IoT Home Hacks

with ESP8266

Hans Henrik Skovgaard

an Elektor Publication

LEARN DESIGN SHARE

● This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
78 York Street
London W1H 1DP, UK
Phone: (+44) (0)20 7692 8344
© Elektor International Media BV 2020
First published in the United Kingdom 2020

● British Library Cataloguing in Publication Data
Catalogue record for this book is available from the British Library

● **ISBN 978-1-907920-83-7**
EISBN 978-3-89576-386-1
EPUB 978-3-89576-387-8

Prepress production: DMC ¦ dave@daverid.com

Cover Art: Marie Skovgaard Mikkelsen

Printed in the Netherlands by Wilco

Table of Contents

Chapter 1 ● Introduction

Welcome to the "ESP8266 IoT"-book

Disclaimer: The constructions I describe in this book have been tested as thoroughly as possible. I have not managed to damage any of my equipment (or myself), but I am sure there's a way. In addition, the technology described hereinafter is purely for fun and should not in any circumstances be used for medical, mission-critical or safety-critical applications.

I will not accept any responsibility for damages of any kind due to actions taken by you after reading this book.

Having said that, let's continue.

1.1 ● Why this book

Like many others, when I first laid my eyes on the ESP8266 it captured my imagination immediately, like when the Raspberry Pi came along.

Before the arrival of the ESP8266, I made a lot of Arduino constructions, some with Internet connections. It was therefore obvious that I should have a go with ESP8266.

Some of the constructions and findings are described in this book. It is my hope that you enjoy working with ESP8266 as much as I have.

1.2 ● Who the book is designed for

In order to use this book, I expect some basics from you:

- **You have some experience with PC software.** That is, you are familiar with operating at least a Windows PC and know how to navigate a file system, edit a file and install/configure software.

- **You have experience in surfing the Internet.** You need to be able to locate and download the files I mention in the book. After download, you should be able to find and install the downloaded files.

- **You know how to assemble a PCB.** Since many of the constructions need additional hardware compared to the ESP8266, you will need to be able to assemble the required PCBs. They are not complicated and I have avoided using SMD components.

- **You have some basic mechanical skills.** Most of the constructions need to be incorporated into present mechanical constructions. It shouldn't be difficult it requires that you operate tools which may cause injury.

Please be careful about what you are doing and take care!

1.3 ● How to use the book

Even though this book is about the ESP8266 microcontroller, you may need to install and configure an embedded Linux system (a Raspberry Pi).

Some basic skills are required to do so, but everything will be described in details, or through links to additional information on the internet.

When configuring Linux, you will see specific commands listed like this:

```
pi@raspberrypi: $  cd /var/www/emoncms/Modules
pi@raspberrypi: $  git clone https://github.com/emoncms/app.git
```

Commands are in **bold.**
The last command is listed on two lines but you should enter it as one long command like:

```
“git clone https://github.com/emoncms/app.git”
```

Throughout the book there will be parts which will be highlighted as shown below:

Warning: This is what a warning will look like. This is where you or your surroundings or the equipment may be damaged if not handled with care and attention: Don’t worry there will not be many of these.

Note: This is how a note looks. You should pay attention to these.

Comment: This is where I may “side-track” a little and give additional information for further reading on the subject.

Files of interest will be shown like this:
`/opt/lamp/htdocs`

Finally, I will show file listings in boxes like the following:

```
#include <stdio.h>
main()
{
  printf(“hello world\n”);
}
```

I encourage you to go through the entire book before you start your own project. It is always good to be well prepared. You should at the very least read a section in advance.

I also encourage you to make notes about what you are doing, just in case you need to do it all over again...I talk from experience.

1.4 ● Organisation of the book

In order to avoid reading the entire book to discover what's covered, here is a very short description of each section.

1.4.1 ● Section 2: Components

This book is mainly about the Espressif ESP8266 microcontroller. I have chosen the Wemos D1 mini pro, but you will most likely be able to use other variants out there.

Since the ESP8266 and Raspberry Pi complement each other very well, there will also be design ideas which incorporate the Raspberry Pi.

In this section, the following components are briefly described:

- Adafruit Neopixel Jewel 7.
- Raspberry Pi
- Grönö lamp
- SSR
- Maxim DS18B20
- QRE1113 (Analog)

1.4.2 ● Section 3: Arduino IDE Installation

This section will guide you through the installation of the Arduino IDE. In order to compile software for ESP8266, you will need to install additional libraries.

If you are familiar with the Arduino IDE there should be no surprises here.

1.4.3 ● Section 4: emoncms Installation

I have chosen to use emoncms for logging and visualisation.
This section describes how to install emoncms on a Raspberry Pi.

In order to make emoncms work, additional software packages will have to be installed but all will be described here.

1.4.4 ● Section 5: MQTT (mosquitto) Installation On A Raspberry Pi

The installation will cover the basic MQTT broker but also a publisher client and subscriber client.

If in favour of it, a Windows MQTT client will be installed as well.

1.4.5 ● Section 6: ESP8266 Security

A very short walkthrough of why I, so far, have not secured my ESP8266 devices.

1.4.6 ● Section 7: ESP8266 Watchdogs

A description of various watchdog types available within ESP8266.

These watchdogs can be included in your designs or left out. It is up to you how critical your system is.

1.4.7 ● Section 8: WiFi Antennas

A walkthrough of the external WiFi antenna possibilities within the Wemos design. A description of a very simple DIY antenna will also be provided.

1.4.8 ● Section 9: General Software Design

This section describes how I have structured the ESP8266 Arduino design. It is here merely for you to understand what is going on. It is up to you to follow it or redesign it.

1.4.9 ● Section 10: IKEA Hack

It's called an "IKEA hack" because a Neopixel Jewel 7 is placed inside a Grönö lamp and controlled by an ESP8266. It is a cheap and easy solution for housing. Other lamps can also be modified.

1.4.10 ● Section 11: Door Lock Monitor

With the help of a QRE1113 and ESP8266 it is possible to monitor the status of a door lock – or any other moving object 1 mm away from the QRE1113.

The status of the door lock is stored in a central location so the information can be used for surveillance.

1.4.11 ● Section 12: Energy Sensor

This design is hooked up to an S0 interface on a power consumption meter. Since the S0 interface outputs a specific amount of pulses per kWh, it is possible to measure the power consumption of your household.

Data will be gathered at a central server, which in this case is a Raspberry Pi.

1.4.12 ● Section 13: Refrigerator Control

This section will describe a small control device that can keep a modified refrigerator at a constant temperature of between 4ºC and the surrounding ambient temperature.

It includes a small WEB interface and log functionality.

1.4.13 ● Section 14: Lamp Control

This is a simple construction consisting of a Wemos PCB with a small relay PCB on top. You can switch any 230V equipment, but I have used it for a small lamp.

1.4.14 ● Section 15: Troubleshooting

Includes my findings during the development of the constructions and the writing of this book. Most likely it will not solve all of your problems.

1.4.15 ● Section 16: PCB Layout

For most of the constructions mentioned in this book, a small PCB has been put together in order to make the constructions easier to build.

The PCB layout will be shown here and the Gerber files will most likely be available for download on Elektor's homepage.

1.5 ● Abbreviations

In this book, I use abbreviations, some of which may not be well known. Below is a list with additional explanation:

Abbreviations	
Abbreviation	**Description**
RPi	Raspberry PI
DIY	Do-It-Yourself (Sometimes with a good result)
PCB	Printed Circuit Board
GCC	GNU Compiler Collection
LED	Light Emitting Diode
RGB	Red, Green, Blue
cURL	A computer software project providing a library and command-line tool for transferring data using various protocols.
GIT	A distributed revision control system developed by Linus Torvalds
RTS	Real Time Clock.
DTIM	Delivery Traffic Indication Message.

1.6 ● References

I would not have been able to write this book without the Internet. If you know where to look, you can find answers to almost every question.

Since much of my findings are the result of many peoples hard work, I also want to give them credit and mention them here.

You will also find testimonials in my code either to the borrowed code or to where I got help to carry on.

For your convenience, there is a file together with the web software containing the following links (it is, after all, easier to cut and paste instead of just pasting).

References	
Ref	**Description**
1	**Adafruit Neopixel 7:** https://www.adafruit.com/product/2226 Arduino Library Use. https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use
2	**Wemos:** https://wiki.wemos.cc/products:d1:d1_mini_pro **Arduino support for Wemos:** https://wiki.wemos.cc/tutorials:get_started:get_started_in_arduino
3	**Grönö lamp:** http://www.ikea.com/us/en/catalog/products/10373221/ http://www.ikea.com/us/en/assembly_instructions/grono-table-lamp__AA-1952499-1.pdf
4	**Aliexpress ABS plastic enclosures case:** https://www.aliexpress.com/item/5pcs-lot-small-IP54-ABS-plastic-enclosures-case-for-PCB-with-cut-out-service-47-37/32753602240.html?spm=a2g0s.9042311.0.0.hX9DYB
5	**cURL:** https://en.wikipedia.org/wiki/CURL, https://curl.haxx.se/ Note: The following "curl" is not related: http://www.curl.com/ it is a different programming language in its self.
6	**Arduino IDE:** https://www.arduino.cc/en/Main/Software
7	**Watchdog references:** **Reverse engineering of the ESP8266 watchdog timer:** https://mongoose-os.com/blog/esp8266-watchdog-timer/ **Arduino Sketch Managed ESP8266 Watchdog:** https://www.sigmdel.ca/michel/program/esp8266/arduino/watchdogs2_en.html **Watchdog Timers:** https://m.eet.com/media/1175014/f-murphy.pdf

1.7 ● Feedback

As I value any feedback about what I do in order to improve, I encourage you to send me comments, improvements or any questions for help.

I will try and answer as soon as possible – usually within one week.

Please e-mail me at hhs@hask.dk

Chapter 2 ● Components

This section briefly describes each of the components or boards used in this book.
It will only be a short description. If more information is required you should look into data sheets for each component or board.

Please note that component descriptions are based on available information at the time of writing. Things change rapidly in this business so some of the descriptions may be obsolete at the time you read this book.

I have tried to build the constructions in a way that it will be possible to upgrade to the latest version.

2.1 ● ESP8266

2.1.1 ● Wemos D1 mini Pro

The controller board chosen for this project is the Wemos D1 mini Pro.

It can be seen as an Arduino clone but it is:

- Rather cheap (5 US$)
- Small (length: 34.2mm, width: 25.6mm, Weight: 2.5g)
- Has built-in WiFi.

It is therefore perfect as an IoT controller – but that is another story.

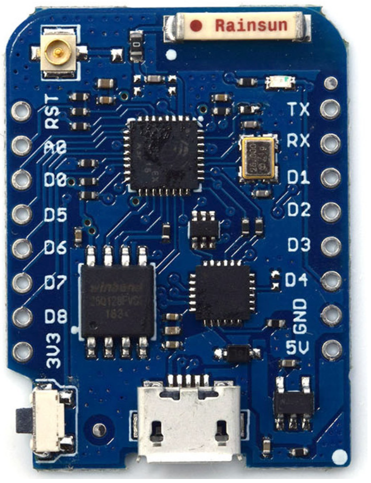


Figure 1 - Wemos D1 Mini Pro

Throughout this book, you can use an ordinary Arduino – with or without a WiFi shield - if you don't like the Wemos controller.

The software has not been tested on an ordinary Arduino. It should, however, be possible to make the software run – with modifications.

The homepage for the Wemos D1 mini pro V1.1.0 can be found here:
https://wiki.wemos.cc/products:retired:d1_mini_pro_v1.1.0

Please note that at the time of writing this version v1.1.0 has been retired and replaced by version 2.0.

This board is larger and will not fit the casings presented in this book. However, the I/O pins are the same, so if you like, you may use version 2.0.

2.1.2 ● Sleep Mode

During the description of the constructions in this book, it should be noted that construction/software is not optimised for Sleep Mode.

In the document:
https://fwww.espressif.com/sites/default/files/documentation/9b-esp8266-low_power_solutions_en.pdf
The following can be found regarding Sleep Mode:

'ESP8266 series chip provides 3 configurable sleep modes. We also provide some low power solutions related to these sleep modes. Users can choose and configure the sleep mode as required.'

The 3 sleep modes are:

- *Modem-sleep*
- *Light-sleep*
- *Deep-sleep*

The differences between Sleep Modes can be seen in the below table:

ESP8266 Sleep Mode differences				
Item		**Modem-sleep**	**Light-sleep**	**Deep-sleep**
WiFi		OFF	OFF	OFF
System clock		ON	OFF	OFF
RTC		ON	ON	ON
CPU		ON	Pending	OFF
Substrate		15 mA	0.4 mA	~ 20 μA
Average current	DTIM = 1	16.2 mA	1.8 mA	-
	DTIM = 3	15.4 mA	0.9 mA	-
	DTIM = 10	15.2 mA	0.55 mA	-

As shown in the table there are power savings to achieve if the right Sleep Mode is selected and the software is designed for it. This is important if you power your device using a battery.

In each case, you should carefully investigate what fits your needs and how to bring the ESP8266 into Sleep Mode and equally important, how to bring the ESP8266 out of Sleep Mode.

2.2 ● Raspberry Pi

Figure 2: Raspberry Pi 3 Model B

This is not a book about Raspberry Pi (RPi) itself. I will only show the basics about what is required in order to get the Raspberry Pi working as a server.

Fortunately, there is lots of documentation on the internet about Raspberry Pi, which you

may consult if you want to know more in depth.
One such place is, of course, the Raspberry Pi homepage:
https://www.raspberrypi.org/

or Wikipedia:
https://en.wikipedia.org/wiki/Raspberry_Pi

The latter gives a good overview of its history and evolution.

For the constructions in this book, I recommend using the Raspberry Pi 2 model B ver 1.2 or later.

The Raspberry Pi should not be hard to obtain, either from the Internet or in a local store near you.

2.2.1 ● Raspberry Pi Configuration

In order to get the Rasperry Pi up and running, you will need some software: the operating system (OS) - installed. My preferred OS for Rasperry Pi is found here:
https://www.raspberrypi.org/downloads/raspbian/

At the time of writing the release is called “Stretch”. This may have changed by the time you are reading this. Former names have included “Wheesy” and “Jessie”. Please keep this in mind when looking for help on the Internet. There may be differences between the releases.

The recommended installation process can be found here:
https://www.raspberrypi.org/documentation/installation/

It should be fairly straight forward as long as you distinguish between “.zip” files (the way images are packed to save space) and “.img” files (the actual image to be written to the SD-card).

When Raspbian is copied to the SD-card you will need to enable SSH as we will use the Rasperry Pi in so-called “headless mode”. That is, without monitor, mouse, or keyboard.

How to enable SSH can be found here:
https://www.raspberrypi.org/documentation/remote-access/ssh/

The following is found under section 3:

'For headless setup, SSH can be enabled by placing a file named ssh, without any extension, onto the boot partition of the SD card from another computer. When the Pi boots, it looks for the ssh file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text or nothing at all.
If you have loaded Raspbian onto a blank SD card, you will have two partitions. The first

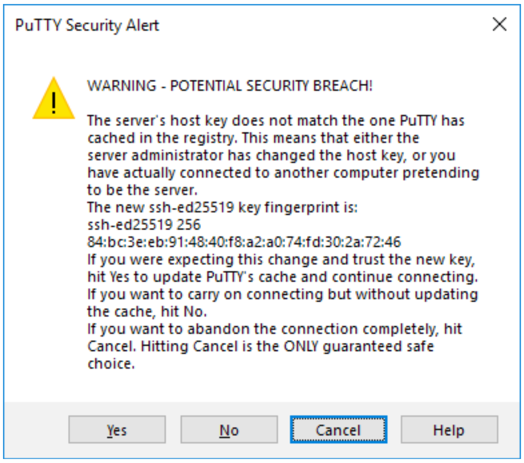
one, which is the smaller one, is the boot partition. Place the file into this one.'

You are now ready to place the SD-card in the Rasperry Pi, connect to your home network, on the same LAN as your PC, and power up the Rasperry Pi for the first time.

I use PuTTY to connect to my Raspberry Pi:
    https://www.putty.org/

You should however, be able to use any SSH client.

When you connect to your newly configured Raspberry Pi for the first time you will get the following security alert:


PuTTY Security Alert

WARNING - POTENTIAL SECURITY BREACH!

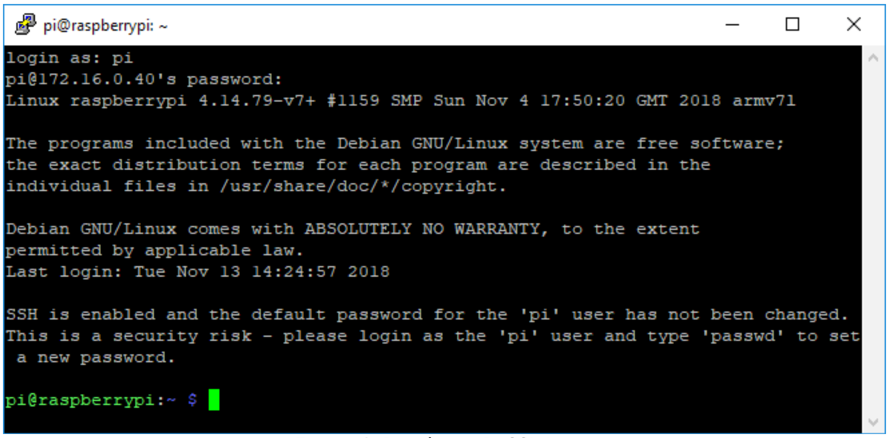
The server's host key does not match the one PuTTY has cached in the registry. This means that either the server administrator has changed the host key, or you have actually connected to another computer pretending to be the server.
The new ssh-ed25519 key fingerprint is:
ssh-ed25519 256
84:bc:3e:eb:91:48:40:f8:a2:a0:74:fd:30:2a:72:46
If you were expecting this change and trust the new key, hit Yes to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating the cache, hit No.
If you want to abandon the connection completely, hit Cancel. Hitting Cancel is the ONLY guaranteed safe choice.

Figure 3: SSH Warning During First Raspberry Pi Login

Answer Yes to continue.

Since this is the first time you log in you should:
    login as: pi, with
    password: raspberry

The below figure shows how it may look:



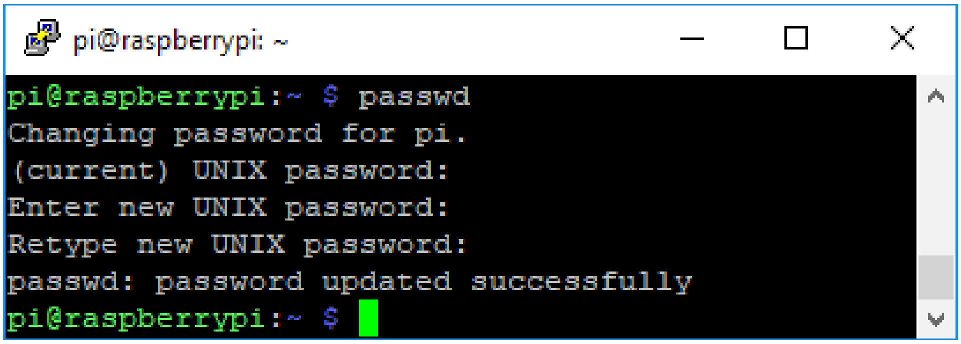
```
login as: pi
pi@172.16.0.40's password:
Linux raspberrypi 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 13 14:24:57 2018

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $
```

Figure 4: Raspberry Pi SSL Login

The first thing to do after successfully logging in is to change the default password. This is achieved by using the following command:

```
pi@raspberrypi:~ $ passwd
```

The process is shown in the next figure:



```
pi@raspberrypi:~ $ passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $
```

Figure 5: Raspberry Pi Password Change

Enter the current password (raspberry). Also, enter the new password and retype the new password for confirmation.
Keep the password in a safe location for future use.

There are other ways of making a “headless” connection to the Raspberry Pi:

Xming: https://sourceforge.net/projects/xming/
VNC: https://www.raspberrypi.org/documentation/remote-access/vnc/

...just to mention a few.

2.3 ● Neopixel Jewel 7

From Adafruit’s homepage:

'Be the belle of the ball with the NeoPixel Jewel! We fit seven of our tiny 5050 (5mm x 5mm) smart RGB LEDs onto a beautiful, round PCB with mounting holes and a chainable design to create what we think is our most elegant (and evening-wear appropriate) NeoPixel board yet.'

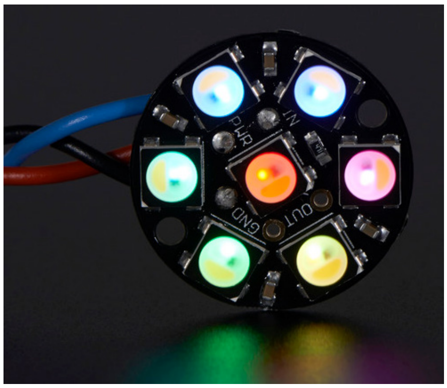


Figure 6: Adafruit NeoPixel Jewel 7

As mentioned, the Adafruit Neopixel Jewel consists of seven individually controllable smart RGB LEDs.

Each RGB LED has an integrated driver, which makes them controllable from a microcontroller, with the right software.

Schematics of the Adafruit Neopixel Jewel can be seen below:

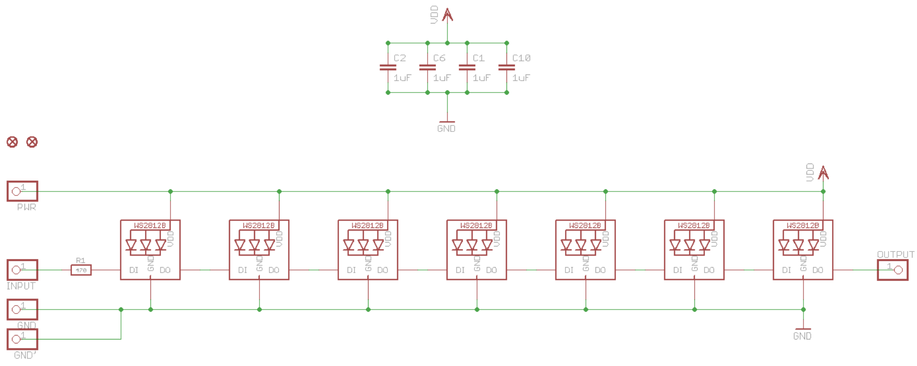


Figure 7: Adafruit NeoPixel Jewel 7 Schematics

In this construction, we use the PWR, GND and INPUT pins. You can put several Jewels in serial by connecting the OUTPUT from one to the next jewels INPUT.

The Jewel cannot be operated by its own: it needs some kind of controller to send instructions to each RGB LED.

2.4 ● IKEA Grönö lamp

I looked at several options on how to make a lamp, where I could house my Wemos/ NeoPixel. As usual, I may say, IKEA came to my rescue.

I wanted something that could soften the colours by the NeoPixel and look nice at the same time. My choice so far is the IKEA Grönö lamp.

I don’t benefit from mentioning IKEA here.

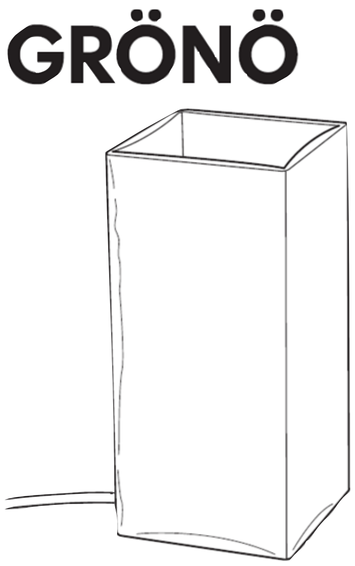


Figure 8: IKEA Grönö Lamp

It is very easy to modify the bulb socket so it can hold the NeoPixel jewel. Also, the lamp is cheap...at least in Denmark.

2.5 ● SSR

The following description of an SSR has been taken from
https://en.wikipedia.org/wiki/Solid-state_relay

'A solid-state relay (SSR) is an electronic switching device that switches on or off when a small external voltage is applied across its control terminals. SSRs consist of a sensor which responds to an appropriate input (control signal), a solid-state electronic switching device which switches power to the load circuitry, and a coupling mechanism to enable the control signal to activate this switch without mechanical parts. The relay may be designed to switch either AC or DC to the load. It serves the same function as an electromechanical relay but has no moving parts.'

'Packaged solid-state relays use power semiconductor devices such as thyristors and transistors, to switch currents up to around a hundred amperes. Solid-state relays have fast switching speeds compared with electromechanical relays, and have no physical contacts to wear out. Application of solid-state relays must consider their lower ability to withstand momentary overload, compared with electromechanical contacts, and their higher "on" state resistance. Unlike an electromechanical relay, a solid-state relay provides only limited switching arrangements (SPST switching).'

The SSR used in this book can be seen here:
https://www.sparkfun.com/products/13015

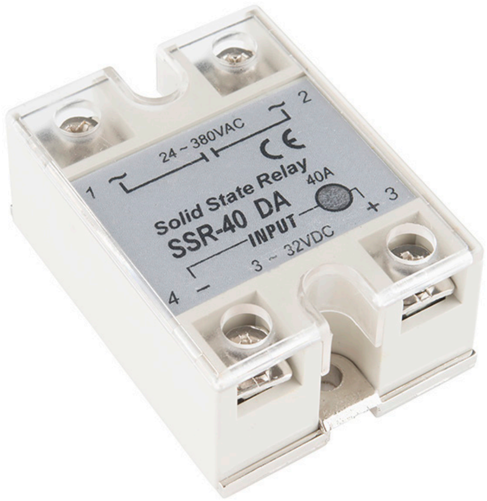


Figure 9: SSR from Sparkfun

The chosen SSR can switch current loads of up to 40A with a 3-32V DC input.

It is clearly overrated but then a heatsink is not necessary.

I don't have any preferences for selecting an SSR. Any similar device will do.

2.6 ● Maxim DS18B20

The temperature sensor used in this book is the DS18B20 from Maxim. The only reason for this is that I have used it a lot in other constructions.

The datasheet can be seen here:
https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.
Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus.

2.7 ● QRE1113 (Analog)

This is a small line sensor breakout board, with an easy-to-use analog output, which will vary depending on the amount of IR light reflected back to the sensor.

The board can be seen in the figure below (Size: 7.62 x 13.97 mm)

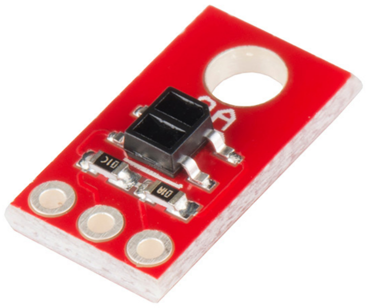


Figure 10: QRE1113 (Analog)

More information can be found here:
https://www.sparkfun.com/products/9453

Below is a description taken from the homepage:

'The board's QRE1113 IR reflectance sensor is comprised of two parts - an IR emitting LED

and an IR sensitive phototransistor. When you apply power to the VCC and GND pins the IR LED inside the sensor will illuminate. A 100Ω resistor is on-board and placed in series with the LED to limit current. A 10kΩ resistor pulls the output pin high, but when the light from the LED is reflected back onto the phototransistor the output will begin to go lower. The more IR light sensed by the phototransistor, the lower the output voltage of the breakout board.'

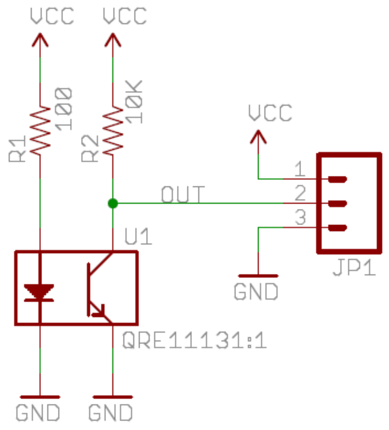


Figure 11: QRE1113 Schematic

2.8 ● Wemos Shields

One of the reasons why I have chosen the Wemos PCB is because of the number of stackable shields. There is one for almost every situation.

The full list can be seen here:
https://wiki.wemos.cc/products:d1_mini_shields

Once again I don't benefit from mentioning this here.

I have personal experience with the "Relay shield" and the "DC power shield", which will be used later in some of the constructions described.

Please note that there will also be many compatible shields made by 3rd parties which can be used.