Getting started with Sensors

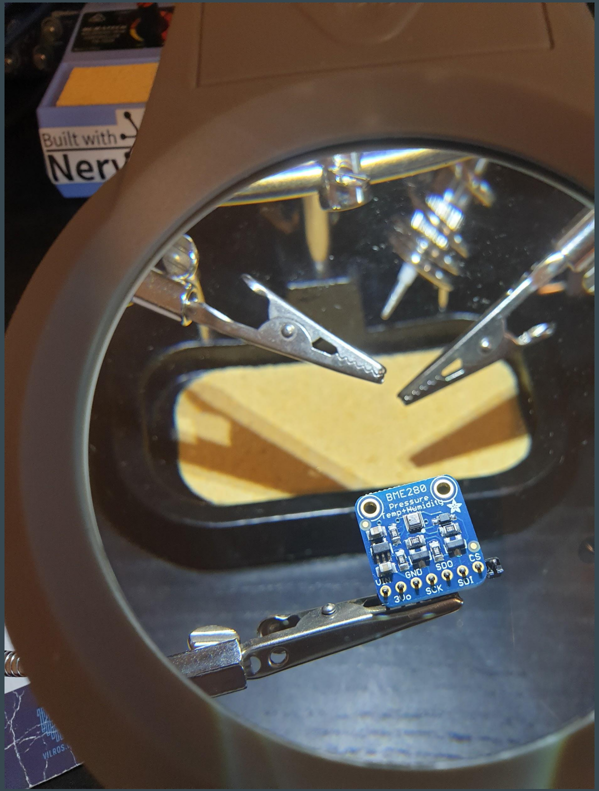
•••

FIONA MCCAWLEY

GITHUB.COM/FIMAC

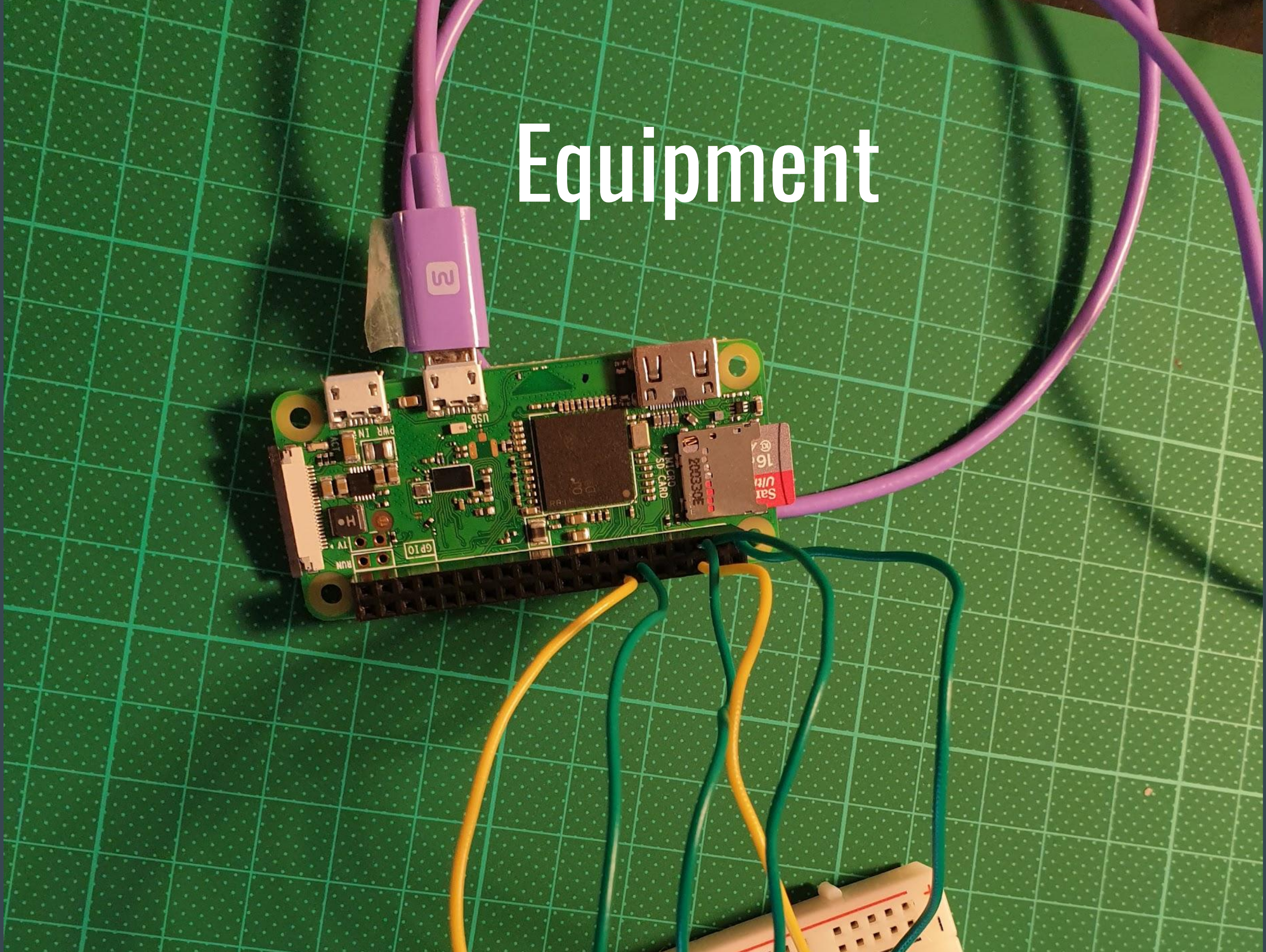
@SAUCERLIKE

Bosch BME280 - Temperature, Pressure and Humidity sensor

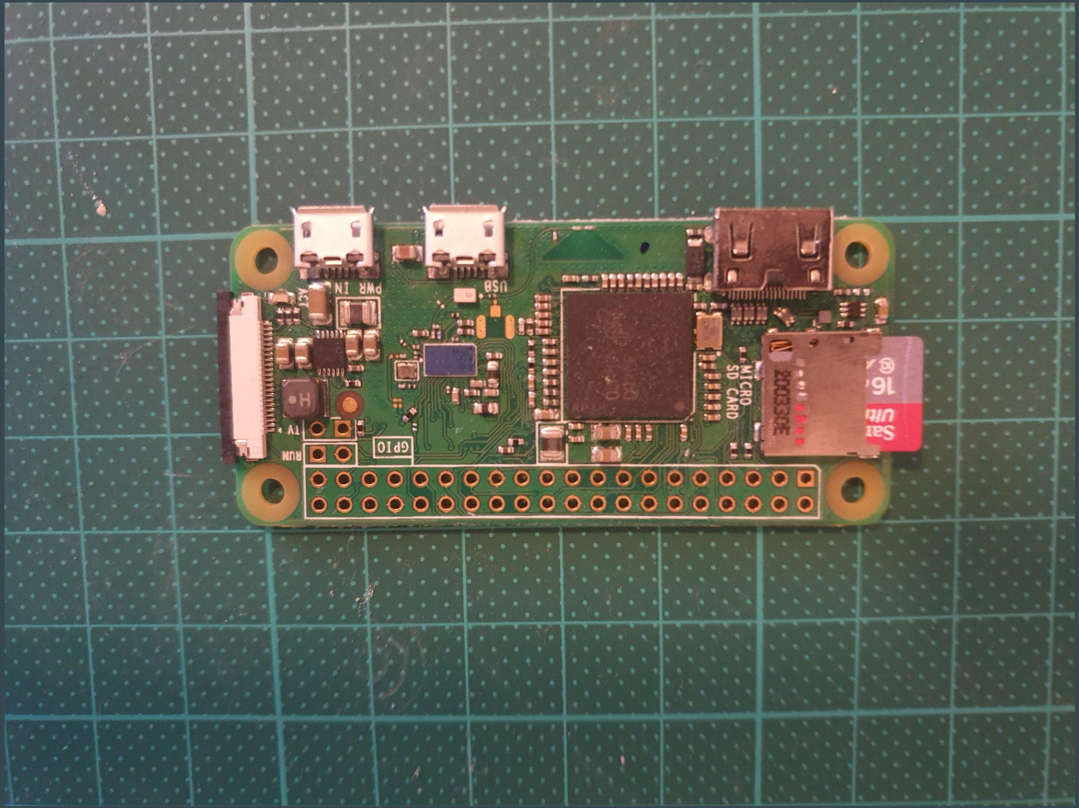


- Equipment
- Setup
- Wiring up the sensor
- Reading from the sensor

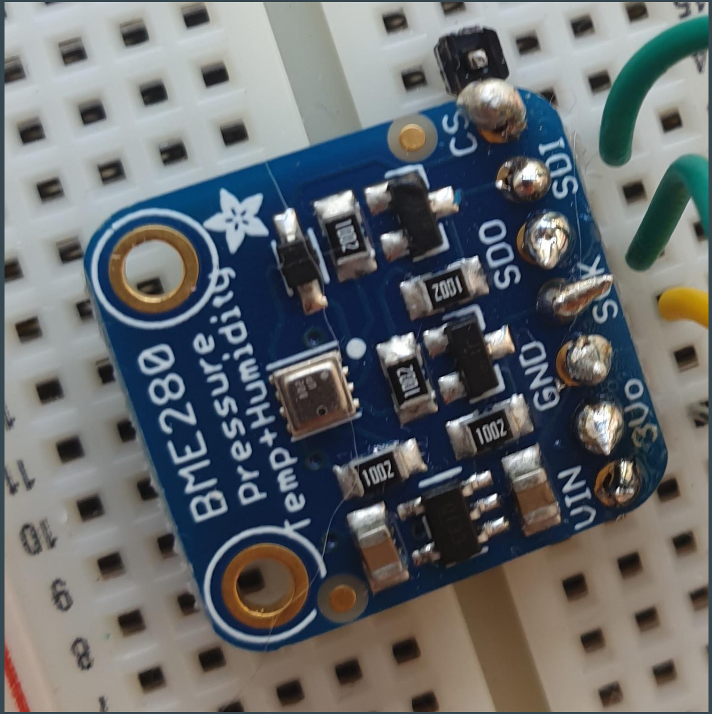
Equipment



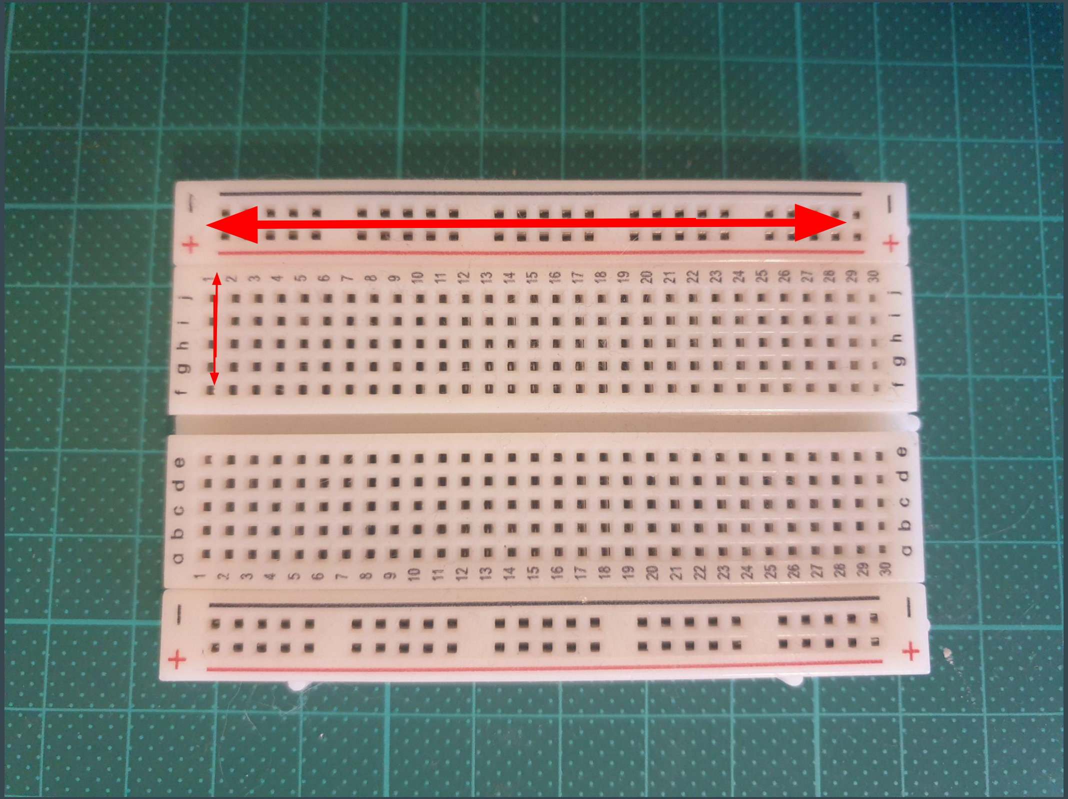
Raspberry Pi Zero W



Bosch BME280 sensor

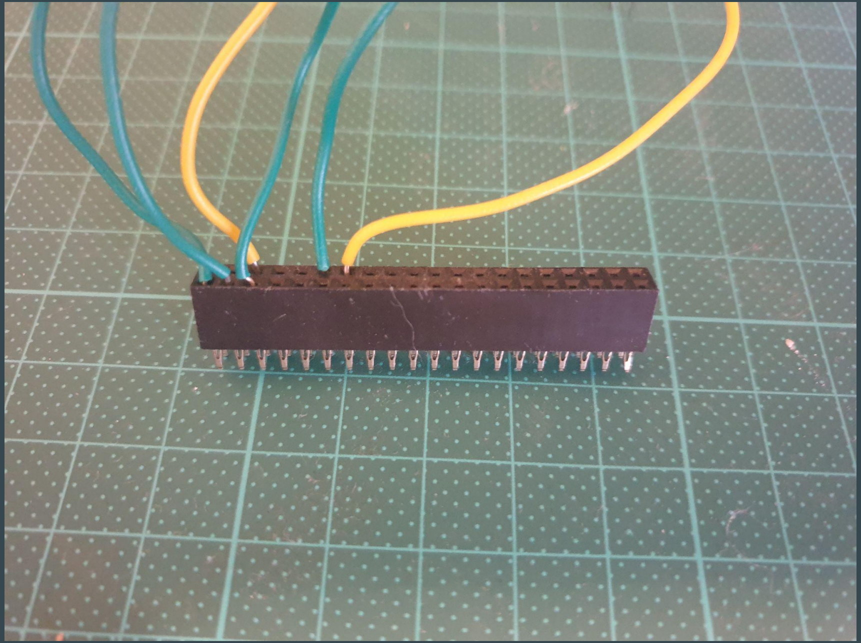


Breadboard



−
+
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
a
b
c
d
e
f
g
h
i
j

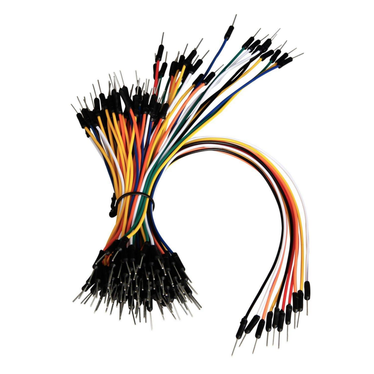
Hammer Header



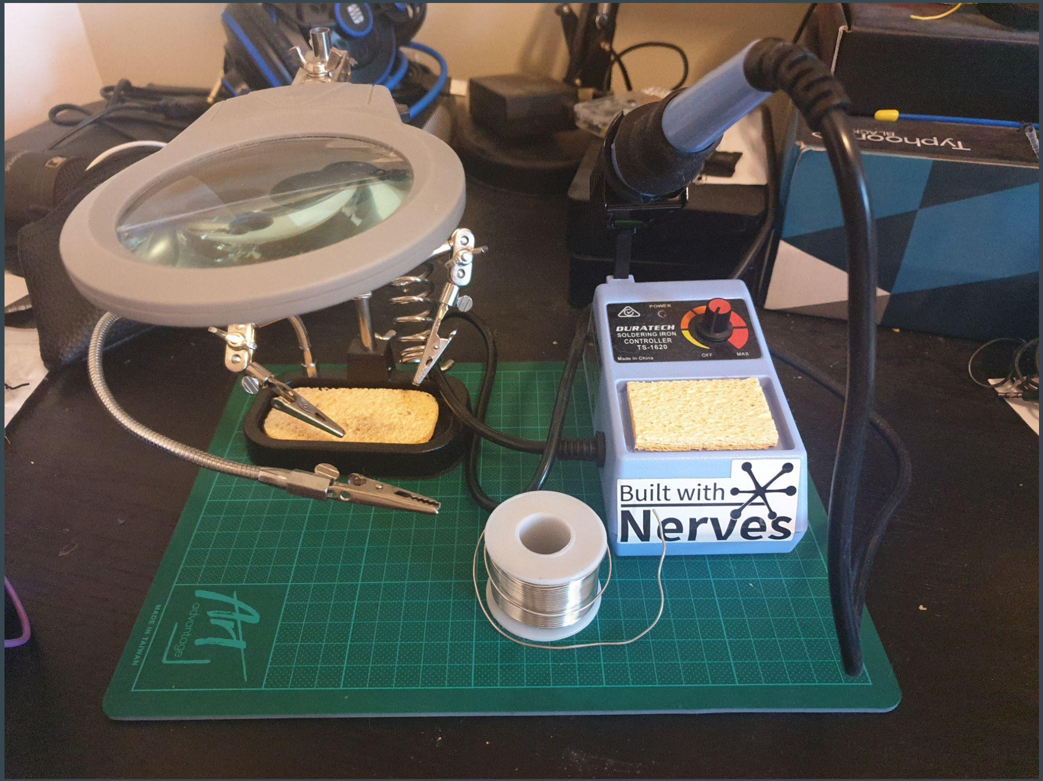
MicroSD card + Reader

Micro USB Cable

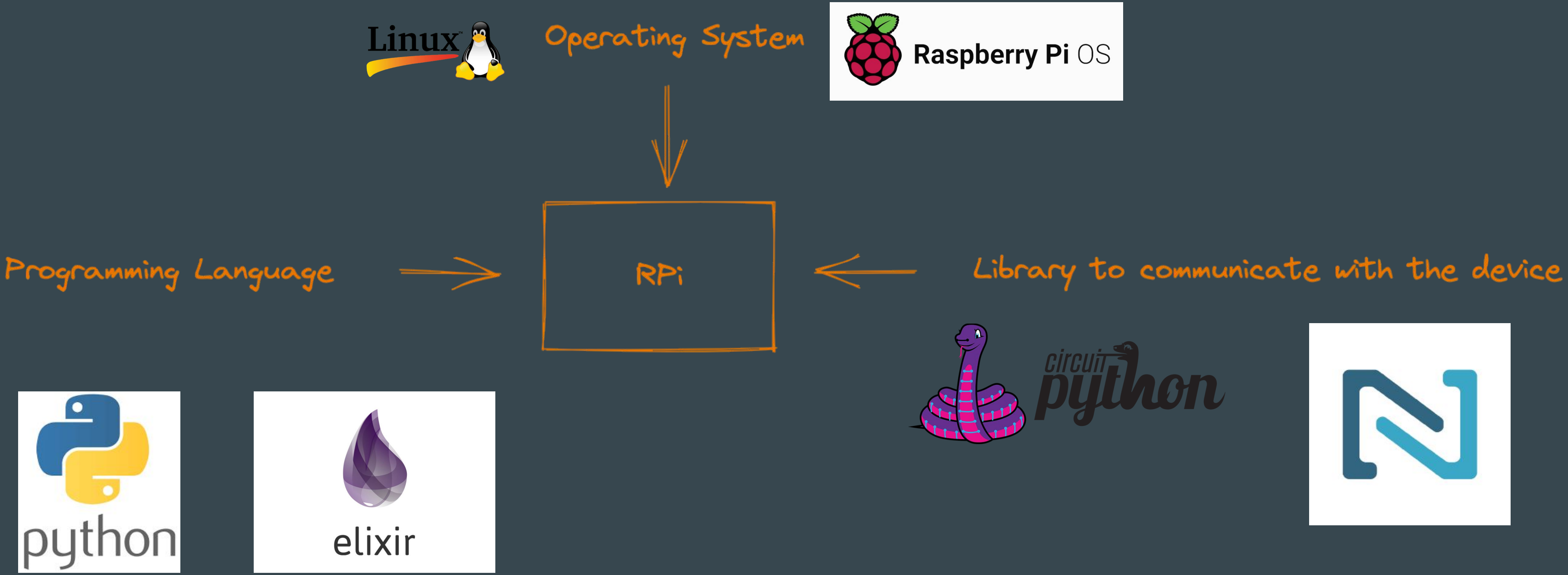
Jumper cables



Soldering setup



Setting up the Raspberry Pi



Operating System

RPi

Programming Language

Library to communicate with the device

Elixir & Nerves

- Builds a lightweight linux system specifically for a device eg Raspberry Pi 3 or Zero.

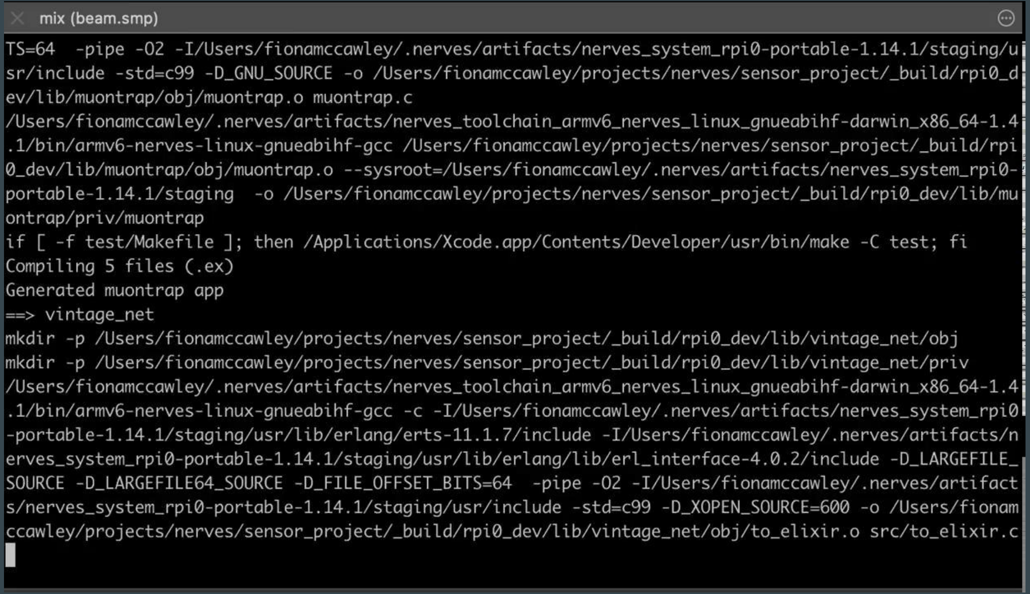
- Provides a bunch of Elixir modules, including networking and logging.

- Provides tooling enabling you to do things like update firmware wirelessly, ssh into a device.

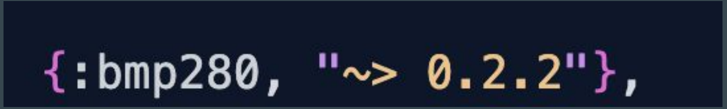
Install Nerves

https://hexdocs.pm/nerves/installation.html

Create new Nerves project and add BMP280 library as a dep

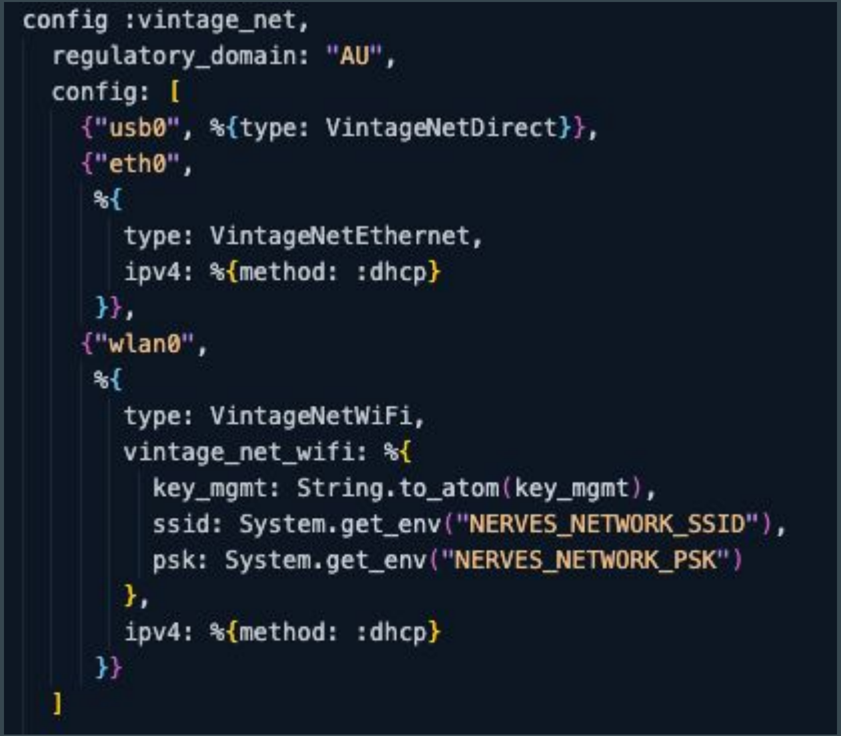


```
TS=64  -pipe -O2 -I/Users/fionamccawley/.nerves/artifacts/nerves_system_rpi0-portable-1.14.1/staging/u
sr/include -std=c99 -D_GNU_SOURCE -o /Users/fionamccawley/projects/nerves/sensor_project/_build/rpi0_d
ev/lib/muontrap/obj/muontrap.o muontrap.c
/Users/fionamccawley/.nerves/artifacts/nerves_toolchain_armv6_nerves_linux_gnueabihf-darwin_x86_64-1.4
.1/bin/armv6-nerves-linux-gnueabihf-gcc /Users/fionamccawley/projects/nerves/sensor_project/_build/rpi
0_dev/lib/muontrap/obj/muontrap.o --sysroot=/Users/fionamccawley/.nerves/artifacts/nerves_system_rpi0-
portable-1.14.1/staging  -o /Users/fionamccawley/projects/nerves/sensor_project/_build/rpi0_dev/lib/mu
ontrap/priv/muontrap
if [ -f test/Makefile ]; then /Applications/Xcode.app/Contents/Developer/usr/bin/make -C test; fi
Compiling 5 files (.ex)
Generated muontrap app
==> vintage_net
mkdir -p /Users/fionamccawley/projects/nerves/sensor_project/_build/rpi0_dev/lib/vintage_net/obj
mkdir -p /Users/fionamccawley/projects/nerves/sensor_project/_build/rpi0_dev/lib/vintage_net/priv
/Users/fionamccawley/.nerves/artifacts/nerves_toolchain_armv6_nerves_linux_gnueabihf-darwin_x86_64-1.4
.1/bin/armv6-nerves-linux-gnueabihf-gcc -c -I/Users/fionamccawley/.nerves/artifacts/nerves_system_rpi0
-portable-1.14.1/staging/usr/lib/erlang/erts-11.1.7/include -I/Users/fionamccawley/.nerves/artifacts/n
erves_system_rpi0-portable-1.14.1/staging/usr/lib/erlang/lib/erl_interface-4.0.2/include -D_LARGEFILE_
SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64  -pipe -O2 -I/Users/fionamccawley/.nerves/artifact
s/nerves_system_rpi0-portable-1.14.1/staging/usr/include -std=c99 -D_XOPEN_SOURCE=600 -o /Users/fionam
ccawley/projects/nerves/sensor_project/_build/rpi0_dev/lib/vintage_net/obj/to_elixir.o src/to_elixir.c
```

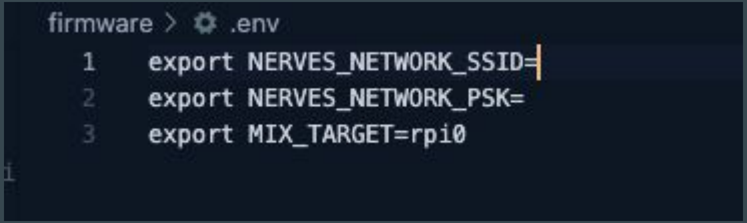


```
{:bmp280, "~> 0.2.2"},
```

Update wifi config

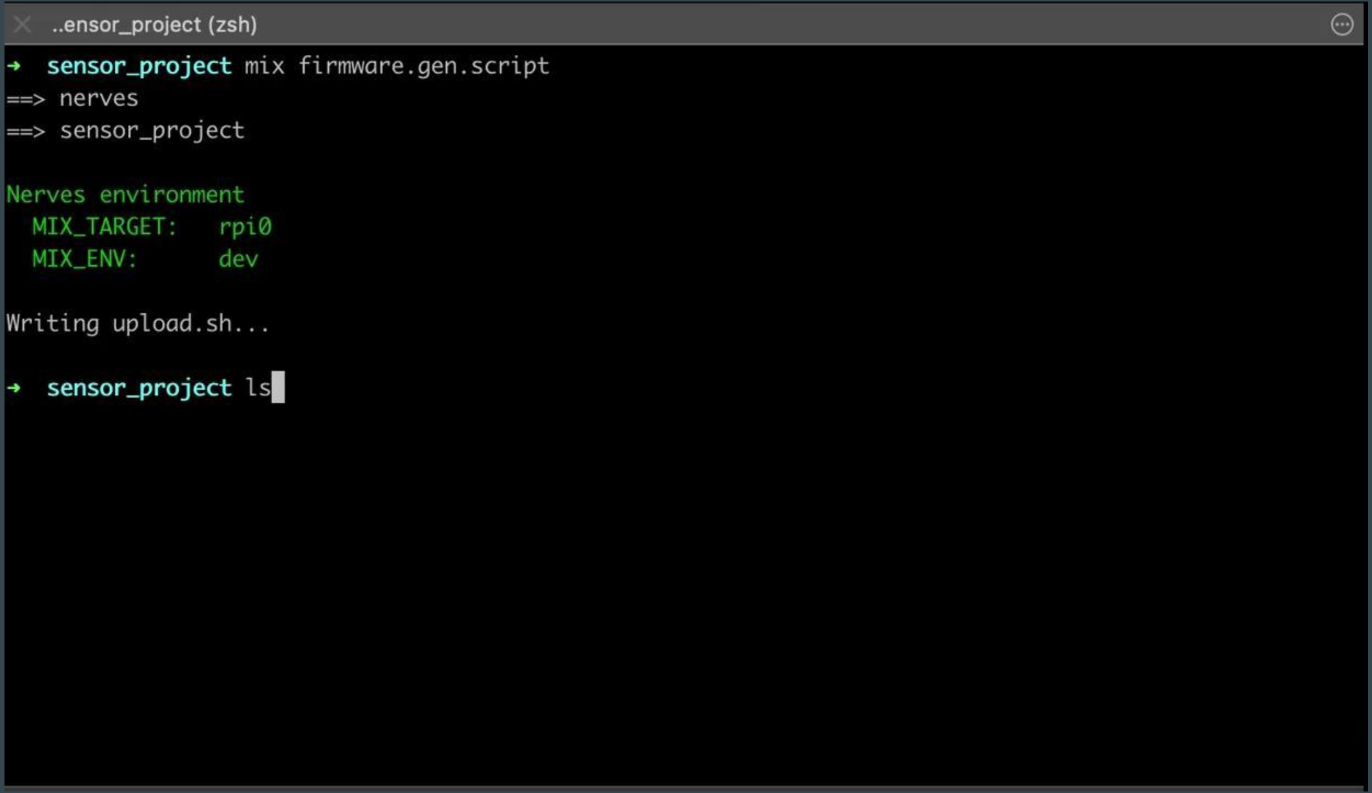


```
config :vintage_net,
  regulatory_domain: "AU",
  config: [
    {"usb0", %{type: VintageNetDirect}},
    {"eth0",
     %{
       type: VintageNetEthernet,
       ipv4: %{method: :dhcp}
     }},
    {"wlan0",
     %{
       type: VintageNetWiFi,
       vintage_net_wifi: %{
         key_mgmt: String.to_atom(key_mgmt),
         ssid: System.get_env("NERVES_NETWORK_SSID"),
         psk: System.get_env("NERVES_NETWORK_PSK")
       },
       ipv4: %{method: :dhcp}
     }}
  ]
```



```
export NERVES_NETWORK_SSID=
export NERVES_NETWORK_PSK=
export MIX_TARGET=rpi0
```

Setup network and usb cable firmware updates



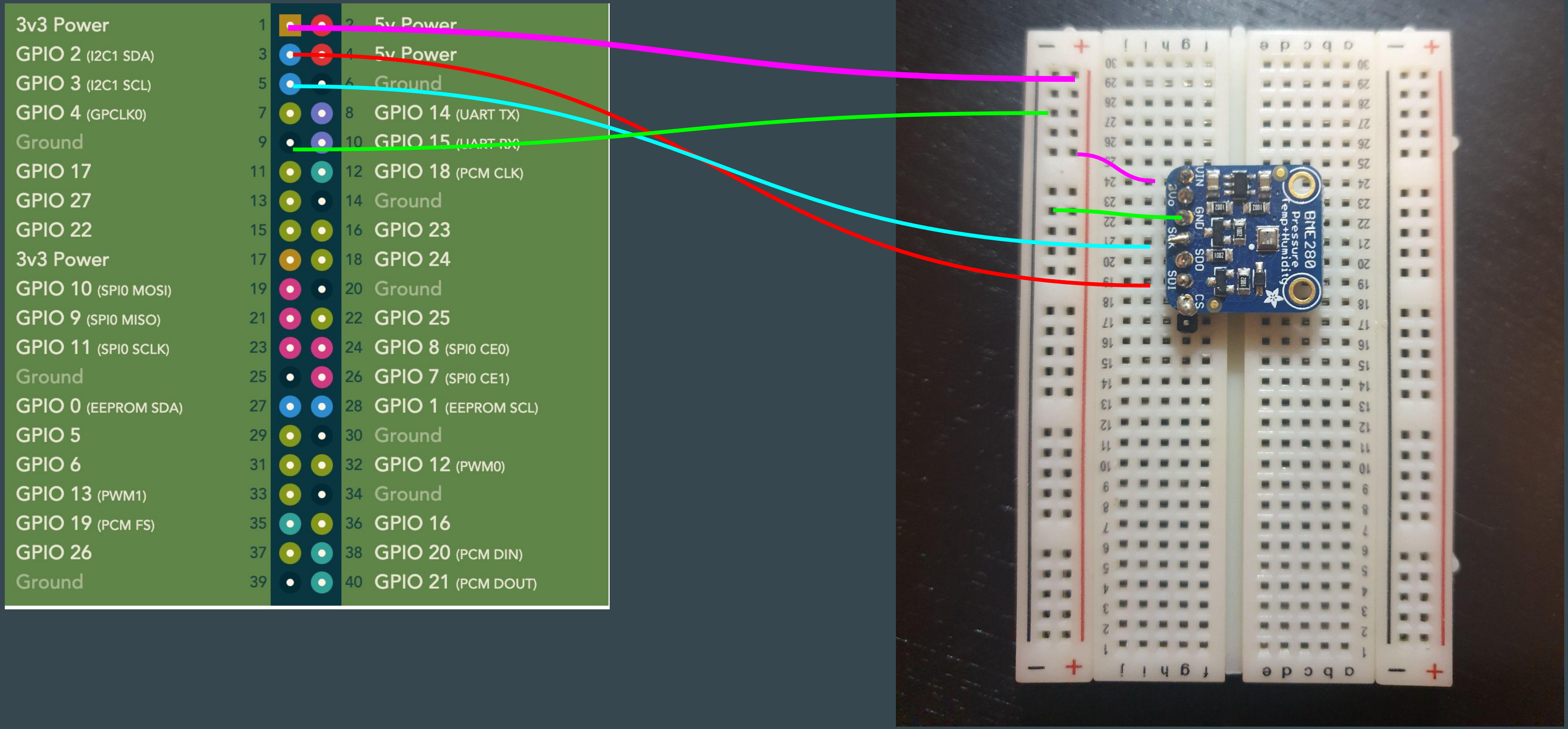
```
➜  sensor_project mix firmware.gen.script
==> nerves
==> sensor_project

Nerves environment
  MIX_TARGET:   rpi0
  MIX_ENV:      dev

Writing upload.sh...

➜  sensor_project ls
```

Burn to SD card

SSH into device, logging and other tools.

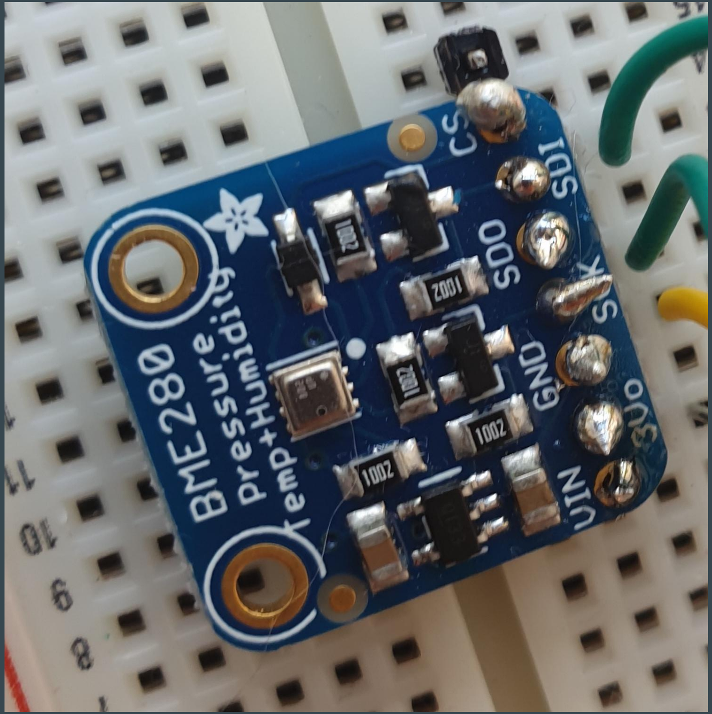
Updating firmware over USB cable or WiFi

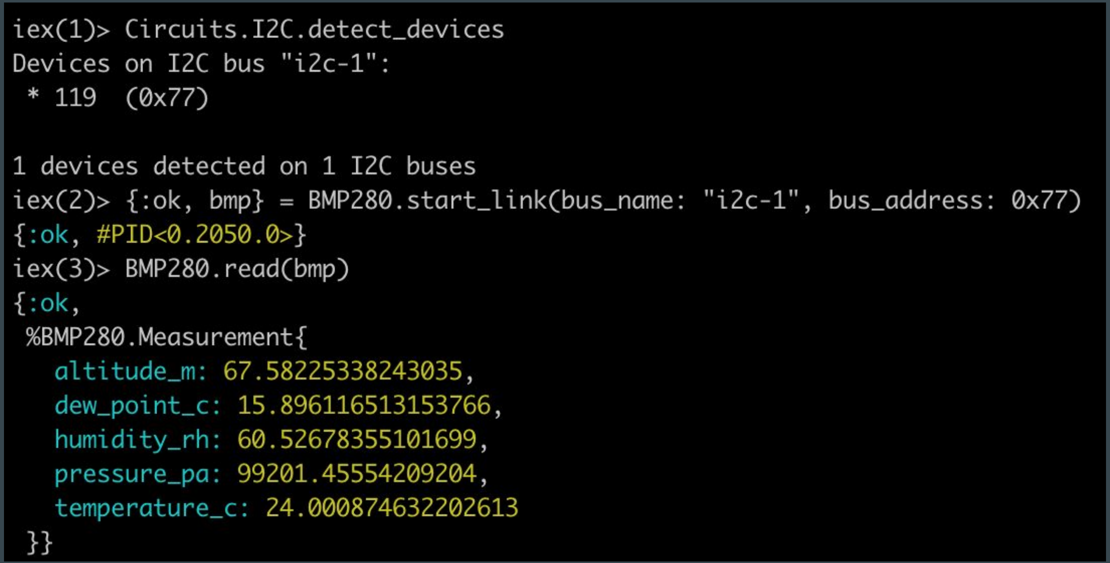
Wiring



3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)

Start reading temperatures





```
iex(1)> Circuits.I2C.detect_devices
Devices on I2C bus "i2c-1":
 * 119  (0x77)

1 devices detected on 1 I2C buses
iex(2)> {:ok, bmp} = BMP280.start_link(bus_name: "i2c-1", bus_address: 0x77)
{:ok, #PID<0.2050.0>}
iex(3)> BMP280.read(bmp)
{:ok,
 %BMP280.Measurement{
   altitude_m: 67.58225338243035,
   dew_point_c: 15.896116513153766,
   humidity_rh: 60.52678355101699,
   pressure_pa: 99201.45554209204,
   temperature_c: 24.000874632202613
 }}
```

Getting started with sensors …

Slides: https://github.com/fimac/talks
Repo: https://github.com/fimac/temp_sensor_nerves

GITHUB.COM/FIMAC

@SAUCERLIKE