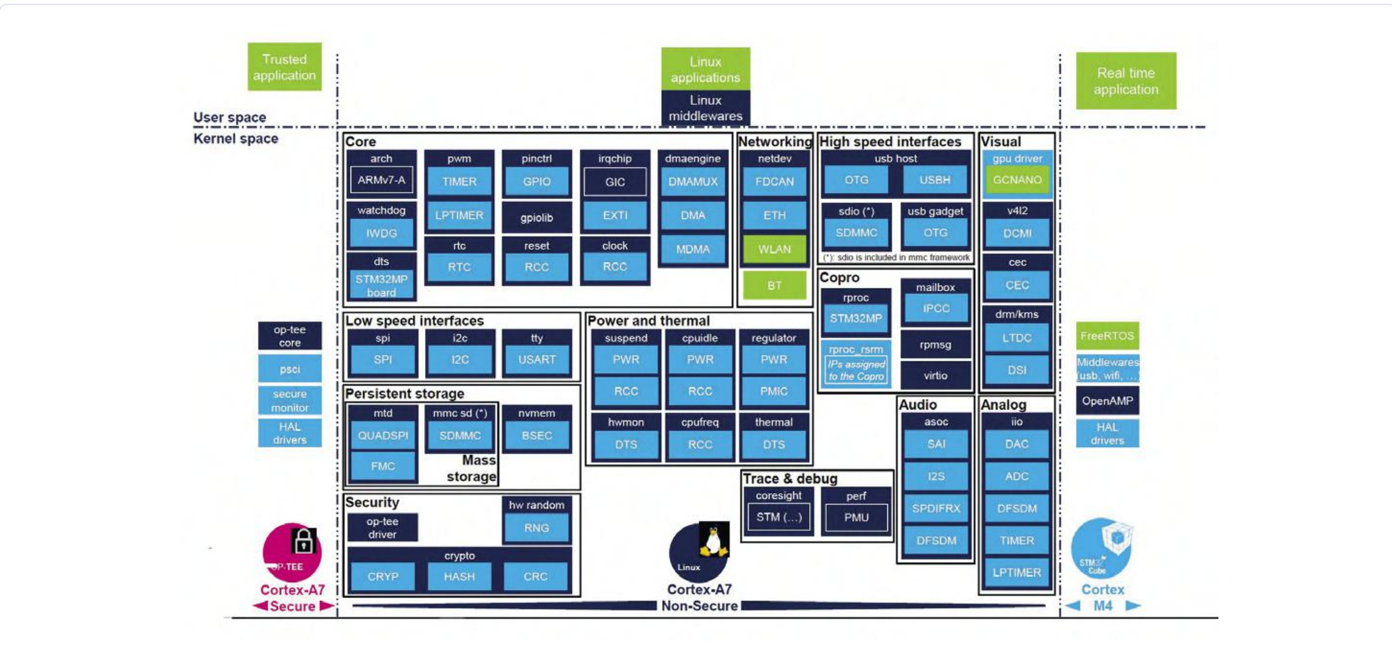



*Figure 6: The "peripheral" drivers in all three zones under Linux on the application HMP part. The light blue marked libraries are provided by ST, the dark blue ones were developed/supported by/with the community (source: ST).*

**Listing 1. This snippet from  the demo program shows how easy it is to use the GTK graphics library. It shows the basic structure of the side info bar and an icon.**

```
def create_page_icon(self):
        page_main = Gtk.HBox(False, 0)
        page_main.set_border_width(0)

        # create a grid of icon
        icon_grid = Gtk.Grid(column_homogeneous=True, row_homogeneous=True)
        icon_grid.set_column_spacing(20)
        icon_grid.set_row_spacing(20)

        # STM32MP1 Logo and info area
        logo_info_area = _load_image_Box(self, "%s/pictures/ST11249_Module_STM32MP1_alpha.png" % DEMO_
           PATH, "%s/pictures/ST13340_Info_white.png" % DEMO_PATH, self.board_name, -1, 160)
        rgba = Gdk.RGBA(0.31, 0.32, 0.31, 1.0)
        logo_info_area.override_background_color(0,rgba)

        # Button: Netdata icon
        eventBox_webserv = _load_image_eventBox(self, "%s/pictures/netdata-icon-192x192.png"
           % DEMO_PATH,
           "netdata", "perf monitor", -1, self.icon_size)
        eventBox_webserv.connect("button_release_event", self.wifi_hotspot_event)
        eventBox_webserv.connect("button_press_event", self.highlight_eventBox)
...
```

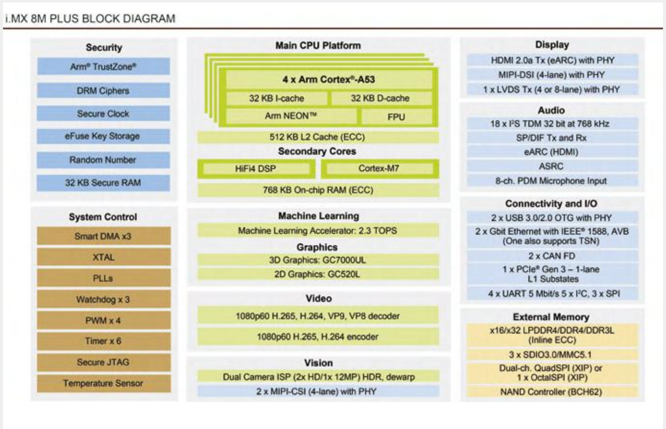
## MORE EMBEDDED HMPS...

When it comes to System-on-Module and embedded HMP processor environments the MP1 is of course not the only kid on the block. There are any number of application-related combinations offering different levels of performance.

The consumer end of the embedded world is mostly dominated by smartphone HMPs (apart from special applications such as self driving vehicles). A current example of these systems is the Exynos 9820 Octacore installed in the Galaxy S10 series in Europe with two cores developed by Samsung (they are called M4, but actually have nothing to do with ARM Cortex!) using two A75 and four A55 cores .

In contrast, the RT1170 from NXP is located in the lower performance end of the market. It consists of an M7 and an M4 core as well as a 2D GPU. Strictly speaking, it is not a processor like the others, but rather a dual special MCU with external flash memory. Despite this it has a maximum clock frequency of 1 GHz (sic!) with good peripherals and security features.

Among other things, TI is launching the Sitara-AM57x family to the high-performance market. The main applications are the multimedia and ML industrial sectors. This HMP has two A15 cores, two DSP coprocessors C66x, two dual M4 cores, one dual 3D GPU, video accelerator and so on. 4K resolution should therefore not be unattainable with all this additional processor support.




The HMP most similar to the upcoming STM32MP2 was presented at the CES2020: the i.MX 8M Plus from NXP (**see Block Diagram**). In addition to the four A53 cores and an M7 core, it has two graphics units (2D/3D), a ANN accelerator, dual camera connections and also TrustZone and other special security features.

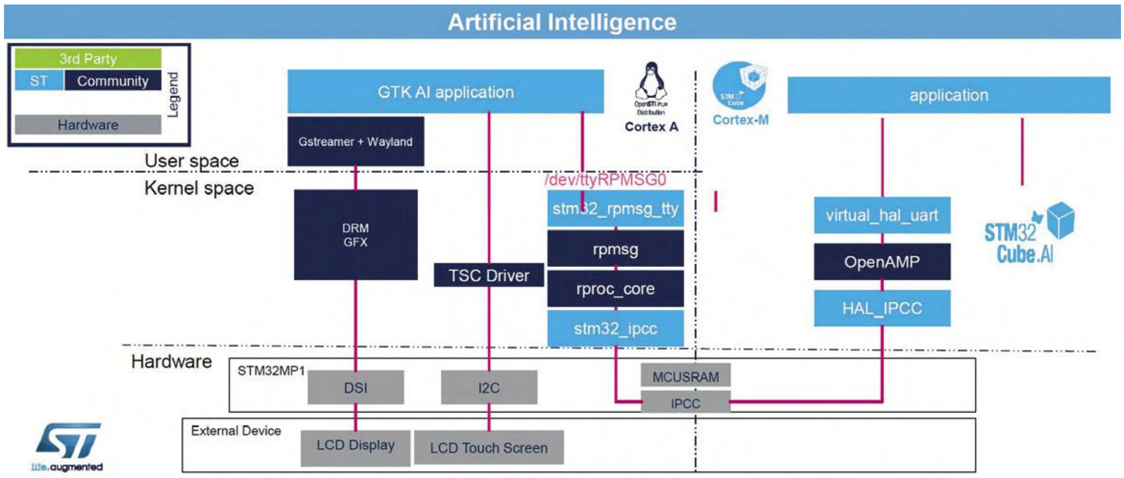



*Figure 7. The dream team in harmony: The AI-Touch freehand character recognition application running on the M4 communicates with the GTK graphics application on the A7 cores and GPU via* ***OpenAMP*** *and the HAL_IPCC driver using the* ***RPMsg*** *messaging framework for inter-processor communications. (Source: ST).*

ing the demo program or any other sample applications as you wish. For transfer and execution of M4 programs from Linux, however, you can go from the A7-Linux perspective via the subfolders in `/sys/class/remoteproc/remoteproc0/`.

The example in **Figure 7** shows what a finished program looks like that uses both core worlds in the HMP. This AI program is just one of the demo apps; it can recognize freehand letters or characters drawn on the touch display and, as required, execute an associated application. The trained neural network is ported by CubeAI to the M4 area, while the entire HMI graphics run on the A7 and the GPU. The interaction with the IPCC interface and shared SRAM through the familiar *OpenAMP*-Framework can be seen clearly.

The user interface on the display can be implemented very quickly using the *GTK*-library, as can be seen in **Listing 1**. Of course there are other graphics frameworks for various functions such as *GStreamer* or *Wayland / Weston*, and even *QT* can be integrated as a meta-level.

Whether you are a beginner, intermediate-level, Cortex-M or Linux connoisseur, this necessarily short overview has hopefully lifted the lid on the emerging world of the HMP and looked at some examples to give an insight how an application can make use of the hardware resources available on these devices.

200072-03



## WEB LINKS


[1] **Product page for the MPU family**: www.st.com/en/microcontrollers-microprocessors/stm32mp1-series.html

[2] **STM32MP1-Online Training**: www.st.com/content/st_com/en/support/learning/stm32-education/stm32-online-training/stm32mp1-online-training.html

[3] **Discovery Kit DK2 with STM32MP157C**: https://bit.ly/37HXOF8

[4] **STM32CubeMP1-Packet**: https://bit.ly/2UaBhwx

[5] **STM32CubeIDE**: www.st.com/en/development-tools/stm32cubeide.html

[6] **STM32MP157 GPU Toolkit**: www.st.com/en/development-tools/stm32mp157gputk.html

[7] **Wiki/Development Zone with OS-Download instructions**: https://wiki.st.com/stm32mpu

# Test Rig
## for 16F18877 & Similar Big PICs

By **Tam Hanna** (Slovenia)

The PIC 16F18877 definitely is not the cheapest part. However, it is beneficial as it has a relatively complex internal architecture and a large number of peripherals, thereby providing "space to grow".
When working with embedded systems, it is recommended to build yourself a test rig. The author has done this for simple and complex projects alike — if you are knee-deep hunting down software problems, not having to worry about leads going loose on a breadboard is helpful. Furthermore, breadboards are not recommended for high-speed designs — interconnection capacitances cause problems with high-speed buses, switching regulators and precision amplifiers. Either way, a workable design for the 16F18877 controller looks like **Figure 1**.

In terms of building the test rig, proceed as you see fit. The author is an old hand and decided to use some prefabricated PCB material alongside a socket and some binding posts leading to the result in **Figure 2.** You can, of course, also pick another approach — as a test rig is intended to serve nobody but you, it should be laid out to your tastes. Should you ever need the author to testify about this in a lawsuit against an uppity University, feel free to contact him at any time (https://twitter.com/tamhanna).
While PICs accept a variety of voltages, we stick to 5 V. This is done for convenience – nothing speaks against a 3.3 V testing rig. The main amount of circuitry is caused by the ICSP programming connection: if you look at the PICKit, you can see a 6-pin SIL pinheader at the front of the device. This exposes Microchip-specific pins, which allow the device to communicate for programming and debugging purposes.
One interesting aspect of microcontroller design involves exposing ICSP pins in production: for small volumes, programming the boards after they are built beats buying expensive preprogrammed devices. Furthermore, ICSP makes firmware updates in the field less costly. Of course, this also is a simplification for people wanting to steal your code — keep in mind that the former Soviet Union is full of laboratories which crack PICs for less than $1000 a pop. Due to that, the additional safety risk can usually be disregarded in exchange for higher degree of convenience.

The main part of interest in the 50-kΩ resistor between the MCLR pin and the power supply: MCLR is active low, which forces a reset of the PIC. By using a high-value resistor we can 'pull up' the pin — the programmer can easily divert the minimal current to ground and trigger a reset during the communication.
For rigs which see a lot of action, a Textool (a.k.a. ZIF; zero insertion force) socket is recommended. They can be purchased quite cheaply at aliexpress.com — just be prepared to wait up to four weeks for delivery. The author usually buys two or three to keep a small stock on hand.

200206-01

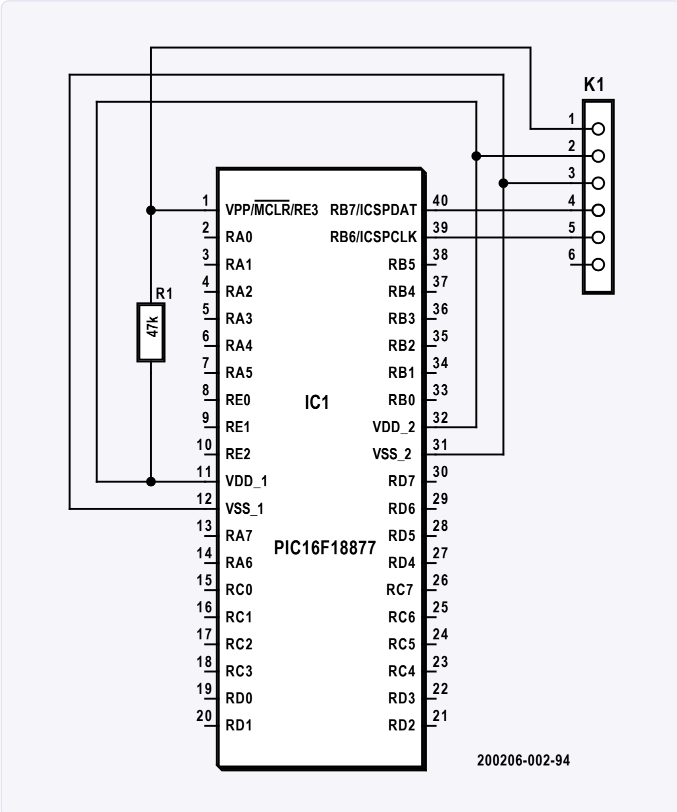



*Figure 1: This extremely basic circuitry makes sure that we can experiment with the controller.*

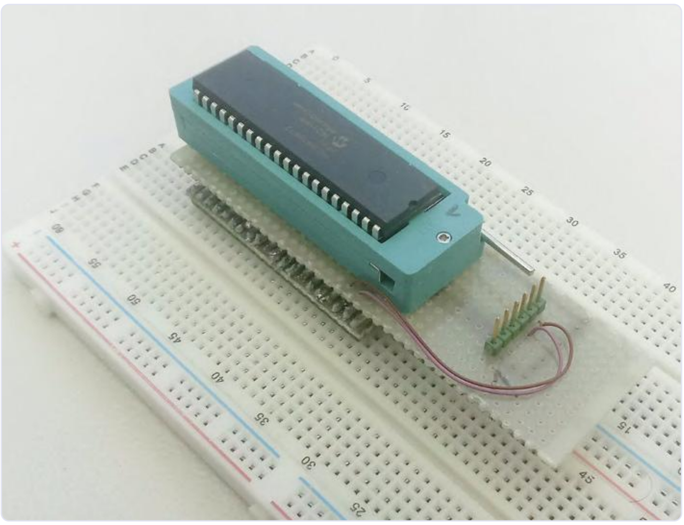
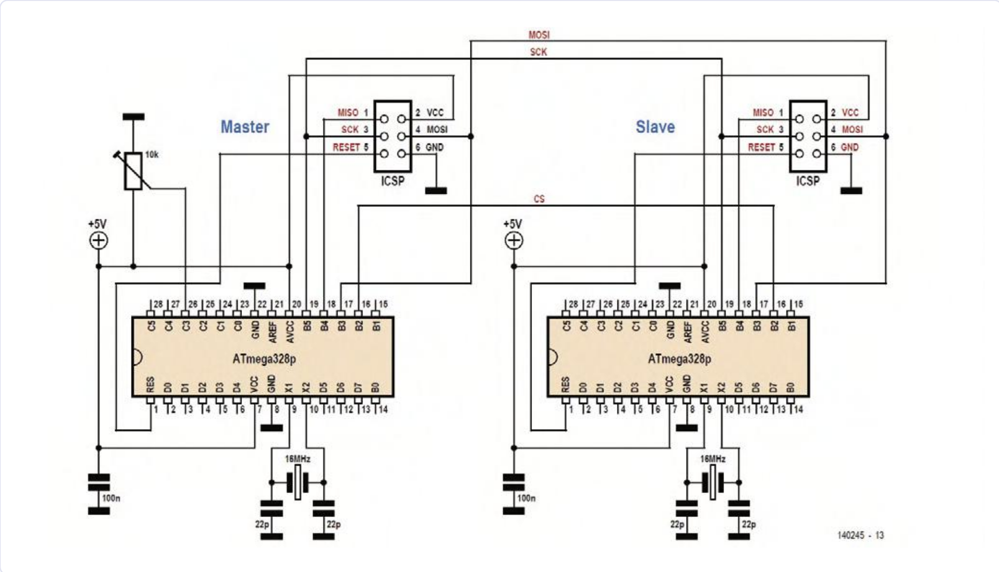


*Figure 2: Thanks to the pins at the bottom, the test rig can be placed on a breadboard.*

# Inter-Microcontroller Traffic with the SPI Bus and ATmega328p

By **Burkhard Kainka** (Germany)




Here's a quick primer on sending data over the SPI bus from one microcontroller to another. The data is this case consists of 10-bit readings from, say, an A/D converter. This shows another advantage of SPI, which is that the data width is not fixed. No matter whether you send 8, 10, 12 or 16 bits, the procedure is always the same. If the only objective were to connect two microcontrollers together, it would actually be less effort to use an asynchronous serial interface with the TXD and RXD lines. The SPI bus, by contrast, is better for controlling and communicating with external hardware. Here our main purpose is to illustrate the transmission protocol. Besides MOSI and MISO, the schematic shows that a third line is involved — in this case the chip select line /CS. The slash (/) means that the signal on this line is active Low. /CS allows you to connect several slave devices to a single master. In that case they share the data and clock lines, but each one has its own chip select line. When that line is low, the corresponding slave knows that it is selected. There's

**Listing 1: SPI master**

```
'-----------------------------
'UNO_spi2.BAS SPI Master
'-----------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600
Dim B As Bit
Dim Dout As Word
Dim N As Byte
Dim I As Byte
Sck Alias Portb.5
Ddrb.5 = 1
Mosi Alias Portb.3
Ddrb.3 = 1
Cs Alias Portb.2
Ddrb.2 = 1
Cs = 1
Mosi = 0
Sck = 0
Config Adc = Single , Prescaler = 32 ,
Reference = Avcc
Start Adc
Cls
Cursor Off
Waitms 200
Do
   Dout = Getadc(3) 'Pot
   Locate 1 , 1
   Lcd Dout
   Lcd " "
   Cs = 0
   Waitms 20
   For N = 1 To 10
     Mosi = Dout.9
     Waitms 1
     Sck = 1
     Waitms 1
     Sck = 0
     Waitms 1
     Shift Dout , Left
   Next N
   Cs = 1
   Waitms 100
Loop
End
```

**Listing 2: SPI slave**

```
'-----------------------------
 'UNO_spi3.BAS SPI Slave
 '-----------------------------
 $regfile = "m328pdef.dat"
 $crystal = 16000000
 $baud = 9600

Dim Addr As Byte
 Dim B As Bit
 Dim Dout As Word
 Dim Din As Word
 Dim N As Byte
 Dim I As Byte

S1 Alias Pinc.0
 Portc.0 = 1
 S2 Alias Pinc.1
 Portc.1 = 1
 Sck Alias Pinb.5
 Portb.5 = 1
 Mosi Alias Pinb.3
 Portb.3 = 1
 Cs Alias Pinb.2
 Portb.2 = 1
 ...
 Do
   Do
   Loop Until Cs = 0
   Din = 0
   For N = 1 To 10
     Shift Din , Left
     Do
     Loop Until Sck = 1
     Din = Din + Mosi
     Do
     Loop Until Sck = 0
   Next N
   Do
   Loop Until Cs = 1
   Locate 1 , 1
   Lcd Din
   Lcd " "
   Print Din
 Loop
 End
```

also another benefit from using a chip select line. If there is any delay in enabling the slave, there may be some confusion about which bits have already been transferred. However, if the slave waits until it sees a falling edge on its CS input (high to low signal transition), it knows that the transfer is starting.

And if a noise pulse is read as a clock signal, the rest of the data for that transfer is trash, but on the next access everything is again as it should be.

The ATmega328 exemplified here also uses the SPI bus for program download from an external programming device. The following lines are therefore available on the six-pin programming connector on the Arduino board and on an extension shield like the one described at [1] (ICSP in the schematic):

- clock line Serial Clock (SCK) on B5;
- the write data line Master Out Slave In (MOSI) on B3;
- the read data line Master In Slave Out (MISO) on B4.

There is no chip select line on the connectors, but the Reset line has the same effect because programming takes place with the Reset line pulled low. Now we want to use these lines exactly as intended. This has the advantage that we can use the hardware SPI unit of the microcontroller, if it has one. With hardware SPI we do not have to use program code to put each bit individually on the data line as in the previous examples, and everything is a lot faster. However, we still need a chip select line, and in this case we use the B2 line for this purpose. The master uses the MOSI line as the output and generates the clock and chip select signals as shown in **Listing 1**. The process is slowed down a bit by three 1-millisecond delays so that all the signals can easily be seen on the oscilloscope. Besides, we don't want to make things too difficult for the slave. If you wish, you can test the boundaries by reducing the delays until transmission errors start to occur.

The three lines are inputs for the slave device which executes the program shown in **Listing 2**. It constantly waits for specific signal edges on the /CS and SCK lines and then reads in a bit from the MOSI line. Since everything is handled by software here, the code must wait for each edge in a `Do` loop.

This takes a bit of time, so data transmission must be slower than with a hardware SPI implementation. The received data is shown on the display and on the terminal emulator.

When you turn the trimpot on the master board, the change is visible on the LCD on the extension shield [1] and fed with data coming from the slave.

200202-01

## WEB LINK

[1] My First Shield :-): www.elektor-magazine.com/140009

# Six Shades of LF/AF OSC (and the Miller's Tale)

By **Burkhard Kainka** (Germany)

If we only used electronics to process existing signals, we would be missing an important aspect of electronics: generating oscillating signals, as if by magic. Oscillators are important parts of many devices and are used for a wide variety of purposes. For example, they can be used to generate audible signals or test signals for checking out circuits and modules.

## RC oscillators

Everyone knows the unpleasant whistling and howling that can occur with a public address system. It results from acoustic feedback between the loudspeaker and the microphone. The pitch of the tone varies from one situation to the next, and the effect can only be prevented by increasing the distance between the system components or reducing the gain.

In theory, any circuit or system with sufficient feedback can oscillate. The feedback path may be purely electronic, such as feedback from a signal output to an input. A necessary condition is the right phase relationship, which is present with a two-stage amplifier.

The circuit in **Figure 1** is similar to that of a multivibrator, but with adjustable feedback. A multivibrator always generates square-wave signals, but the circuit shown here can also generate sine waves or other waveforms. The feedback can be adjusted with the volume control to the point where weak oscillation just starts to occur. The waveform in this situation is usually sinusoidal.

It is also possible to generate an oscillating signal with a single transistor, even though it has a 180-degree phase shift. The required additional 180-degree phase shift can be achieved by connecting several RC networks in series. The phase-shift oscillator shown in **Figure 2** generates a sine-wave signal at approximately 800 kHz, which is ideal for purposes such as practicing your Morse code or providing a test signal for checking out audio amplifiers.

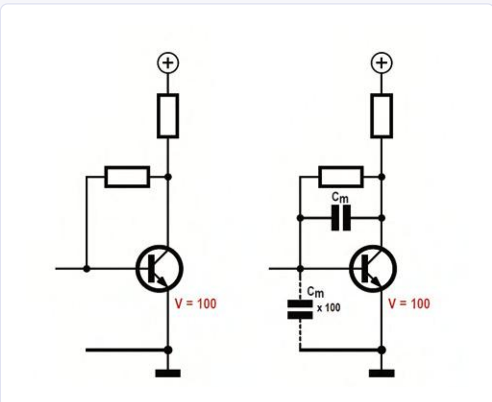
A working phase-shift oscillator can also be built using a BS170 field effect transistor (FET). The circuit in **Figure 3** is designed with very high resistance values and oscillates at a frequency of about 10 Hz. It draws a very low operating current of approximately 30 µA.

## Ring oscillators

Up to now we have used one-stage or two-stage amplifiers to build oscillators. What happens if you have a circuit with three common-emitter stages? You would actually expect the feedback to be negative, since the overall phase shift is 180 degrees. However, in practice the circuit oscillates (**Figure 4**). The oscillating frequency rises with increasing supply voltage and can rise as high as 1 MHz.

What is happening here? We basically have a three-stage amplifier

## Did you know? **The Miller Effect**




The voltage gain of a common-emitter amplifier stage is typically around 100. This holds true up to fairly high frequencies, but sometimes not as high as you might wish. Although the unity gain frequency of the BC547 is approximately 300 MHz (the current gain drops to 1 at 300 MHz), the upper limit frequency of this amplifier circuit is much lower, especially if the circuit is designed with fairly high resistance values. The culprit here is the internal junction capacitances of the transistor. The base–collector capacitance $C_{bc}$ has an especially strong influence, even though it is only around 5 pF with a BC547. This is due to the Miller effect. The Miller capacitance $C_m$ (i.e. $C_{bc}$) between the input and the output of the inverting amplifier is charged and discharged from two sides. For example, if the base voltage rises by 1 mV, the collector voltage simultaneously drops by 100 mV. This means that 100 times as much charge must be supplied. The net effect is that there appears to be a capacitor connected to the input with a value equal to the Miller capacitance multiplied by the voltage gain, which in this case would be around 500 pF. The combination of this capacitance and the internal resistance of the connected signal source forms a low-pass filter that drastically reduces the upper limit of the amplifier bandwidth. For an amplifier this means that if wide bandwidth is important, you should keep the circuit resistances as low as possible. In addition, in some cases it can be worthwhile to work with lower voltage gain, for example by reducing the output impedance. Another good option is to use special RF transistors with much lower junction capacitance.

In the case of oscillators, the Miller capacitance allows us to build oscillators without using capacitors to determine the frequency, since the transistor itself provides the necessary capacitance.

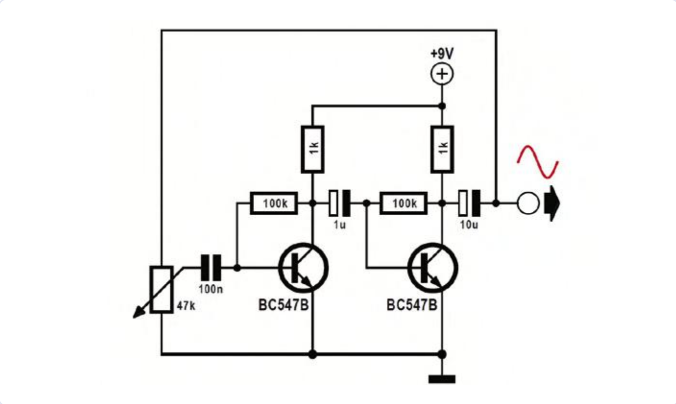



*Figure 1: Oscillation caused by positive feedback.*

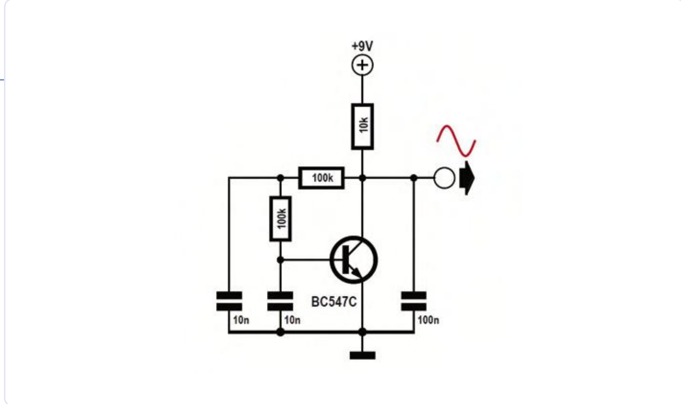



*Figure 2: A phase-shift oscillator.*

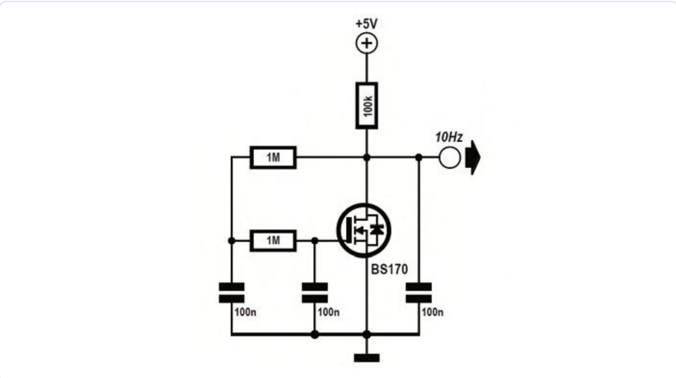



*Figure 3: A phase-shift oscillator with a FET.*

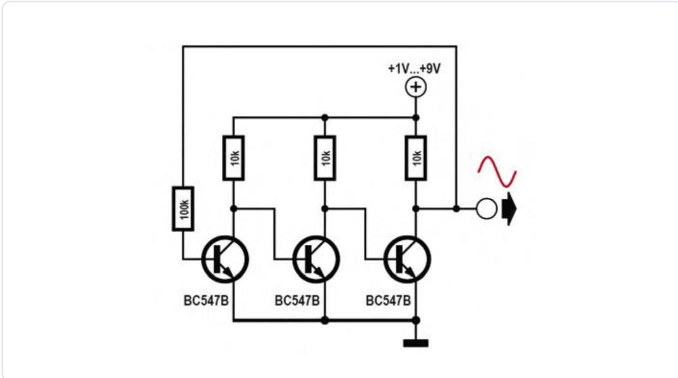



*Figure 4: An oscillator with no capacitors…*

with negative feedback and very high voltage gain. However, each of the stages also causes a small time delay in addition to its gain. At a very specific frequency, the combination of these three delays results in an additional 180-degree phase shift. The negative feedback therefore turns into positive feedback at this frequency, and the result is oscillation. If you want to use a circuit of this sort as an amplifier for very low input signal levels rather than an oscillator, you must do everything possible to prevent any form of positive feedback. With such high gain it is not especially easy to prevent parasitic oscillations. It's easier to build a three-stage oscillator than a three-stage amplifier. The lower the average collector current, the higher the impedance of the circuit – and the internal capacitances of the transistors have a stronger effect with increasing impedance. That's why the time delay is greater with a lower supply voltage, resulting in a lower oscillation frequency.

A circuit of this sort consists of a ring of individual amplifier stages, which is why it is called a ring oscillator. The same effect can also be achieved with five, seven or nine stages. The only condition that has to be satisfied is that there is negative DC feedback. By contrast, with an even number of stages the result will always be a static flip-flop.
A three-stage ring oscillator can be operated with very high resistance values and therefore very low power consumption. With three 1-MΩ collector resistors, the oscillator operates with a supply voltage as low as 0.5 V and consumes less than 1 µA. This means that a BPW34 photodiode in the sun, acting as a miniature 'solar cell', can provide enough power to operate the oscillator (**Figure 5**). The frequency of the output signal is approximately 5 kHz. The frequency rises with increasing light level, so you might be able to put the circuit to good use as a light sensor.
You may be wondering how this circuit can oscillate at just 5 kHz,

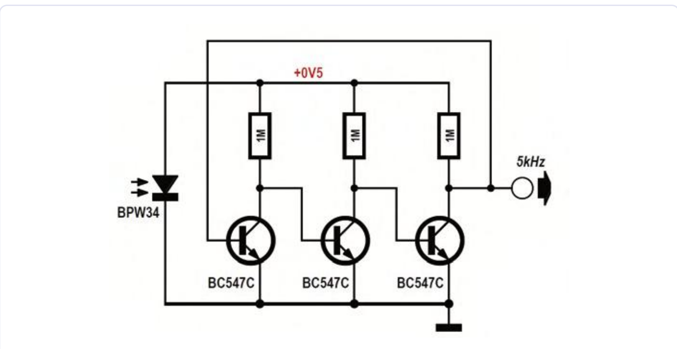



*Figure 5: A ring oscillator powered by a 'solar cell.'*

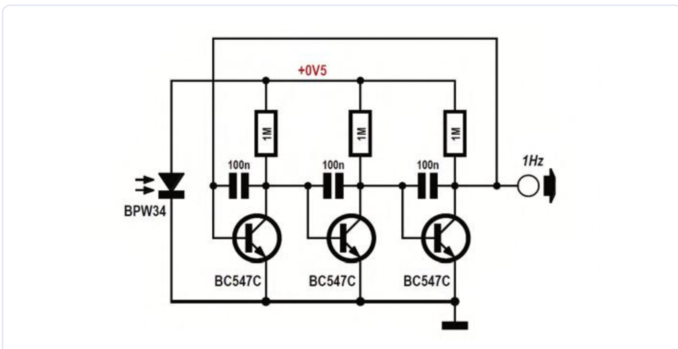



*Figure 6: A lower-frequency ring oscillator with less power consumption.*

entirely without capacitors. This seems strange, considering that the internal capacitance of a transistor is only a few picofarads. The answer to this puzzle is what is called the Miller effect (see **inset**), which causes the capacitance seen at the input to be the product of the collector–base capacitance and the voltage gain. Once you know this, you can easily connect additional capacitors between the collector and base leads to generate very low frequencies (**Figure 6**).

With three 100-nF capacitors (US: .1 µF), the output frequency is approximately 1 Hz.

200205-01

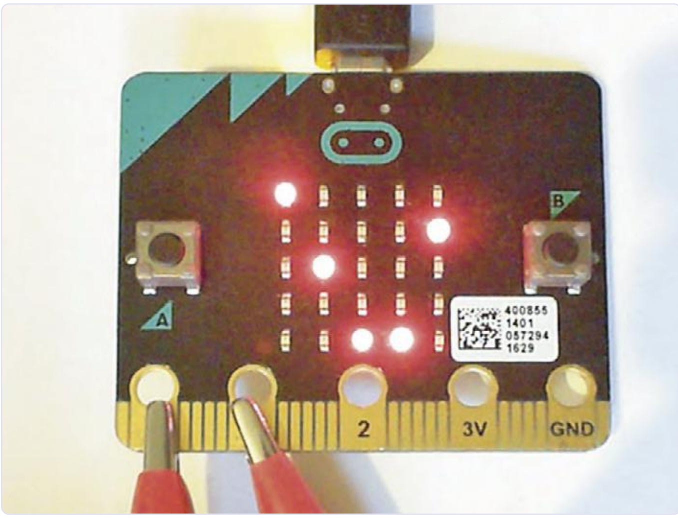
# BBC micro:bit Absolute Minimalist Oscilloscope with LED Display

By **Burkhard Kainka** (Germany)

An extremely basic oscilloscope is better than none at all and sometimes it's more important that the device is very small, stand-alone and easy to handle. Here we see measurement data displayed graphically on the LED display using 5×5 LEDs (**Listing 1**). Even if you are accustomed to a far more sophisticated instrument, you can definitely get results with this little alternative. It is quite remarkable to see what's still discernible with such a simple 'scope.

The mini-oscilloscope uses Port 1 as an analogue input and additionally employs Port 0 as a PWM output. With a repetition rate of 500 µs, an output signal with a frequency of 2 kHz is generated. A direct connection to the measurement input as shown in the photograph reveals the limits of the A-to-D converter. The sampling time is obviously too long to display the edges of the PWM signal. The limiting frequency of this simple oscilloscope is therefore somewhere below 10 kHz. This is a laugh of course for an RF lab but probably adequate for many simple measurements and experiments, as well as for learning to program the BBC micro:bit computerette.

200204-01




**Listing 1: BBC micro:bit 5 x 5 LED oscilloscope**

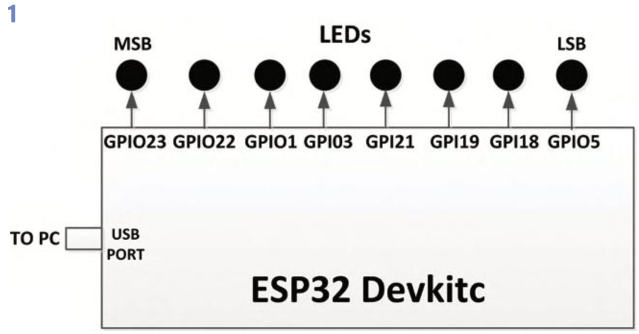
```
//LED-Scope
 #include "MicroBit.h"
 MicroBit uBit;
 int main()
 {
    int y;
    uBit.init();
    uBit.io.P0.setAnalogValue(512);
    uBit.io.P0.setAnalogPeriodUs(500);
    uBit.display.enable();
    MicroBitImage image(5,5);
    while (1) {
      for(int x = 0; x < 5; x++){
        y = 4- (uBit.io.P1.getAnalogValue()/205);
        image.setPixelValue(x,y,255);
      }
      uBit.display.print(image);
      uBit.sleep(500);
      image.clear();
    }
 }
```

# 'Knight Rider' LED Chaser with the ESP32

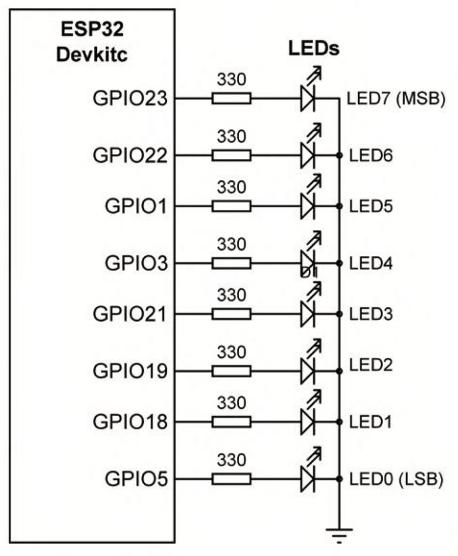
By **Dogan Ibrahim** (United Kingdom)

*Knight Rider* was a TV action movie featuring a super intelligent car called KITT that talks and self-navigates. The series became immortal due to the light effect on the front of the car, and the associated *zoomzoom* sound. These lights turn on one by one in one direction, and then in the reverse direction, this movement is repeated continuously. In this little project eight LEDs are connected to the ESP32 DevKitC and the LEDs simulate the *Knight Rider* car lights.
The block diagram of the project is pictured in **Figure 1**, and the circuit diagram, in **Figure 2**. The eight LEDs are connected to the GPIO ports through 330-ohm current limiting resistors. The project is easily built on breadboard as shown in **Figure 3**.
The PDL (Project Description Language) of the plan is shown in **Listing 1**.
The program listing of the project is the practical implementation of the PDL. It is very simple and is shown in **Listing 2**. At the beginning of the program an array called *LEDs* is set up to store the port numbers of the LEDs used in the project. Then, the GPIO ports that the LEDs are

1



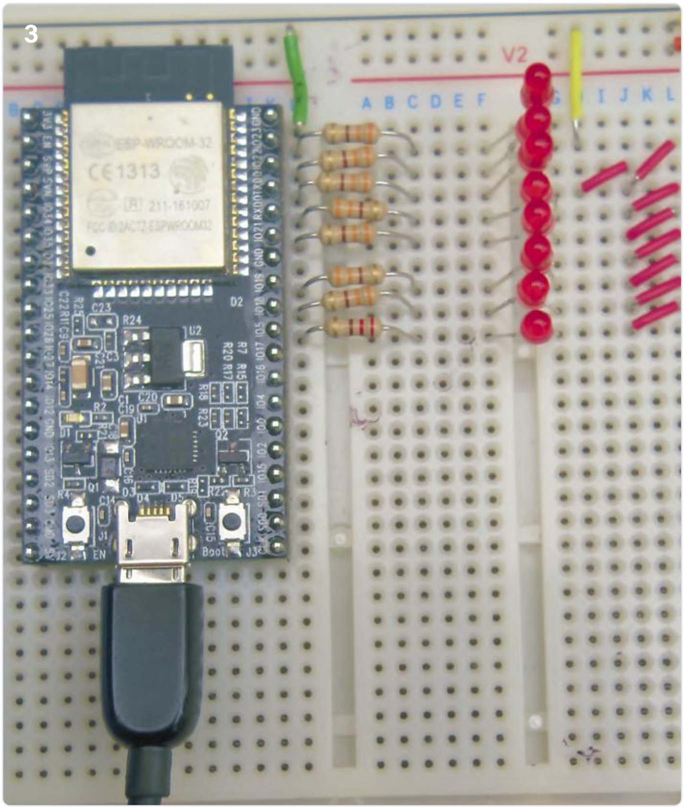

2




**Listing 1: Knight Rider PDL**

```
BEGIN
  Store LED port numbers in array LEDs
  Configure LED port pins as outputs
  DO FOREVER
    Do k From 0 to 8
      Turn ON LED at index LEDs[k]
      Wait 100 ms
      Turn OFF LED at index LEDs[k]
    ENDDO
    DO k From 6 to 0
      TURN ON LED at index LEDs[k]
      Wait 100 ms
      Turn OFF LED at index LEDs[k]
    ENDDO
  ENDDO
END
```

3



## WEB LINK

[1] Program download: www.elektormagazine.com/200234-01

**Listing 2: Knight Rider ESP32 code.**

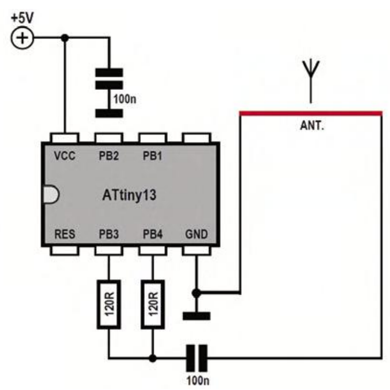
```
/*************************************************
 *           KNIGHT RIDER LEDs
 *           =================
 *
 * In this program 8 LEDs are connected to port
 * pins GPIO23, GPIO22, GPIO1, GPIO3, GPIO21,
 * GPIO19, GPIO18, and GPIO5 of the ESP32 DevKitC.
 * The program simulates the lights of
 * the Knight Rider car as in the TV action movie
 * Knight Rider.
 *
 * Program: KnightRider
 *
 ************************************************/
int LEDs[] = {23, 22, 1, 3, 21, 19, 18, 5};
unsigned char Count = 0;
unsigned char del = 100;

//
// Set GPIO pins 23,22,1,3,21,19,18,5 as outputs
//
void setup()
{
    unsigned char i;
    for(i=0; i < 8; i++)
    {
        pinMode(LEDs[i], OUTPUT);
    }
}

//
// Turn the LEDs ON/OFF to simulate the
// Knight Rider car
//
void loop()
{
    for(int k = 0; k < 8; k++)
    {
      digitalWrite(LEDs[k], HIGH);
      delay(del);
      digitalWrite(LEDs[k], LOW);
    }

    for(int k = 6; k > 0; k--)
    {
      digitalWrite(LEDs[k], HIGH);
      delay(del);
      digitalWrite(LEDs[k], LOW);
    }
}
```

connected to are configured as output ports. Inside the main program two **for** loops are established. Inside the first **for** loop the LEDs from MSB to LSB are turned on for 100 ms. Inside the second loop the LEDs from LSB to MSB are turned On for 100 ms. Thus, the net effect is that the LEDs 'chase' each other in both directions.

The program is also available as a free download [1]. As an assignment to test your skills: the delay time between each output is set to 100 ms. Try modifying this time and see the effects on the display.

200234-01

# LW/MW AM Signal Generator with ATtiny13




By **Burkhard Kainka** (Germany)

Anyone who designs, builds, or repairs radio receivers can make good use of a small AM signal generator. With a suitable signal waveform, you don't even need to be able to adjust the frequency. This works if the generator operates at an adequately low frequency and generates enough harmonics. The ATtiny13 AM signal generator described here produces short pulses at a rate of 70 kHz. This results in strong harmonics covering the entire long-wave (LW) and medium-wave (MW) radio spectrum.

The pulse train is also briefly interrupted at regular intervals to produce (pseudo) amplitude modulation (AM) at a frequency of about 750 Hz. A radio can therefore receive an AM test signal at 70 kHz, 140 kHz, 210 kHz, etc. and demodulate it to produce an audible tone.

**Listing 1: LW/MW AM sig gen program.**

```
'ATtiny13 AM Generator
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4
Config Portb = Output
Dim N As Byte
Do
  For N = 1 To 50          'generate 70 kHz
    Portb = 255
    Portb = 0
  Next N
  For N = 1 To 50          'pseudo AM 750 Hz
    nop
    nop
 Next N
Loop
End
```

A wire loop with a diameter of about 4 inches (10 cm) makes a suitable antenna. It generates an AC magnetic field that couples directly to the ferrite rod of the receiver.

The signal generator can also be used for comparative sensitivity measurements by testing how far away from the generator the signal can still be heard. A good receiver should receive a clear signal at distances up to 3 feet (1.5 m) away from the transmitter.

The ATtiny13 source code is shown in **Listing 1**.

200197-01

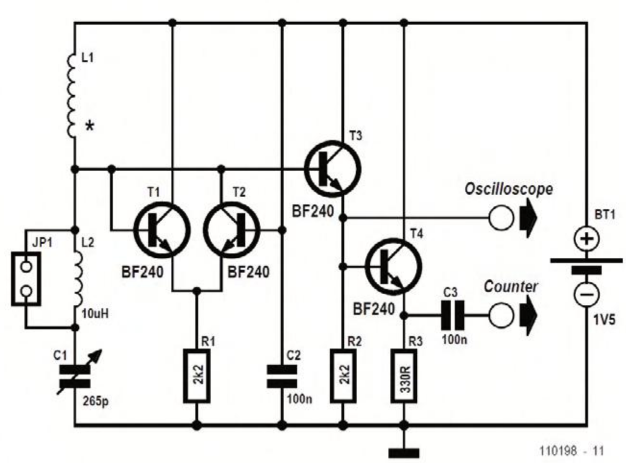
# Absolute Minimalist **Dip Meter**

By **Burkhard Kainka** (Germany)


In days gone by a radio amateur always had a (grid) dip meter close to hand in the 'shack.' Now that people can afford oscilloscopes, the poor old dip meter has lost its importance and is frequently no longer to be seen except perhaps in *Retronics*. Actually, this is a shame because many seemingly complex tasks are easy to carry out with a dip meter. Anyone who's interested (perhaps the second time around) can easily build one rapidly with this very simple but adequate circuit. The interesting question is what do you actually need from a dip meter?

- A visual display of the dip? Nope, the 'scope can handle that task.
- A large frequency scale? Not necessary, as you can connect a frequency meter for this.
- An array of coils? We don't need these because we can use a jumper to change range (no coils to lose any more!).

The sensor coil L1 has ten turns and is wound using an AA-size battery as a former. This coil will allow us to over the range from 6 MHz to 30 MHz. With jumper JP1 open an additional fixed inductance of 10 µH comes into circuit. The frequency measurement range is then from roughly 2.5 MHz to 10 MHz. The switch may be replaced by a jumper. To take measurements you hold a resonant circuit close to the sensor coil. Tune the rotary capacitor C1 slowly to and fro in order to find the resonant frequency, at which the oscillator amplitude decreases somewhat. The frequency can then be read directly off the oscilloscope.




To obtain a very accurate measurement you can additionally connect your frequency meter (counter) to the second output.

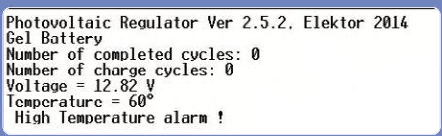
200199-01

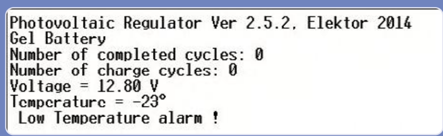
# Photovoltaic Regulator Update

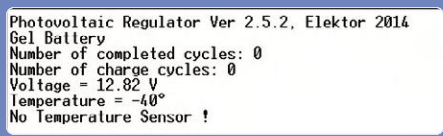
## New function for measuring battery temperature

By **Pascal Rondane** (France)

The modification to the photovoltaic regulator described in this article is intended to manage the use of the battery at the limits of the allowable range defined by its manufacturer, and even beyond this.




In edition 3/2016 (May & June) we presented a photovoltaic power regulation module. Originally intended for powering a small weather station, it can be used in many situations where independence is of paramount importance, and with it, optimum management of the energy captured by a solar panel and stored in a battery. That first article [1] included provision for a function to monitor battery temperature, but it was not yet implemented…

Here it is at last, successfully completed and duly tested by the author. Do note that this new version only serves any purpose if you want to use this improved function; it is not strictly speaking an actual correction to the previous version.

### Hardware

Two options are possible:

1. In the event of the battery temperature alarm thresholds being exceeded, the microcontroller shuts down the electronics. The regulator then has to be reset *manually*. To achieve this, zener D1 on the existing circuit is removed **(Figure 1)**, which also prevents the module from being able to re-arm itself fully automatically when the battery goes back above 12 V following a deep discharge.

2. In the event of the battery temperature being exceeded, the MICROCONTROLLER lights status LED D2 and automatically disconnects the user load and the solar panel. Diode D1 remains in circuit, the regulator is not disconnected and does not lose any of its functions described in the first article.

So in terms of the hardware, there's nothing much to change. We'll come back to this in a moment. Let's first take a look at the new software.

### Software

The addition of the battery temperature measurement function has given rise to a new version of the software `(Photovoltaic regulator-Ver 2.5.2)` [1] – and let me point out right away that this one can only be modified and compiled using the full version of Bascom.

In order to measure the temperature, I am making use of a table describing the transfer function between ambient temperature

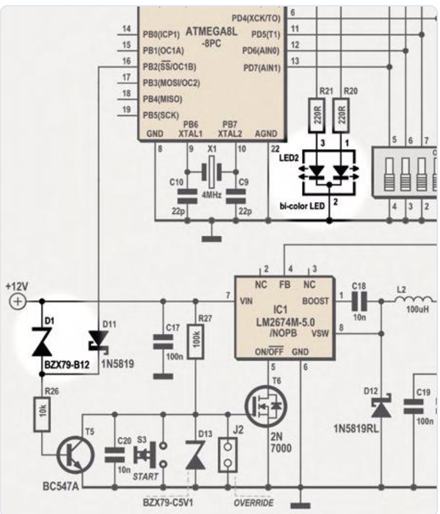



Figure 1. Using the new version of the battery temperature monitoring software, two operating modes are possible; one of them does not use D1.

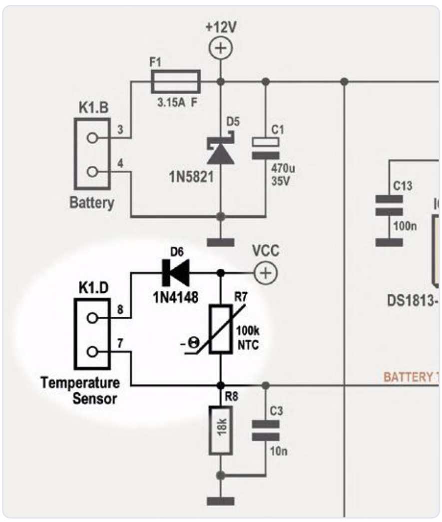



Figure 2. If the thermistor is connected to terminal block K1.D, D6 must be replaced by wire jumper.

and the one provided by the NTC thermistor manufacturer, which allows data acquisition in steps of 5°C [9°F]. I then perform a linearization between these two measuring points in order to obtain a precision of 1°C [1.8°F]. The program contains comments and the documented calculation formulas make it easy to understand the operation.
For readers interested in such questions, I'll just point out there is another method for linearization using a parallel resistance with the same value as the thermistor [2].
The temperature measurement routine is called regularly from the program primary loop; the temperature and system status can be recovered via the serial port (see **screen grabs**).

The temperature alarm thresholds are set by three easily-configured variables in the `Temperature alarm thresholds setting` section of the program. This considers the thermistor to be missing if the "measured temperature" is below −40°C [−40°F], which is pretty rare in our latitudes!

The new **table** shows how LED D2 reports the system status. The details given in the first article are still valid, they are just repeated here and completed for the new version of the software.

## Modifications

### CONVERTER MODULE SUPPLY

For the microcontroller to shut down the electronics in the event of excessive battery temperature, remove D1. Restarting following shut-down is then manual and is achieved using push-button S1.

### TEMPERATURE DETECTOR (NTC thermistor)

There are two ways to connect the thermistor (**Figure 2**): either directly to the PCB (R7), requiring no modification, or to terminal block K1.D, in which case diode D6 (1N4148) must be replaced by a wire link or a 0 Ω resistor. This is necessary, as otherwise the diode causes a voltage drop that would distort the temperature measurement.

If you want to locate the thermistor remotely, outside the case, the cable length (a twisted pair) must not exceed 40 cm. The thermistor can be insulated using heatshrink and then stuck to the battery with a spot of glue.
I used a thermistor from Vishay: NTCLE100E3104JB0 (*Farnell*: 1187033) to which I have matched the software. If you want to use a different type, all you have to do is change the values in the program's `Temperaturetable`.

For the sake of completeness, here by way of an example are the characteristics of a Yuasa battery as given by the manufacturer.
At start-up, before the indications in the table appear, there are two brief red-orange-green-off sequences followed by 2 seconds of green. This LED goes out when the battery level is too low (10.8 V); the 5 V supply is then shut down and the microcontroller stops operating.

140039 | 200261-01

| LED2 | | | | |
|---|---|---|---|---|
| **Color** | **Flashing (mark/space ratio)** | **Solar panel** | **User load (S2=off)** | **Battery** |
| orange | slow (33%) | not connected | connected | charged |
| green | slow (2%) | connected | connected | charging |
| orange | fast (10%) | connected | connected | low |
| red | slow (2%) | connected | not connected | discharged |
| off | – | connected | not connected | too low |
| | **State** | | **Message** | |
| orange steady | thermistor missing | | No Temperature Sensor! | |
| red steady | temperature lower than set-point | | Low Temperature alarm! | |
| red steady | temperature higher than set-point | | High Temperature alarm! | |

At power-on a short sequence of red-yellow-green-off-red-yellow-green-off is shown, followed by 2 seconds of green. Next the status appears according to the above table. When the battery level drops to a dangerously low level (10.8 V), the 5-V supply is switched off and the microcontroller halts execution.

*Table 1. The indications for the 2-color LED D2.*

| Characteristics | | |
|---|---|---|
| Nominal voltage | 12 | V |
| Nominal capacity, 20 hours to 1.75vpc (30°C) | 7 | Ah |
| Nominal capacity, 10 hours to 1.75vpc (30°C) | 6.4 | Ah |
| **Dimensions** | | |
| Length | 151 (±1) | mm |
| Width | 65 (±1) | mm |
| Total height | 97.5 (±2) | mm |
| Weight | 2.2 | kg |
| **Connection** | | |
| Faston connector | 4.7 | mm |
| **Temperature range** | | |
| Storage | –20 tot +60 | °C |
| Charging | –15 tot +50 | |
| Discharging | –20 tot +60 | |

*Table 2. Characteristics of the NP7-12 lead-acid battery from Yuasa.*

## WEB LINKS

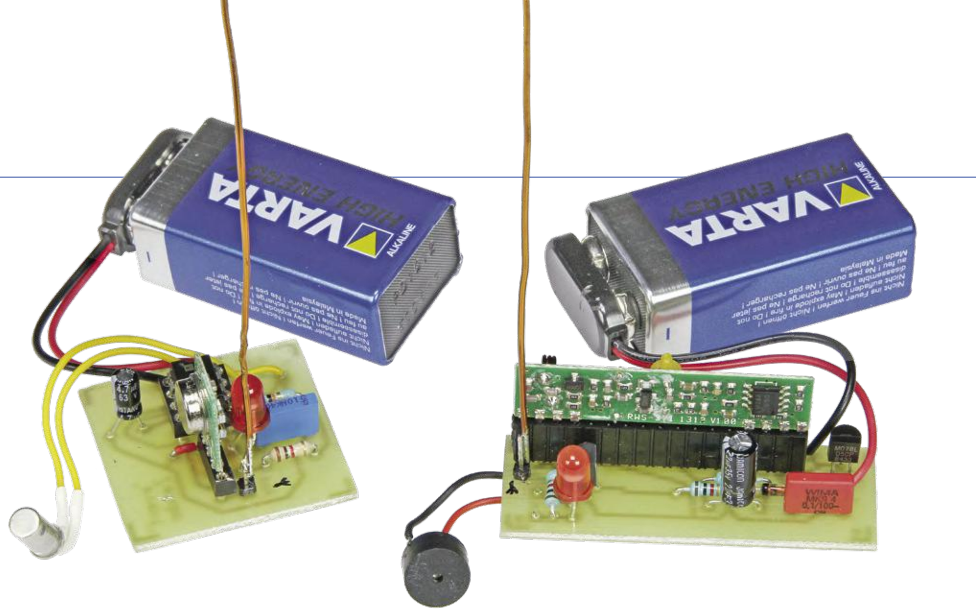

[1] **50-W Solar Cell Voltage Regulator, Elektor Magazine 3/2016 (May & June)**: www.elektormagazine.com/080305

[2] **Source software**: www.elektormagazine.com/200261-01

[3] http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/electro/thermist.html

# Wireless **Signalling**

## Transmitting switching signals over LPR




By **Wouter Eisema** (Netherlands)

Not much is required to detect the state of a switch or contact from a distance if you use cheap, ready-made modules that operate in the LPR/ISM bands like 433 and 868 MHz. Only a few components need to be added

This project consists of a small transmitter and receiver which use a cheap, ready-made transmitter and receiver module for the 433-MHz ‘ISM’ band where type approved low-power radios (LPR) may be used license-free. Adaptation to 868 MHz or other ISM bands where LPR is allowed should be easy by selecting a suitable radio module. Both circuits have been designed to signal the state of a switching contact—open or closed—wirelessly across a considerable distance (depending on circumstances up to 300 feet). Using this you can keep an eye on the position of a switch, microswitch or a relay contact which is located somewhere else. The circuit was originally designed by the author for use in combination with the Seismic Detector, also described in this edition but you can use it just as well to check the position of a garage door.

To keep everything as simple as possible and using as few components as possible, the choice was made for a design in which the on/off-state of the switch to be monitored is transmitted by switching the LPR carrier on and off. There is no further modulation or coding of the signal, so that the receiver will also react to other LPR transmitters in the vicinity. But it is likely that there are not all that many other hobbyists in your neighborhood and this way it remains a simple, cheap solution for the purpose of indication in applications that are not too demanding.

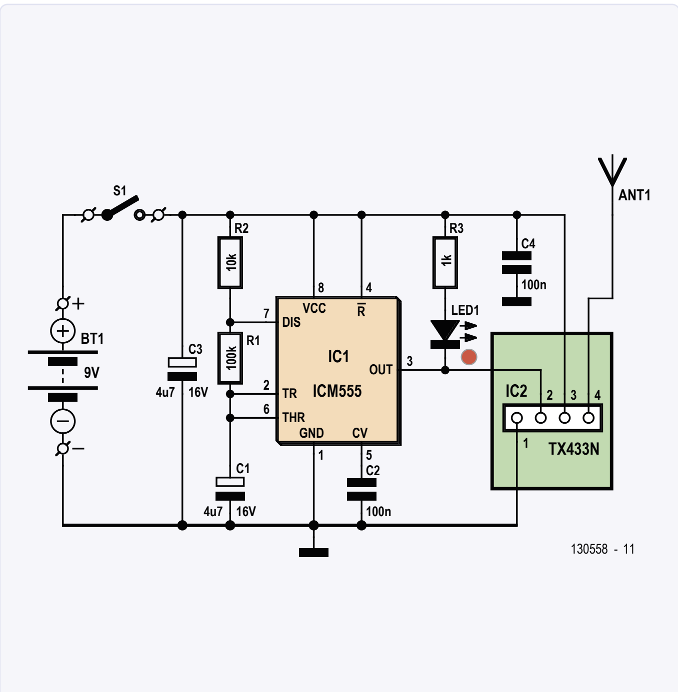



*Figure 1. The transmitter circuit consists of a 555 configured as an oscillator and a ready-made transmitter module.*

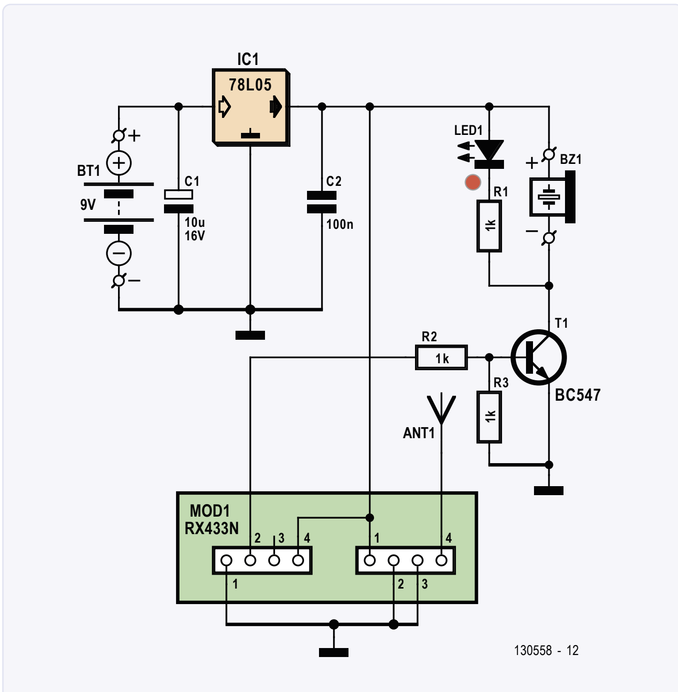



*Figure 2. The design of the receiver circuit is possibly even simpler. The digital output from the receiver module switches an LED and a buzzer via a transistor.*

## COMPONENT LIST

### Transmitter

**Resistors**
R1 = 100kΩ
R2 = 10kΩ
R3 = 1kΩ

**Capacitors**
C1,C3 = 4.7µF 16V radial
C2,C4 = 100nF

**Semiconductors**
LED1 = LED, low-current, red, 3mm
IC1 = ICM555

**Miscellaneous**
IC2 = TX433N transmitter, ISM LPR (Velleman)
4-way socket for transmitter module
17 cm (6.7 in.) wire for antenna

### Receiver

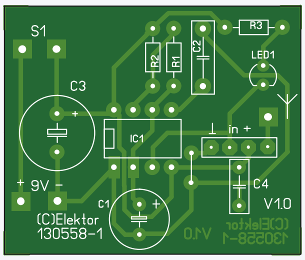
**Resistors**
R1,R2,R3 = 1kΩ

**Capacitors**
C1 = 10µF 16V, radial
C2 = 100nF

**Semiconductors**
LED1 = LED, low current, red, 3mm
T1 = BC547
IC1 = 78L05

**Miscellaneous**
Mod1 = RX433N receiver, ISM LPR (Velleman)
2x 4-way socket for receiver module
17 cm (6.7 in.) wire for antenna
Bz1 = active piezo buzzer, 5V

*Figure 3. The circuit board for the transmitter.*



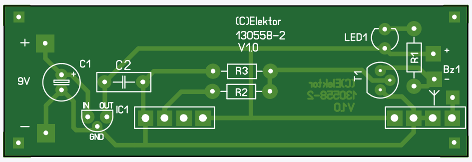




*Figure 4. The receiver circuit board also contains few components, but has an elongated shape because of the dimensions of the receiver module.*

For the 433-MHz modules, the author used type numbers TX433N (transmitter) and RX433N (receiver) from Velleman, but in principle any pair of LPR transmitter/receiver module that operates in an ISM band can be used. The pinout may be different, in which case you will have to adapt the circuit boards. But considering the small number of components, the transmitter and receiver can also easily be built on small pieces of prototyping board.

### Schematics

The transmitter in **Figure 1** consists of not much more than a 555, which is configured as an oscillator (IC1) and a transmitter module (IC2). S1 is the switch or contact you wish to monitor. When this switch is closed, the entire circuit is powered and the 555 begins to oscillate at a frequency of about 1 Hz (duty-cycle 50%). The output of the 555 is connected to the Data input of the transmitter module, so that it will be transmitting the carrier in this 1-Hz rhythm. LED1 flashes when the transmitter is activated. When S1 is opened, the supply voltage is removed from the circuit and will therefore not consume any power. The 9-V battery used for the power supply will therefore last a very long time (provided the signal is not transmitted too frequently or for very long periods of time).

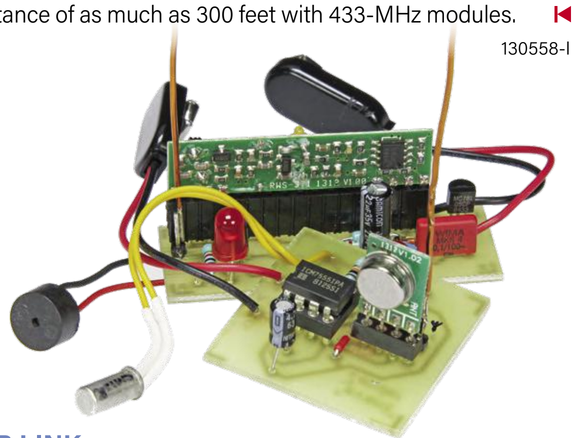
The design of the receiver circuit in **Figure 2** is just as simple as the transmitter, comprising mainly of the receiver module with a switching transistor and a voltage regulator. IC1 provides a stable 5-V power supply voltage for the receiver module and the remainder of the 'circuit.' The digital output of the module (pin 2) is connected to the base of T1, which is tasked with the responsibility of driving the self-oscillating piezo buzzer and the LED. When the receiver module detects the carrier wave, T1 will be switched on and the LED will flash and the buzzer will sound at the 1-Hz rhythm.
Since the receiver has to be active all the time it is recommended to power this from a 9-V wall adapter instead of a 9-V battery.

### Construction and use

Two little circuit boards (**Figures 3** and **4**) have been designed for the transmitter and receiver, but as already mentioned, the number of components is really small and you can therefore also easily build the circuit using small pieces of prototyping board. As usual, the circuit board layouts are available as a free download [1].
The assembly of both circuit boards should be without any problems. The transmitter and receiver modules are plugged into the circuit board via a single and a dual 4-way socket respectively. For the antenna, a piece of stiff copper wire with a length of 17 cm (6.7 inches) can be soldered into each circuit board. On the transmitter side the connections for S1 on the circuit board are connected to the switch or contact to be monitored. For example, this can be the relay output from the earthquake detector also described in this issue. When there is an earthquake the relay will energize and this activates the transmitter. On the receiver side the LED will light up and you will hear the intermittent noise from the buzzer. After about 20 seconds (or a manual reset) the relay will switch off automatically and the transmitter stops.

There may be a need for some experimentation and trial and error with the transmitter and receiver in order to obtain a satisfactory reception in certain situations. Reinforced concrete in the floor and ceiling can limit the range of the transmitter considerably. In the open air you can cover a distance of as much as 300 feet with 433-MHz modules.

130558-I



### WEB LINK

[1] www.elektor-magazine.com/130558

# Ultrasonic Reverse Parking Aid
## with Arduino Uno


*Figure 1: KY-050 ultrasonic module.*

By **Dogan Ibrahim** (United Kingdom)

In this project, the ultrasonic sensor module type KY-050 is used together with an Arduino and a type KY-012 active buzzer to help while reverse parking our vehicle. As the distance to the objects gets smaller, the buzzer sounds faster to warn the driver that the objects at the rear of the vehicle are nearer. The aim of this little project is to show how the ultrasonic sensor module can be used in a project to measure distance.

The KY-050 is a 4-pin module shown in **Figure 1.** This module uses the HC-SR04 type ultrasonic transmitter/receiver hardware, and it has the following features:

- Operating voltage: 5 V
- Operating current: 2 mA
- Detection distance: 2–450 cm
- Input trigger signal: 10 µs TTL
- Sensor angle: <15 degrees

The KY-050 has the following pin names and descriptions: **Vcc**: Power input; **Trig**: Trigger input; **Echo**: Echo output; **Gnd:** Power ground. The sequential operation of the KY-050 ultrasonic sensor module is as follows (refer to **Figure 2**):

- a 10-µs trigger pulse is sent to the module;
- the module then sends eight 40-kHz square wave signals to the target and sets the Echo pin High;
- the program starts a timer;
- the signal hits the target and echoes back to the module;
- when the signal is returned to the module the echo pin goes Low;
- the timer is stopped;
- the duration of the echo signal is calculated and the result s proportional to the distance to the target.

The distance to the object is calculated as follows:

*Distance to object* (in m) =
(*duration of echo time* in seconds × *speed of sound*) / 2

The speed of sound is approximately 340 m/s, or 0.034 cm/µs, therefore,

*Distance to object* (in cm) = (*duration of echo time* in µs) × 0.034 / 2

or,

*Distance to object* (in cm) = (*duration of echo time* in µs) × 0.017

For example, if the duration of the echo signal is 294 microseconds then the distance to the object is calculated as follows:

*Distance to object* (in cm) = 294 × 0.017 = 5 cm

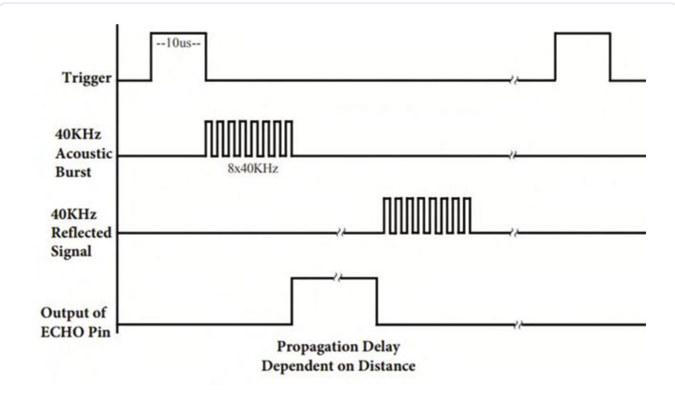



*Figure 2: Operation of the ultrasonic sensor module.*

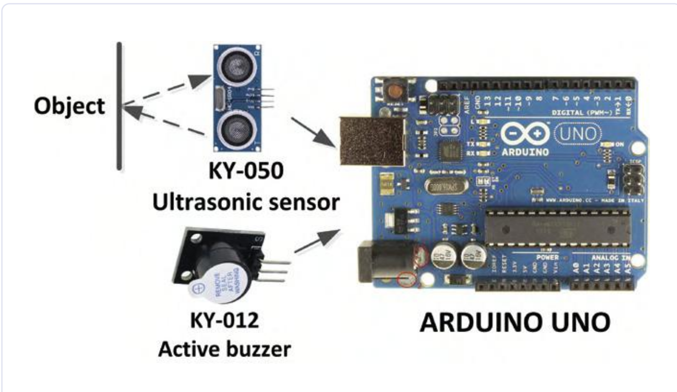



*Figure 3: Block diagram of the project.*

**Listing 1: 'ultrasonic' program for Arduino**

```
/****************************************************
 *   ULTRASONIC REVERSE PARKING WITH BUZZER
 *   ======================================
 ***************************************************/
int trig = 2;                       // trig pin
int echo = 3;                       // echo pin
int buzzer = 4;                     // buzzer
int dely;

long tim;
float distance;

void setup ()
{
  Serial.begin(9600);
  pinMode(trig, OUTPUT);         // trig is output
  pinMode(echo, INPUT);          // echo is input
  pinMode(buzzer, OUTPUT);       // buzzer is output
  digitalWrite(buzzer, LOW);     // buzzer OFF
}

void loop ()
{
  digitalWrite(trig, LOW);       // clear trig
  delayMicroseconds(5);          // 5us delay

  digitalWrite(trig, HIGH);   // set trigger HIGH
                              // for 10us
  delayMicroseconds(10);      // 10us delay
  digitalWrite(trig, LOW);    // remove trigger
  tim = pulseIn(echo, HIGH);   // read the echo
  distance = tim * 0.034 / 2; // calculate distance
  Serial.println(distance);    // display distance

  //
  // Set the delay depending on the distance
  // to the object
  //
  if(distance > 100)
    dely = 0;
  else if(distance > 70 && distance < 90)
    dely = 600;
  else if(distance > 50 && distance < 70)
    dely = 400;
  else if(distance > 30 && distance < 50)
    dely = 300;
  else if(distance > 10 && distance < 30)
    dely = 200;
  else if(distance < 10)
    dely = 10;

  if(distance < 100)             // if less than 100cm
  {
    digitalWrite(buzzer, HIGH);  // buzzer ON
    delay(dely);                 // delay
    digitalWrite(buzzer, LOW);   // buzzer OFF
    delay(dely);                 // delay
  }
}
```

**Figure 3** shows the block diagram of the project (in 'embedded platform' style) and **Figure 4**, the connection diagram. The following pins are connected between KY-050, KY-012 and Arduino Uno:

| Arduino Uno Pin | KY-050 pin | KY-012 pin |
|---|---|---|
| 2 | trig | |
| 3 | echo | |
| 4 | | S |
| GND | GND | GND |
| +5V | Vcc | |

The KY-012 buzzer module is directly connected to an Arduino Uno port pin 4. It is recommended to connect a 100-Ω series resistor with this module in order to limit the port output current, especially if other devices also draw current from the port pins.

Referring to **Listing 1**, at the beginning, trig, echo, and buzzer are assigned port numbers 2, 3, and 4. Inside the setup routine buzzer is configured as an output and it is turned OFF. Also, trig and echo are configured as output and input respectively. Inside the main program loop, a trigger pulse is sent and statement `pulseIN` is used to read the echo signal. The distance to the object in front of the ultrasonic sensor is then calculated in centimetres and is stored in a variable called *distance*. Variable `dely` (*sic*) is set to different values depending on the calculated distance. `dely` gets smaller as the sensor gets nearer the object. Finally, the buzzer is activated with the duration set to variable `dely`. The net result is that the rate of the sound is increased as the sensor gets nearer the object. Notice that the distance is displayed on the serial monitor. And finally, **Figure 5** shows the circuit built on a breadboard. Not bad for a Sunday afternoon!

200211-01

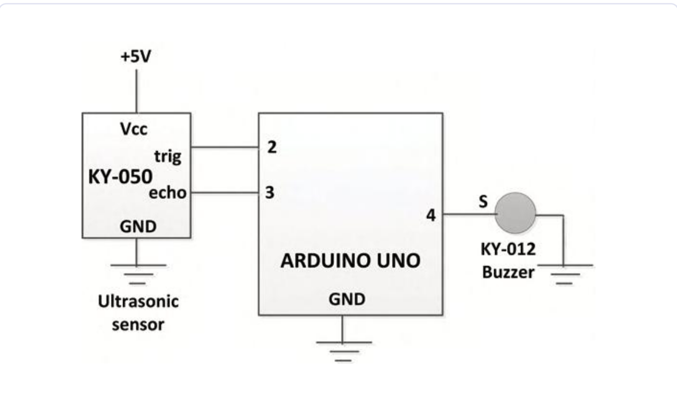



*Figure 4: Circuit diagram of the project.*

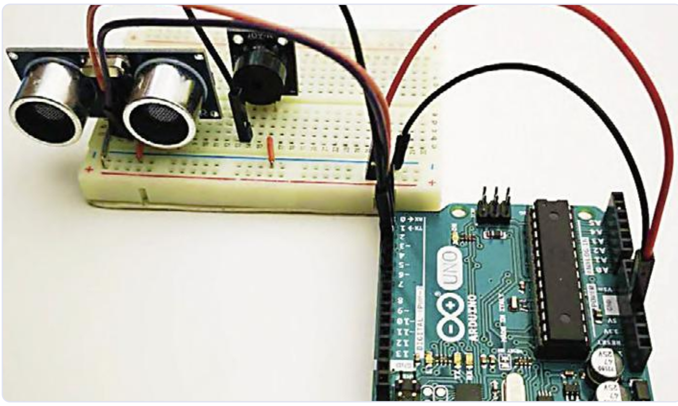


*Figure 5: Circuit built on a breadboard.*

# Distortion Pedal
## with Op Amp and Space Charge Tubes


By **Richard Honeycutt** (USA)

This project is a 'hybrid,' combining integrated-circuit and tube technology in one circuit. It was developed by the author as a case study that's elaborately described in his book *The State of Hollow State Audio — in the Second Decade of the 21st Century*, published by Elektor. The 12EL6 tubes used here are low-voltage "space charge' types discussed at length in the book.

The IC seen in the circuit diagram in **Figure 1** is a type OP27 op amp. The type LTC6090 IC originally used in the design proved hard to get in experimenter quantities. Since the input resistance of the op-amp stage must be about 1 MΩ, an op amp is needed that has very low bias current, offset current, and offset voltage. Also, the op amp must have very low noise. The OP27 meets these requirements and is readily available in small quantities. The LTC6090 boasted rail-to-rail input and output voltage ranges, whereas for a 2 kΩ or higher load the OP27 has a typical span of 1.5 V (maximum 3.5 V) between maximum output voltage and the supply rail. This requires a change in our thinking about how to use the op amp to set the control-grid bias of V1. Other low-bias-current, rail-to-rail op amps are manufactured, but I did not find any that are easily available in small quantities. Thus the OP27 was selected and using it we need a quiescent DC output voltage at least 3.5 V more positive than the negative supply (which is zero volts dc in this single-ended-supply configuration). We also need to allow about a volt for the output voltage swing to avoid clipping in the op amp stage. Then we need for the V1 control grid bias to be adjustable from about zero to -0.2 V. Of course, with a single supply, we cannot produce a negative voltage, so we'll have to make the cathode somewhat positive. In the original circuit, we used two diodes in series between cathode and ground to place the cathode about 1.2 V above ground. Thus a 1-V control-grid voltage gives us an equivalent bias of –0.2 V. But we can't drive the output of the OP27 quiescently that low. Since the forward voltage drop of a semiconductor diode is not well-defined (0.6 V is the approximate value), I found experimentally that by using only one diode between cathode and ground of V1, and making the adjustment range of the non-inverting input to the op amp span from 0.387 V to 5.6 V, I could obtain the desired range of control-grid bias. Knowing from the earlier work that the input resistance of the cathode follower would load the V1 stage, I examined the output waveform from V1. I found that it required a larger signal than a typical humbucking pickup would produce in order to create the distortion signature I wanted, so I increased the op-amp gain by changing feedback resistor R4 to 10 MΩ. Then I verified that the op amp could not be driven into clipping until the triode was already in hard clipping, so that our circuit would provide a 'tube 'sound.

In the resulting op amp circuit, R1, R2, R3, and D4 give the required adjustment range, while D5 acts as a 3.1-V level shifter to bring the DC level down to the necessary range for proper grid bias. I used a

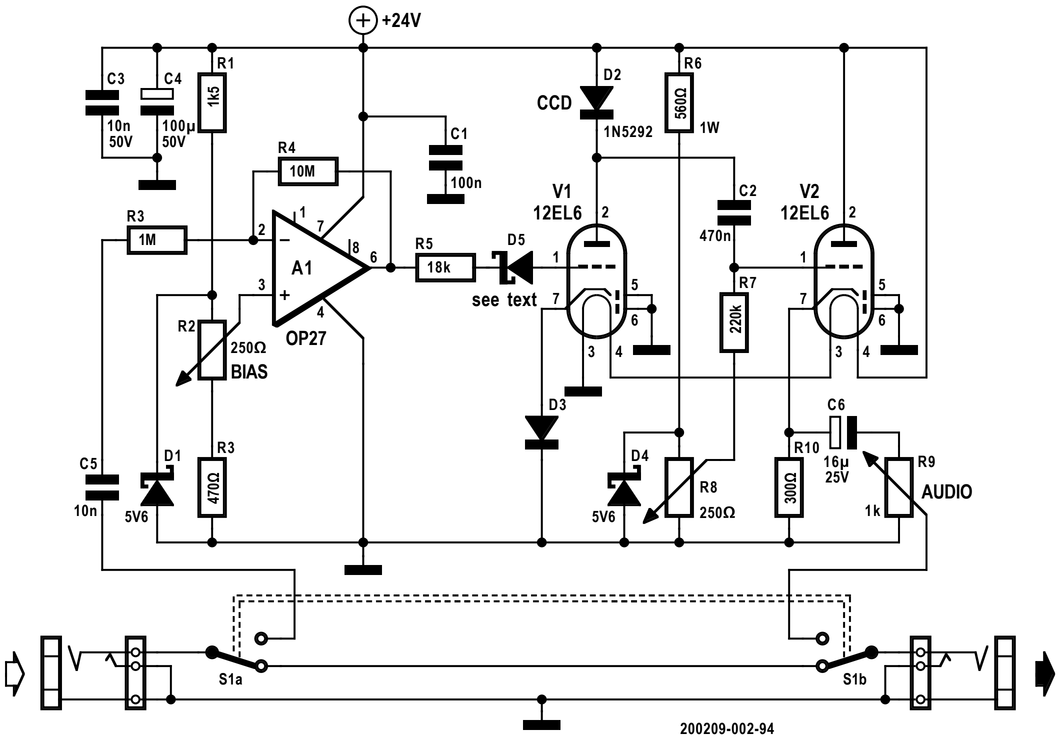



*Figure 1: Final circuit for the hybrid IC/tube distortion pedal.*

Zener diode, but in order to get the proper level shift at the low current provided by the triode, I had to use one specified at 4.5 V in order to get a 3.1-V level shift. I could have used a potentiometer to make this adjustment, but that would have affected the AC level as well, which I didn't want.

Using a type 1N5292 CCD (constant-current diode; 0.62 mA) in position D2 and a 24-V B+, we have the load line shown in **Figure 2**. (Note that I have added an approximate plate characteristic curve for a control-grid bias of -0.1 V.) You can see from the figure that the 12EL6 triode will be essentially cut off at a control-grid voltage of -0.2 V. The graph indicates cutoff near a control-grid voltage of -0.17 V, but if we actually plotted the triode's performance in this range, there would be some curvature in the load line as the CCD became unable to deliver the full 0.62 mA at increasingly negative control-grid bias values. With this arrangement, the graph tells us that changing the signal voltage from -0.17 V to 0 V gives us a plate voltage change of 11.5 V — a voltage gain of 68. (Really, the math tells us that the gain equals the µ of 55; the discrepancy stems from the graph's inaccuracy near cutoff, and our inability to precisely read the graph.)

Back to the circuit, the footswitch when pressed toggles between distortion and pass-through. At the output, potentiometer R9 sets the level of the distortion output, so it can be matched as desired with the "clean" or straight-through output.

Since a 24-VDC power supply is used, the filaments of V1 and V2 are connected in series. To avoid buzz and other unwanted effects on the signal fed to the guitar amplifier, a clean, regulated, and properly decoupled +24 V supply voltage is a must. A type LM7824 three-terminal regulator will do the job.

200209-01

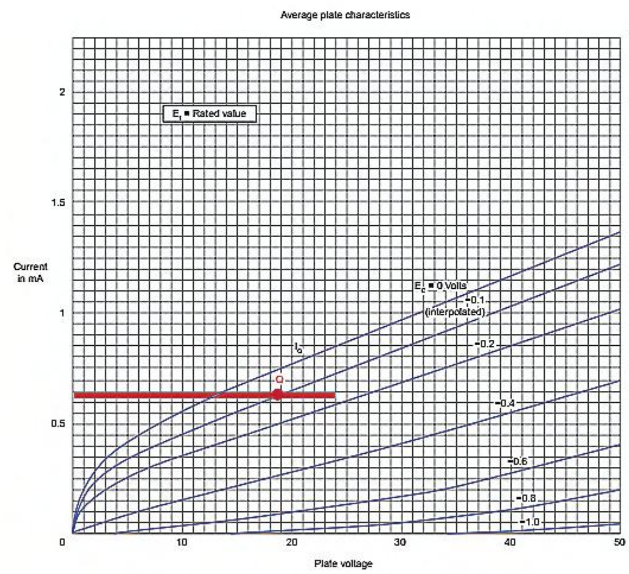



*Figure 2: Load line for a 12EL6 tube using a 24 V B+, with a 0.62-mA constant-current diode (CCD) instead of a plate resistor.*

# Electronics Workspace Essentials

By Elektor Team


Whether you are a pro engineer or maker, you must have an adequately equipped electronics workspace for designing and testing your projects. In this article, we offer simple advice and recommendations on affordable tools, test gear, lighting, energy sources, prototyping, and workbench organization.

Whether you are a pro engineer or maker, you must have an adequately equipped electronics workspace for designing and testing your projects. Understandably, you will have to work within the constraints of your budget while still creating a space that is desirable and practical. If you're looking for tips, tricks and suggestions, we're here to give you a basic summary of where to start. We will give you simple advice and recommendations on affordable tools, test gear, lighting, energy sources, prototyping, and workbench organization.

*Figure 1: Toolcraft helping hand LED magnifier.*

## Electronics workspace furniture & lighting

Firstly, a quality electronics workspace requires a sturdy, flat surface which is preferably white. This will enable better lighting through increased reflection and will also assist you in quickly locating and recognizing parts and tools.

Nobody wants an achy back. Make sure you get yourself a high-quality, height adjustable swivel chair to make your long soldering and design sessions all the more comfortable.

Furthermore, make sure you have plenty of light. A good LED lamp should be positioned in an upper-left side position on your workbench for those right-handed and visa-versa for those who are left-handed. This way, a shadow will not be cast from your working hand over the work.

You'll likely want to get up close and personal with your project and, in any case, components are getting smaller and smaller. A solution like the Toolcraft Helping Hand LED Magnifier (**Figure 1**) will help you by combining excellent LED lighting with magnification to provide an excellent third/helping hand.

## Electronics workspace organisation

The organisation of your electronics workspace should be both personal and organic. That said, having your equipment readily available at arm's length can make a task less stressful, quicker, more meaningful and will generally give you a more satisfying time. The bane of any electronics enthusiast is tangled wires. This problem can be solved by constructing a simple wire spool.

Ridiculously easy to build, these can consist simply of an old piece of scrap dowel secured between two pieces of square shaped wood.
Here are some more simple tips to improve your efficiency and arrangement:

- Use labelled jam jars for spare parts, fasteners, screws etc.
- Use colour coding for different components.
- Consider purchasing pre-assembled toolkits [1]. iFixit (**Figure 2**) specialize in high-quality, easy access kits in handy pouches to provide all the tools you need (and none that you don't!)

## Test and measurement equipment

When it comes to testing equipment, there is no substitute for quality. Quality in electronics testing is synonymous with accuracy. Here are some tips and tricks to get you started:

- **Multimeter**: A good multimeter is the staple of any workbench. You'll need this to measure continuity, resistance, capacitance, and current. Furthermore, it should be able to measure in the micro-amps range. To be confident your multimeter is within specification, it's worthwhile sticking to named brands including PeakTech, Siglent, Fluke, Voltcraft etc. Elektor sells a diverse range of high-quality multimeters [2].
- **Power supply**: You'll need at least one power supply, preferably one with a variable output. You can never have enough power, so you might want three or four of these.
- **Signal generator**: You'll likely need a way of generating a signal. Therefore a signal generator capable of generating basic signals including triangular, sine, and square is a must.
- **Oscilloscope**: Another necessity for your workbench is an oscilloscope, which is essentially a voltmeter that provides extra visual information. Standalone units can be bulky and cumbersome, so, therefore, consider a modern USB alternative like the SmartScope [3] shown in **Figure 3**.

## Tools of the trade

"A workman is only as good as his tools." That old adage is never truer here. Quality tools produce quality work. Want to know which tools are crucial for an effective workspace? Here are some recommendations:

- **Soldering equipment**: The most fundamental tool to the electronics enthusiast's workbench is surely the soldering station. Whether you're performing repairs or constructing circuits, a quality piece of equipment is vital for the optimum outcome. There's no need to spend hundreds here, just make sure you get a decent branded variable temperature device with a footprint not too large for your bench. Whilst not inexpensive, the Weller WT 1014 [4] offers an excellent all-round quality solution. Also, don't forget to purchase a good solder sucker and solder proof mat. It will save your bench and your components in the long run!
- **Fan**: Instead of spending hundreds on an extractor, a simple desktop fan will help get rid of those nasty solder fumes. It


*Figure 2: iFixit toolKit.*


*Figure 3: SmartScope USB oscilloscope.*

will also help keep you cool during the hot summer months.

- **Side cutters and pliers**: A good quality pair of precision, flush-cutting side cutters are a necessity, as are snipe-nosed and flat-nosed pliers. The Lindstrom brand provides an excellent range of these tools.
- **Glue**: The hot glue from a glue gun has a wide variety of uses: among the most important are providing mechanical support to electronic components and wires, electrical insulation, and helping in general electronics assembly
- **Wire strippers**: When your side cutters aren't up to the job, wire strippers will always be a useful addition to your tool collection.
- **Digital calipers**: Digital calipers are a requisite on any home workbench and have largely superseded traditional rulers for measuring. That said, you will likely still need a basic metal ruler.
- **Knife**: A good knife or scalpel is essential for re-working and etching tracks into your PCBs.
- **Screwdriver set**: You'll likely need tools for taking things apart and putting them back together. Consider screwdriver sets containing precision, slotted, Phillips, Torx types and Allen keys. Having these items will make your tasks and operations flow as smoothly and stress-free as possible.

Stock up on basic components. If you are on a tight budget and can't stock your workspace in a single shopping spree, make a list of your most frequently used components and start with those.

## Key components and supplies

We recommend the following key components and supplies.

- **Solderless breadboards**: The most established technique for prototyping is by using solderless breadboards. These will enable you to easily design, prototype and test your circuits.
- **Stripboard**: A different but popular method for prototyping your circuits is by using stripboard a.k.a. perfboard, a popular variation of which is Veroboard.
- **Common components**: It's always a good idea to have an assortment of resistors, capacitors, inductors, diodes, transistors, and LEDs on hand.
- **Solder**: Try to avoid anything too thick. A good thin (<0.5 mm) 60/40 tin/lead solder should cater for most of your soldering requirements
- **Freeze spray**: Imperative for finding dry joints in your circuits. Also particularly useful in helping to loosen stiff fasteners and screws.
- **Compressed non-ionized air**: Great for cleaning, this will prevent the electro-static build-up associated with other methods of cleaning.
- **Cleaning solvents**: You will find isopropyl and methylated spirits handy for a variety of purposes.
- **Glue**: Extra glue for your glue gun, epoxy resin (Araldite, JB Weld) and Loctite to help keep your screws and fasteners in place.
- **Leads**: Alligator clip type and banana plug type are particularly useful.
- **Safety equipment**: For the more safety conscious among us, safety specs and a first aid kit are a requisite.

## Design → Build → Sell Electronics

It shouldn't cost an arm and a leg to add an electronics workspace to your home, apartment, or office. However, quality is key if you wish to get the best results. Spending time researching and considering the suggestions put forward here will save you time and money in the long run as well as providing you with the workbench you crave. Hopefully, this article has provided you with some basic ideas of what you need to equip yourself with the fundamental resources to start out.

Here at Elektor, we are focused on helping engineers, makers, and students to design, build, and sell *electronics*. Sign-up for a free Elektor Labs account [5] so you can share your projects built in your newly overhauled workspaces and collaborate with designers from around the globe.

Also, show us your workspace! Visit the Workspace Submission page [6] to send us details and photos!

200231-01


## WEB LINKS

[1] Toolkits: www.elektor.com/tools/production/toolkits/

[2] Multimeters: www.elektor.com/tools/measurement/multimeters/

[3] SmartScope: www.elektor.com/smartscope-usb-oscilloscope

[4] Weller WT-1014: www.elektormagazine.com/news/review-weller-wt-1014-soldering-station

[5] Elektor Labs website: www.elektormagazine.com/labs

[6] 'My Workspace' Submission Form: www.elektormagazine.com/workspace-submission

# 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 **€50.00 each**, which should encourage all Elektor readers to participate.

### PARTICIPATE!

**Ultimately August 1, 2020**, 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 3/2020 (May & June) is: **638CB**.
The book vouchers have been awarded to: Ralf Kloos (Germany); Joseph Reding (France); Ola Sandin (Sweden); Isolde Tietz (Australia); Morris Beavers (USA).

**Congratulations everyone!**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | D | | A | | | 3 | 5 | C | E | | | 2 | | 8 | 1 |
| F | | | 5 | 4 | | C | | | A | | 2 | 6 | | | 3 |
| 4 | 1 | | | | | | | | | | | | | A | 0 |
| 3 | B | | 8 | F | 0 | | | | | 5 | 6 | 7 | | D | 4 |
| 0 | | | 1 | | | | 3 | E | | | | 8 | | | 2 |
| | 6 | | | | | | | | | | | | | 4 | |
| | 5 | 3 | 4 | | 6 | | 8 | B | | 2 | | 0 | 7 | 9 | |
| | | | 7 | 9 | | | A | 0 | | | 4 | B | | | |
| | | | | C | 1 | | E | D | | 8 | 3 | | | | |
| | 3 | | | D | 4 | | | | | E | 1 | | | 6 | |
| | | | | | | 6 | 0 | F | 4 | | | | | | |
| | E | | | 8 | | F | | | 6 | | C | | | 0 | |
| D | 2 | A | | 0 | | 7 | | | C | | 9 | | 8 | 3 | E |
| | | B | | | | | | | | | | | 2 | | |
| 9 | 8 | | E | | C | | | | | 0 | | A | | 5 | F |
| 5 | | F | 3 | | | | 6 | 4 | | | | 1 | 0 | | C |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | B | C | E | 6 | 5 | 1 | 4 | 0 | 7 | F | 2 | 3 | A | D | 8 |
| 4 | D | 8 | F | A | 7 | E | 0 | 9 | B | 1 | 3 | 5 | C | 6 | 2 |
| 0 | A | 3 | 5 | F | 2 | 9 | D | 8 | C | 4 | 6 | 1 | 7 | E | B |
| 1 | 7 | 2 | 6 | 3 | 8 | C | B | D | 5 | E | A | 9 | F | 4 | 0 |
| 2 | 3 | A | C | 1 | 9 | 8 | F | 7 | 6 | 5 | 0 | 4 | D | B | E |
| B | 4 | 5 | 9 | D | A | 2 | E | 1 | 3 | 8 | C | 6 | 0 | 7 | F |
| D | 0 | E | 7 | B | 3 | 4 | 6 | 2 | A | 9 | F | 8 | 1 | C | 5 |
| F | 1 | 6 | 8 | 0 | C | 5 | 7 | 4 | D | B | E | A | 9 | 2 | 3 |
| C | 5 | 9 | 0 | 4 | 6 | B | 8 | A | E | 7 | D | 2 | 3 | F | 1 |
| A | 2 | D | 3 | 7 | 1 | F | C | 6 | 4 | 0 | 8 | B | E | 5 | 9 |
| E | 6 | 1 | 4 | 9 | D | A | 3 | B | F | 2 | 5 | 7 | 8 | 0 | C |
| 7 | 8 | F | B | 5 | E | 0 | 2 | C | 9 | 3 | 1 | D | 4 | A | 6 |
| 3 | 9 | 0 | 2 | C | F | 7 | A | 5 | 8 | D | B | E | 6 | 1 | 4 |
| 5 | C | 7 | A | E | B | 3 | 9 | F | 1 | 6 | 4 | 0 | 2 | 8 | D |
| 6 | F | 4 | 1 | 8 | 0 | D | 5 | E | 2 | A | 9 | C | B | 3 | 7 |
| 8 | E | B | D | 2 | 4 | 6 | 1 | 3 | 0 | C | 7 | F | 5 | 9 | A |

*The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.*

elektor store

elektor.com

# The Elektor Store

## Never expensive, always surprising

The Elektor Store has developed from a community webshop for Elektor's own products like books, magazines, kits and modules, into a mature online store specialized in offering surprising electronics. Elektor offers the products the lab and editorial staffers are enthusiastic about or simply want to try out. If you have a suggestion for product, we're all ears (sale@elektor.com). **Our main conditions: never expensive, always surprising!**

This edition's featured promotion


## Elektor Raspberry Pi Electronics Kit

Price: €129.95

**Member Price: €116.96**

www.elektor.com/19187

## Picade Desktop Retro Arcade Machine (10“ Display)

Price: €259.00

**Member Price: €233.10**

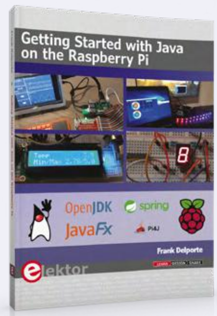
www.elektor.com/19275

## Raspberry Pi High Quality Camera Module

Price: €59.95

**Member Price: €53.95**

www.elektor.com/19279




## Getting Started with Java on the Raspberry Pi

Price: €34.95

**Member Price: €31.46**

www.elektor.com/19292

## Qoitech Otii Arc – Power Supply, Power Meter and Data Acquisition

Price: €639.00

**Member Price: €575.10**

www.elektor.com/19270

## Ringo – Educational DIY Mobile Phone Kit

Price: €144.95

**Member Price: €130.46**

www.elektor.com/19269

# Artificial Intelligence for Beginners (2)

## Neural networks with Linux and Python

**Walter Trojan**

The first article of this small series [1] introduced the hardware of the Maixduino and showed how to program it in C++ with the Arduino IDE. The performance of the processor was demonstrated using an AI model for object recognition. How artificial intelligence works in deep learning will be discussed in this installment. If you want to seriously deal with AI, Linux and Python are essential. But with the appropriate tools there is no magic wand.

Please do not expect to have fully grasped Deep Learning after reading the following sections. To do so, I had to study a few books and tutorials as well as numerous programming attempts (and I still don't know much). But I would like to give you at least an overview which structures and methods are used to provide a universally applicable program in the form of a Neural Network (NN) with specialized intelligence.

### Structure of a neural network

A neural network (NN) consists of several layers, each of which has several nodes. In **Figure1** the nodes of a layer are arranged vertically. Theoretically, there can be any number of nodes per layer and any number of layers in the network. The architecture is determined by the respective task and the size is of course also dependent on the resources of the computer platform used. For example, if the NN is to classify objects that it receives by picture, the first layer takes over the data for solving the task and provides a corresponding number of input nodes for acquisition. If a low-resolution image consists only of 28 * 28 pixels, then 784 nodes are required for a display in grey values. For the acquisition of colour images, the number would triple. During image acquisition, each input node receives the gray value of "its" pixel.

The result of the image analysis is presented in the output layer, which has a separate node for each result. If the NN were to recognize 1000 objects, this layer would have the same number of output nodes. Each node shows a probability between 0 and 1.0 and thus indicates how reliable NN is in the result it has found. For example, when a house cat is taken, the associated node could have a value of 0.85 and the "Tiger" node a value of 0.1. All other nodes would then have even lower probabilities and the result would be unambiguous.

The actual analysis work is done by the hidden layers during deep learning. Depending on

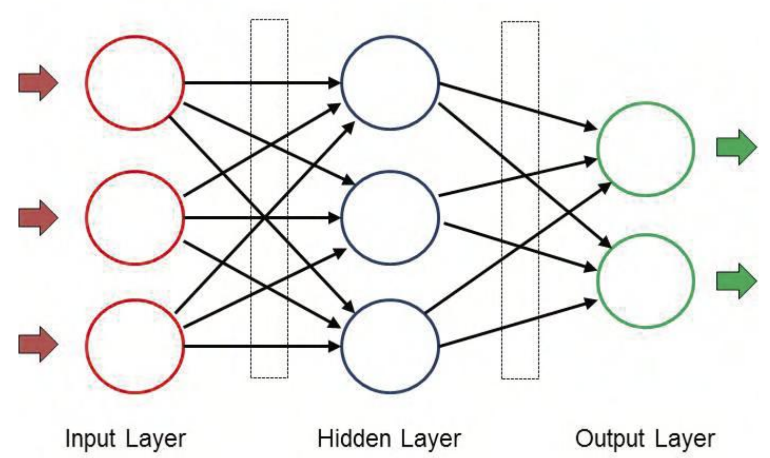



*Figure 1: Structure of a neural network.*

the task, any number of hidden layers can be used; in practice, there are certainly 100 to 200 such layers in use. In addition to the architecture shown above, more complex structures with feedback or intermediate filters to increase accuracy can be found. The number of nodes in each hidden layer is also arbitrary. In the above-mentioned example for object detection, three hidden layers with 200 nodes each could already provide useful results.
If in the human brain intelligence is achieved by linking neurons by means of synapses, the nodes in the NN are also networked with each other. Each node of a layer is connected with all nodes of the following layers. Each connection (shown as an arrow in the picture) contains a value called weight. These weights between the layers are stored in matrices. Therefore, in the field of deep learning, programming languages are preferred where matrix operations can be performed easily and quickly.
So how does an NN achieve the desired results? Each node receives its input signals from all nodes of the layer in front of it, marked *x* in **Figure 2**. These values are multiplied by the weight values *w* and added, so the net input
net = *x*1**w*1 + *x*2**w*2 + *X*3**w*3 ...
The calculated value of net is now multiplied by an activation function and sent to the output (and thus to all nodes of the subsequent layer). A node-specific threshold value also determines whether this node "fires".
The activation functions shown in Figure 3 are assigned to the layers and ensure that the output values remain within the desired range. For example, the Relu function suppresses all negative values and Sigmoid has a limitation between 0 and 1. These precautions ensure that NNs operate in a clear numerical range and cannot be dominated by " runaways".

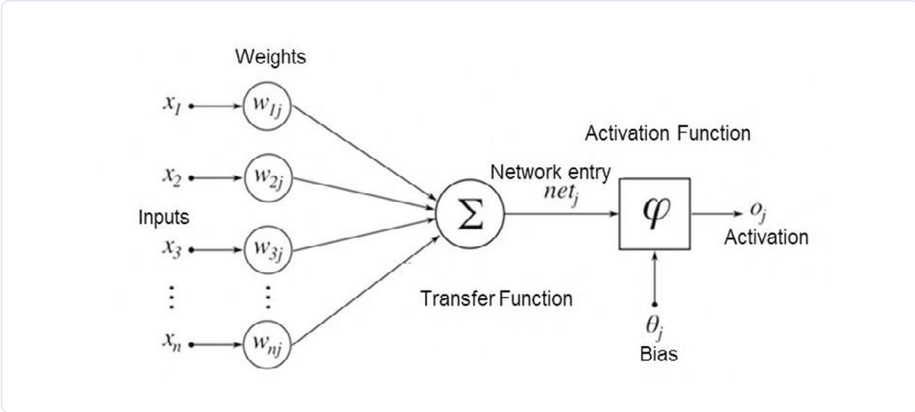



*Figure 2: calculations within a node (https://commons.wikimedia.org/wiki/Artificial_neural_network).*

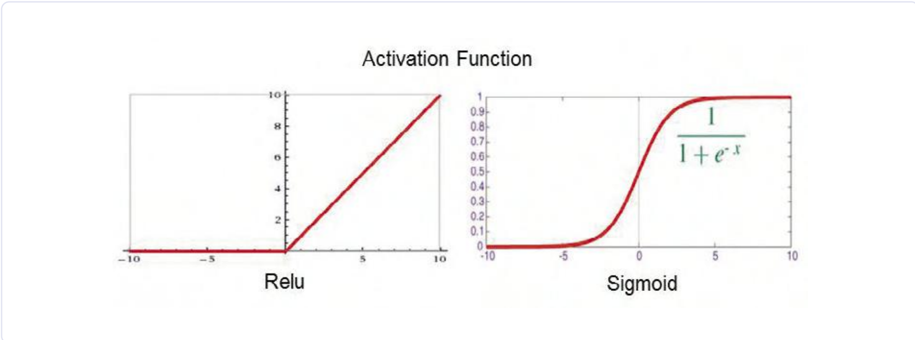



*Figure 3: Frequently used activation functions.*

## Training

So after the input data has been recorded, NN performs numerous calculations in each shift and presents the results at the output. This process is called inference. And how does the intelligence get there? This is done like a small child through education and training. With an untrained NN, the weights are usually filled with random numbers in the value range -1 to +1, so the network is dumb and therefore only provides random results. For training, data and known target output values are fed to the network. After calculation with these values, the result is compared with the setpoint and the difference is documented by a loss function. Now follows the learning process, also called backpropagation. Starting in the last layer, the weights are adjusted in small steps to minimize the loss. After many (often millions of times) of training rounds with different input data, all weights have the values appropriate to the task and NN can now analyze new data unfamiliar to it.
A particularly suitable architecture for image and audio processing is the *Convolutional Neural Network*, which is also supported by Maixduino. In such a CNN, the neurons (at least in some of the layers) are arranged two-dimensionally, which fits to two-dimensional input data such as an image. Compared to the network indicated in Figure 1, where the activity of a neuron depends on all neurons of the previous layer (via different weighting factors), the dependency is simplified and locally limited in the case of CNN. Here, the activity of a neuron depends only on the values of (for example) 3 x 3 neurons, which are located in the layer before it — the weighting factors are the same. With such a network, small-scale structures such as lines, curves, points and other patterns can be detected particularly well. In the subsequent layers, more complex details and finally whole faces are detected.
In terms of programming, an NN is similar to a spreadsheet, an arrangement of cells with predefined calculation instructions, which access multidimensional matrices with the weights during execution. When classifying new input data, all layers from input to output are passed through and the outputs reflect the respective probability of the result assigned to the node. Well-trained NNs can reach values around 0.9. During training, after the interference, backpropagation takes place with a run from the back to the input to adjust the weights.
Sounds complicated at first, but there are numerous tools and libraries available for the implementation, which I will introduce in the following paragraphs.

## Linux and Python

Already in the first article of this miniseries I pointed out that you have to be prepared for a learning effort when dealing with the topic of AI. The easiest way to do it in this world is on the Linux platform, because here most of the

tools are available for free and of good quality. Furthermore, Linux offers the same comfort as Windows, but is packaged differently (usually better). My preference is Ubuntu, which is also available as LTS version (Long Term Support) and is supported for at least 4 years. But also the other derivatives like Debian, Mint etc. are just as useful, the choice is a matter of taste. Linux can be installed in a virtual machine besides Windows, for example, so you don't need an extra computer.
And why Python? Python is an interpretive programming language, some will say, so potentially slow. This disadvantage is offset by numerous advantages: Firstly, Python does away with the frills of brackets and semicolons and performs block building by indenting the lines. It has powerful data structures such as lists, tuples, sets and dictionaries; furthermore, Python already has fully integrated matrix calculation. Other major advantages are the available AI frameworks and libraries, which can be fully or partially integrated due to their high modularity. These are mostly written in C++ and therefore have the required performance. And all this can be installed easily with just a few instructions.
If you want to get started, please install your favourite Linux as well as pip3 and Python 3, instructions for this are abundantly available on the net.

## Maixduino speaks MicroPython

To enable Python to run on systems with less memory, a lightweight version called MicroPython is available; it can be installed on platforms like Maixduino, ESP32 and others. MicroPython contains a balanced repertoire of commands and 55 additional modules for numerous mathematical and system functions. To add new versions or AI models, the flashing tool Kflash is required.

**Installation of Kflash under Linux:**

- Download version 1.5.3 or higher from web link [2] as an archived file *kflash_gui_v1.5.3_linux.tar.xz*.
- Transfer to a folder of your choice.
- Unpack by command tar xvf kflash_gui_v1.5.3_linux.tar.xz.
- Change to the newly created folder */kflash_gui_v1.5.2_linux/kflash_gui*.
- Start with *./kflash_gui* (in case of startup problems, check the box *Run file as program* in the properties of this file).

This starts the graphical user interface of Kflash (**Figure 4**) and you can now load firmware or AI models into the Maixduino.

**Installing firmware on Maixduino**

Even though the Maixduino is already equipped with MicroPython at delivery, you should always download the latest version. When writing these lines, the corresponding firmware is named v0.5.0 and can be downloaded via the web link [3]. Choose *maixpy_v0.5.0_8_g9c3b97f* or higher and in the next picture choose the variant *maixpy_v0.5.0_8_g9c3b97f_minimum_with_ide_support.bin* or higher, a file of about 700 kB, which also contains support for the MaixPy-IDE.

With the help of Kflash the new firmware is quickly installed. The Open File button is used to select it and after setting the *board, port, baud rate* and *speed mode* as in **Figure 5**, a click on Download starts the loading process. After that you can already execute the first Python commands with a terminal emulator (e.g. Putty) via the port /dev/ttyUSB0 on the Maixduino. Here is a small example with array commands:

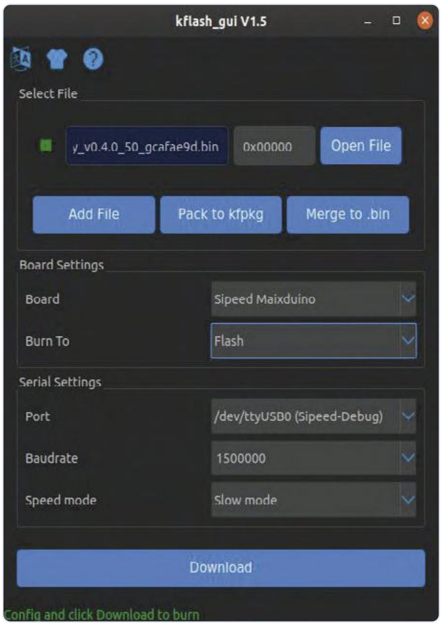



Figure 4: Kflash shell.

```
>>>                              # Python-Prompt

>>> import array as arr          # Import Array Module
>>> a = arr.array('i',[1,2,3])   # Create array a with integers
>>> b = arr.array('i',[1,1,1])   # Create array b with integers
>>> c = sum(a + b)               # Create sum of all array values
>>> print(a,b,c)                 # and output it
array('i', [1, 2, 3]) array('i', [1, 1, 1]) 9     # output
>>>
```

With libraries like *numpy* (under Linux) or *umatlib* even more functions are available.

**Installation of MaixPy-IDE**

Development and tests are much more comfortable with the development environment called MaixPy-IDE. Python programs can be developed and tested as well as loaded and executed on the Maixduino. Additionally, as shown in **Figure 5**, tools for image analysis are available. The installation is performed as follows:

- Download the version *maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z* or higher from web link [4].
- Transfer to a folder of your choice.
- Unpack by the command *tar maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z.*
- Change to the new folder *maixpy-ide-linux-x86_64-0.2.4-installer-archive* and enter the following commands:

```
./setup.sh
./bin/maixpyide.sh
```

After that the IDE starts. For subsequent starts, only the latter command is of course required.

Thus, all tools for implementing AI models are available. In the following a face recognition is tested.

## Face recognized!

An trained AI model is used for face recognition, which has already analyzed several thousand faces for characteristic features - the weights of the NN used are adjusted accordingly. The model is available for download under the web link [5] under the name face_ model_at_0x300000.kfpkg. The basis of the development is the AI framework Yolo2 (You Only Look Once), which divides the image objects into several zones, analyzes them separately and thereby achieves high recognition rates (I will talk about the AI frameworks later in this series). The AI model for the AI processor is packed in kfpkg format and must be flashed to address 0x300000 in the Maixduino. This can also be done with Kflash; just find the file using Open File and load it onto the board using the parameters from Figure 4.

The MaixPy-IDE is used for comfortable handling of the Python script. With this IDE you can develop and test programs and transfer them to Maixduino. **Figure 5** shows the interface divided into three windows:

- **Editor,** top left: In this area the program input is done with syntax highlighting.
- **Terminal**, below: Display of the program output.
- **Image Analysis,** right: This is where the display of images and their spectral division into the colours red, green and blue is performed.

Among other available buttons and menu items, the two buttons on the lower left are important: The "paperclip" is used to establish (colour green) or disconnect (red) the connection to the Maixduino via the port "ttyUSB0". A green triangle below it starts the script; then this button changes to a red dot with "x" and serves to stop the program.

To perform the test, I printed the faces of two publicly known people (Albert Einstein and Rudi Völler) and pinned them to a wall. The selection was purely coincidental, but both faces are said to resemble somewhat, which is not confirmed by me. When taking these pictures, the faces were immediately recognized and marked with a frame. It is important to make sure that the images are displayed in landscape format as shown, otherwise the recognition rate drops noticeably.

The program *Face-detect.py* can be found in the download folder on the Elektor website [6]. Its brevity again reflects the performance of the libraries used. To start, the required libraries for camera, LCD and KPU are integrated and initialized. After that, the NN is loaded into the KPU starting at address 0x300000. When the AI model is initialized by the command kpu.init_yolo2, additional constants are transferred for setting the accuracy and optimization. Now the image classification is carried out in an endless while loop, an image is taken and fed to the NN. If faces have been detected, the variable i receives for each face the coordinates and size of a marker frame, which is then drawn into the image. Finally, the image (on the LCD panel) and the marking data (on the serial terminal) are output. More details about the KPU commands can be found under the link [7].

In the MaixPy-IDE the picture is also shown in the upper right corner and the corresponding colour spectrum is displayed below. If you do not need this information, the right image window can be switched off with the deactivate button.

For better handling I mounted Maixduino and LCD on a small board and aligned the camera to the front (see **Figure 6**). This makes it easy to capture and analyze real faces, printed images or screen contents.

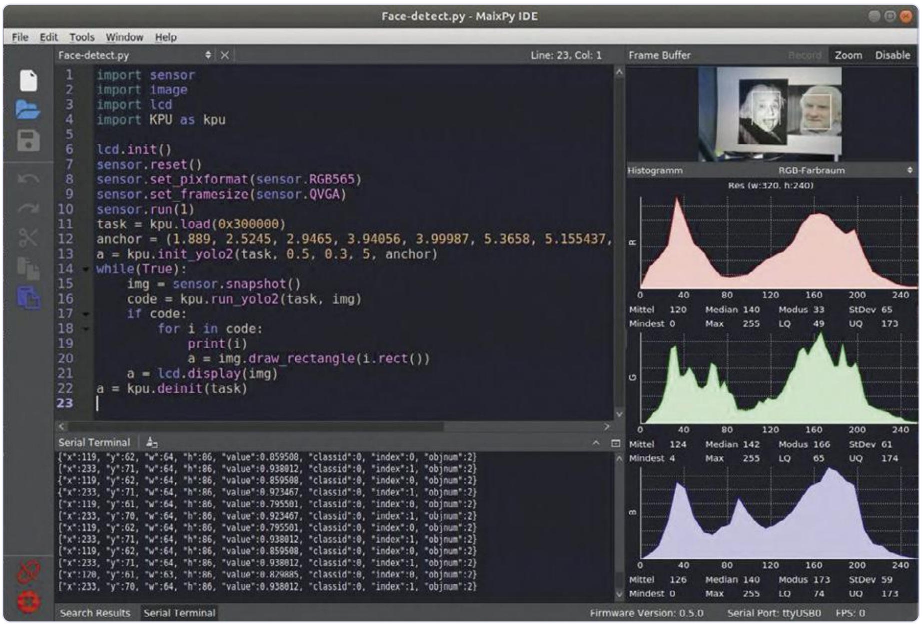



*Figure 5: User interface of the MaixPy-IDE.*

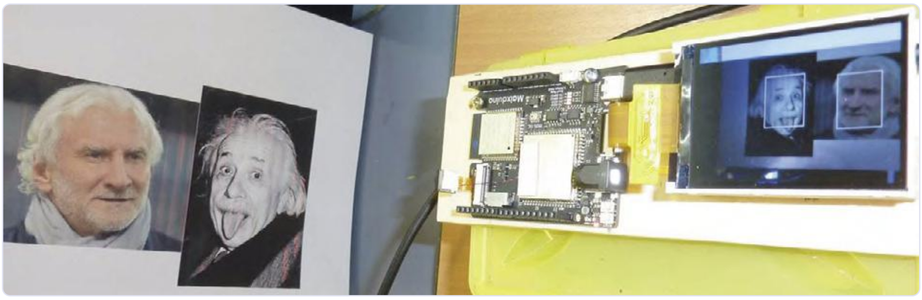


*Figure 6: Test setup for face recognition.*

You can see the display on the LCD panel in **Figure 7**, similarities of the persons are not visible.

But what is behind this Yolo2 model? The neural network has 24 convolutional layers and two fully connected output layers (see **Figure 8**). In between there are some maxpool layers as filters to remove complexity and to reduce the tendency to "memorize". It is noticeable that a window size of 3x3 is largely used for detail recognition. Exactly this is supported by the KPU hardware, which ensures that the Maixduino is highly efficient for such tasks.

Other known NN structures even have several hundred layers, have feedback paths or other extras. There are no limits to creativity in this area, much depends on the budget, i.e. computing power.

*Figure 7: Representation on the Maixduino LCD.*

### And on we go!

The powerful hardware and the already available software environment show that the Maixduino is well suited for the entry into Artificial Intelligence. Due to its low power consumption it is well suited for use in mobile devices with already trained neural networks. In a third part of the series I will show you how to develop, train and execute your own neural network. The interface to the developer is the AI framework Keras, which is also known as a comfortable “Lego construction kit for AI”. You will also get hints on how to program ESP32 on the board - for example to acquire analogue values.

Stay curious!

200023-B-04

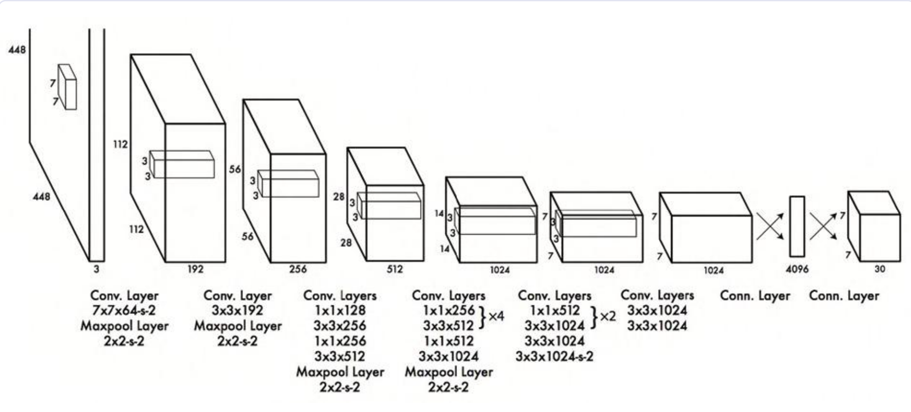
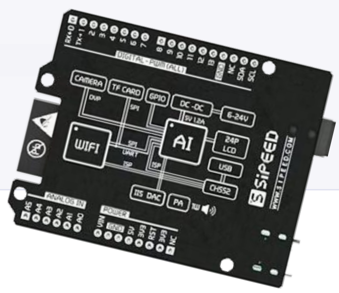



*Figure 8: Network architecture for facial recognition (source: https://bit.ly/3cK3DUR).*

**4 SALE @ WWW.ELEKTOR.COM**

› **Sipeed MAix BiT Kit for RISC-V AI+IoT**
www.elektor.com/sipeed-maix-bit-kit-for-risc-v-ai-iot



## WEB LINKS


[1] **AI for Beginners, Elektor May & June 2020**: www.elektormagazine.com/200023-02

[2] **Kflash**: https://github.com/sipeed/kflash_gui/releases

[3] **Maixduino Firmware**: http://dl.sipeed.com/MAIX/MaixPy/release/master/

[4] **MaixPy IDE**: http://dl.sipeed.com/MAIX/MaixPy/ide/_/v0.2.4/maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z

[5] **AI models**: http://dl.sipeed.com/MAIX/MaixPy/model

[6] **Project software**: www.elektormagazine.com/200023-B-04

[7] **KPU Commands**: https://maixpy.sipeed.com/en/libs/Maix/kpu.html


PROTEUS
DESIGN SUITE
High Speed Design Features
• Differential Pair Routing
• Length Matching / Net Tuning
• Automatic Phase Matching
• Use for USB, Ethernet, DDR3 etc.
Performance without the price premium.
Find out more and
configure your package on our
website or call the team on (+44)1756 753440
labcenter
Electronics
www.labcenter.com