ARTICLE PHOTON2 TUTORIAL

Air Quality Monitor

FireFli PTY LTY ~ FE Basson

Introduction

The purpose of this project is to build a simple air quality sensor that will monitor the eCO2 and TVOC levels or it's surrounding area. The project is based around the Particle Photon2 development board and the Sensirion SGP30 Volatile Organic Compounds sensor.

This gas sensor, which can identify a variety of volatile organic compounds (VOCs) and H2, is used to monitor the quality of the air within buildings. It will transmit a Total Volatile Organic Compound (TVOC) reading and an equivalent carbon dioxide reading (eCO2) over I2C. This component will measure the concentration of eCO2 (equivalent computed carbon dioxide) between 400 and 60,000 parts per million (ppm) and the concentration of TVOC (Total Volatile Organic Compounds) between 0 and 60,000 parts per billion (ppb).

We will then add a WAVE 2" TFT display and mount the breadboard, sensor and display inside a 3D printed enclosure for a more complete prototype.

Prerequisites

You should have a basic understanding of Particle WebIDE or Visual Studio Code and how to import libraries.

You should have a Particle Photon2 and a Particle account. If you do not have an account, visit https://www.particle.io and register and account.

You should have at least one Particle Photon2 claimed and active on your account.

Bill of materials

Items you will need to complete this turorial

1 x Particle Photon2
1 x Breadboard
1 x USB cable
1 x ADAFRUIT SGP30 development board (or similar)
1 x 3.7V LiPo battery (or suitable power supply)
1 x Jumper Wires (male2-male and male-2-female)
1 x 2” WAVE TFT Display
4 x 2 x 6mm Machine screws

4 x 2 x 3.4mm Heat Inserts (optional ~ only for enclosure)
1 x 3D printed enclosure (optional ~ STL’s provided)
1 x 40mm 5V fan ~ (optional)
1 x n-Channel MOSFET ~ (optional)

Software used

Visual Studio Code

Cura Slicer

Autodesk Fusion360

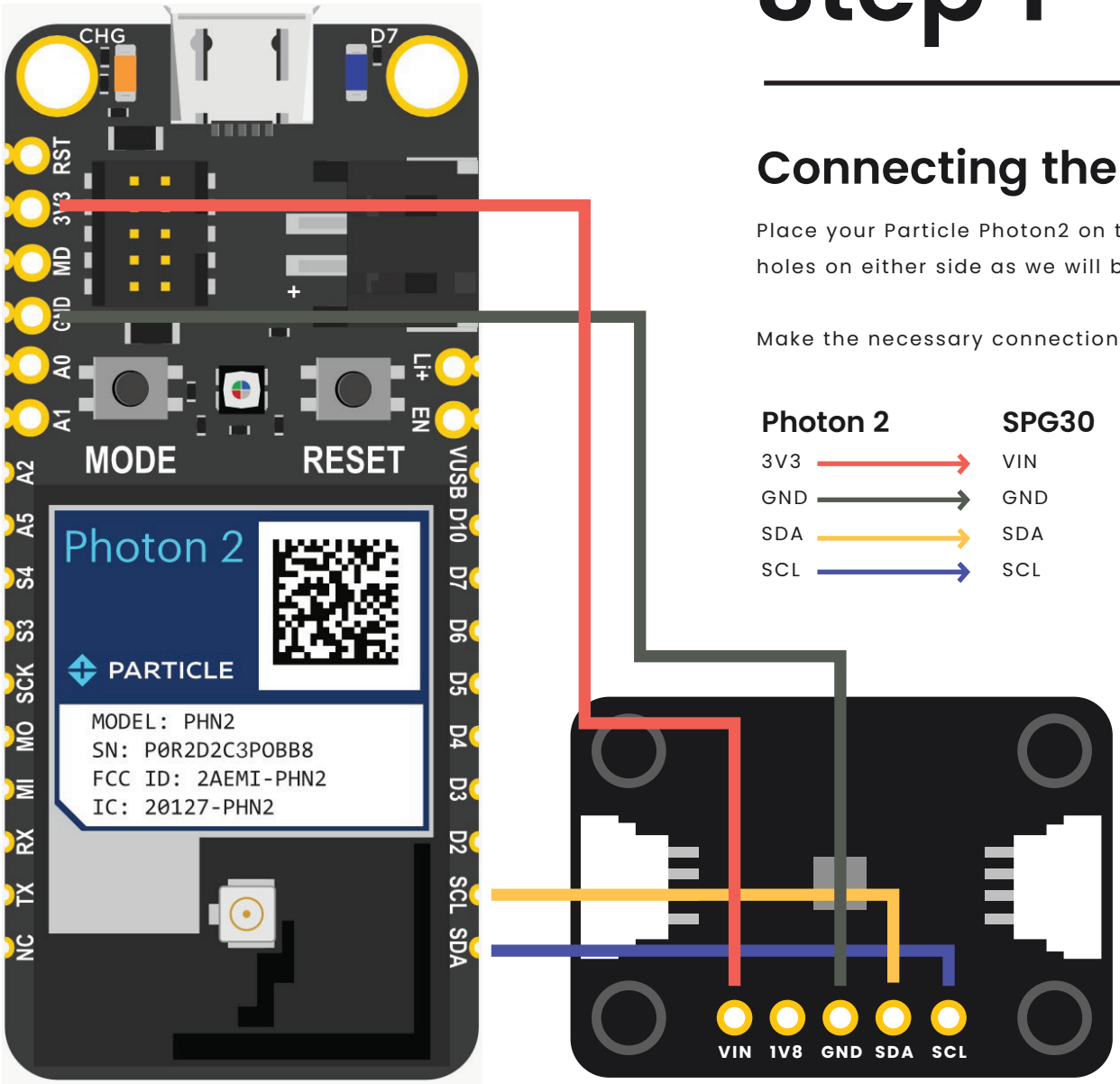
Step 1

Connecting the SGP30 Sensor

Place your Particle Photon2 on the breadboard and ensure there are exposed holes on either side as we will be connecting wired on both sides of the Photon2.

Make the necessary connections between die Photon2 and the SPG30 module

Photon 2	SPG30
3V3	VIN
GND	GND
SDA	SDA
SCL	SCL



Step 2

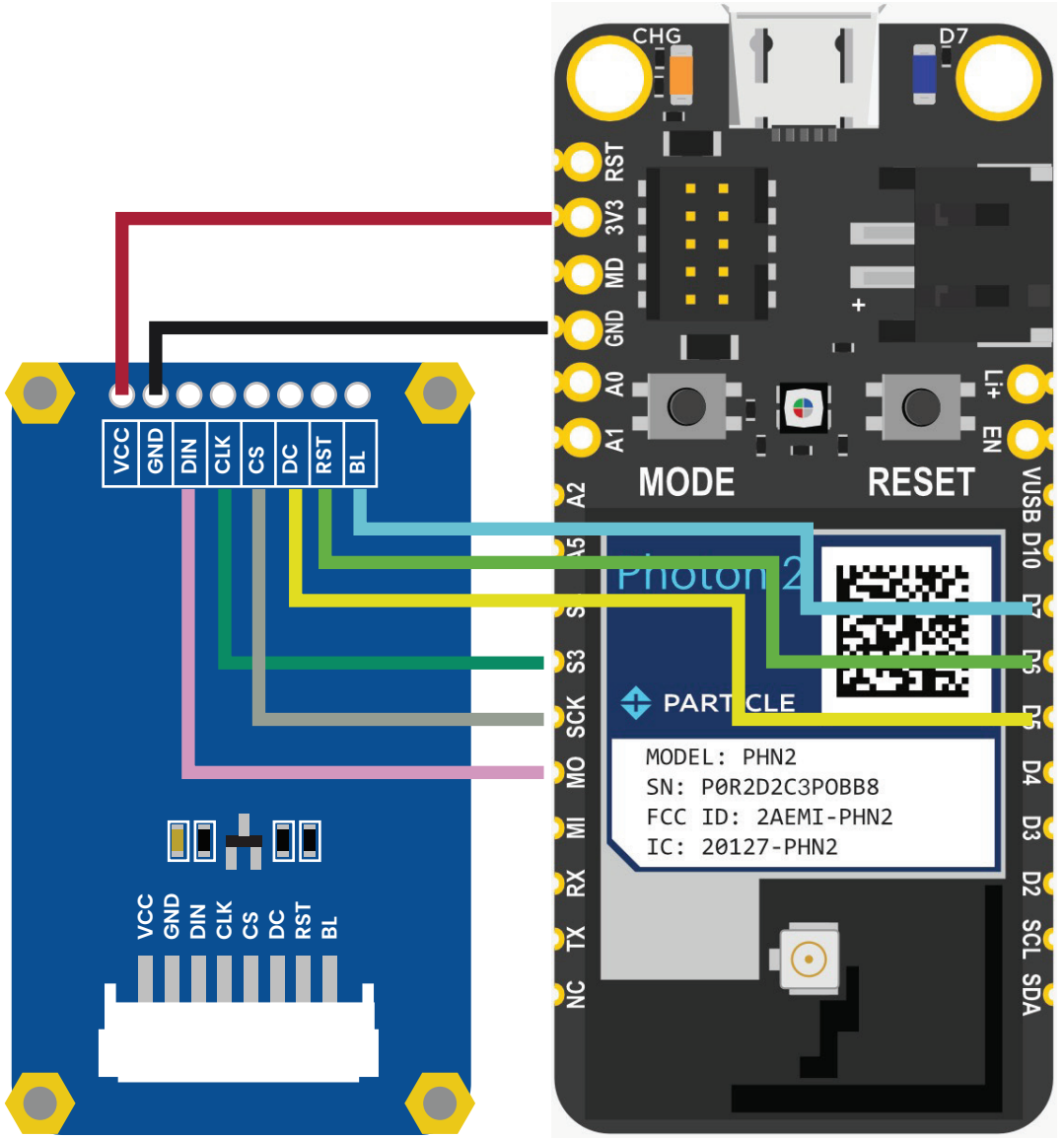
Connecting the TFT Dispay

This part can be a little more involved so pay careful attention to the wiring.

Photon 2	WAVE 2”
3V3	VCC
GND	GND
DIN	MO
CLK	SCK
CS	S3
DC	D5
RST	D6
BL	D7

NOTE

You can use either the connector and wire harness that came with the screen, or solder headers onto board depending on you connection preference



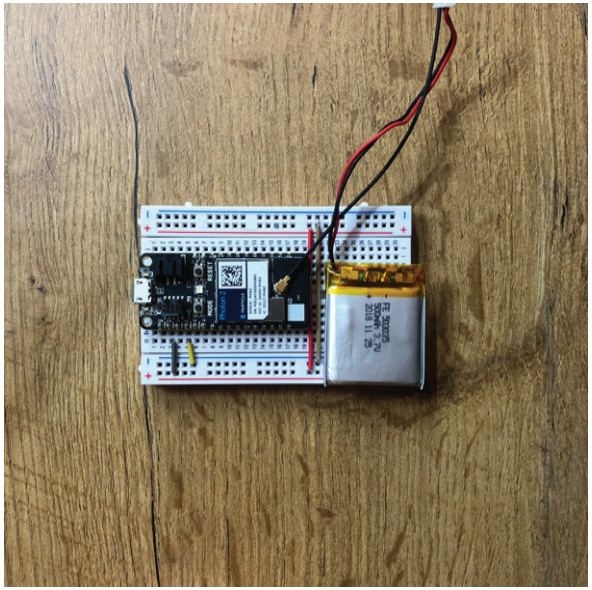
Step 3

Recap

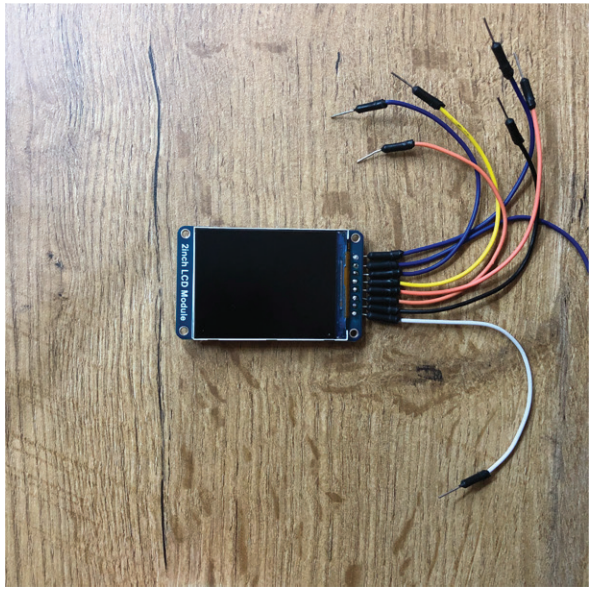
Referring to the images below, you should have something similar

NOTE

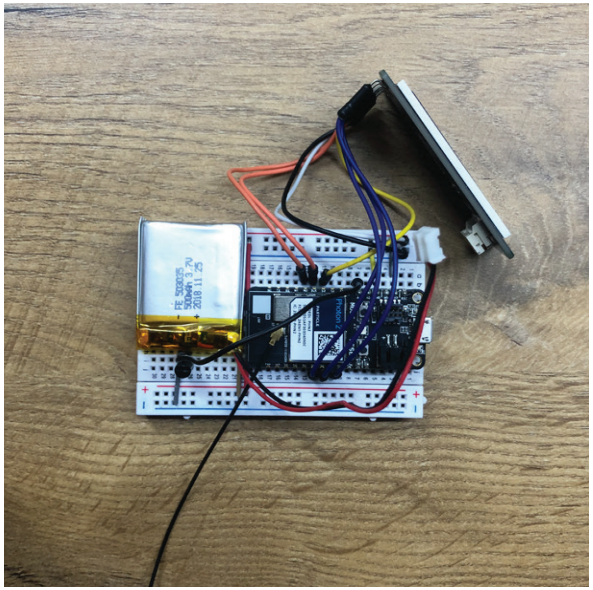
Depending on the jumper wires you selected, it might look a little different but as long as the pin outs match all should be good.



Step 3.1



Step 3.2



Step 3.3

Step 4.1

Installing into the enclosure

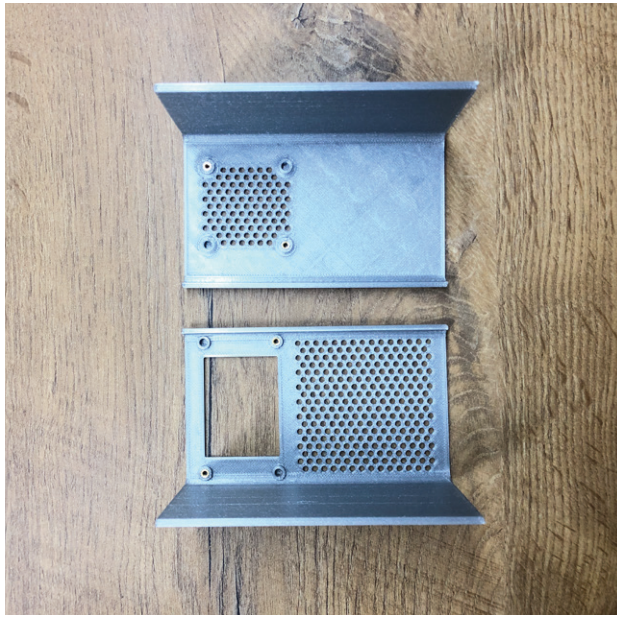
Using you soldering iron, install the heat inserts into the designated areas as per the images on the right **Fig 4.1**.

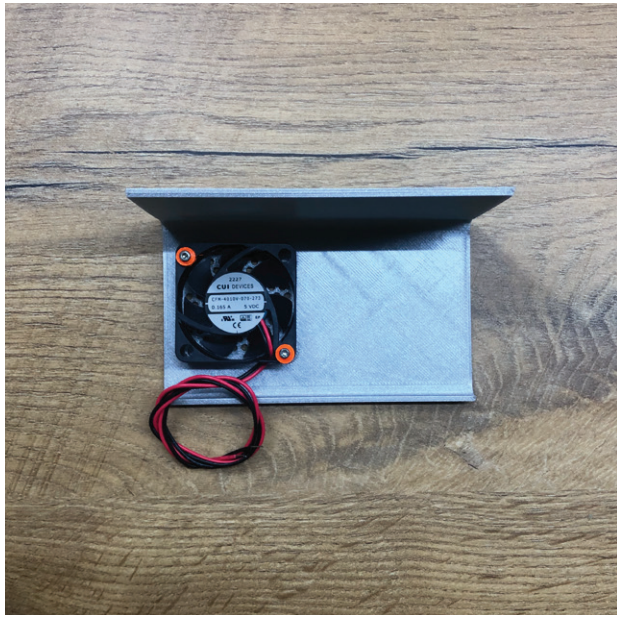
Two inserts for the screen should suffice but I would recommend all four to keep the lid in place.

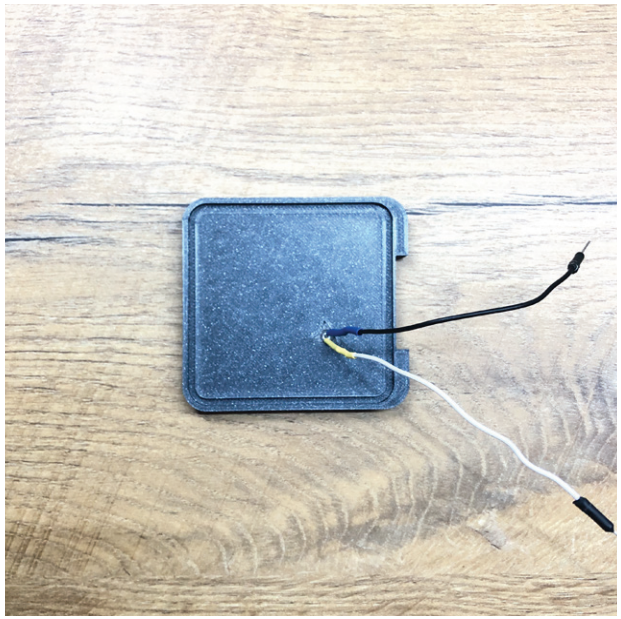
Next fix the fan in place as indicated in **Fig 4.2**.

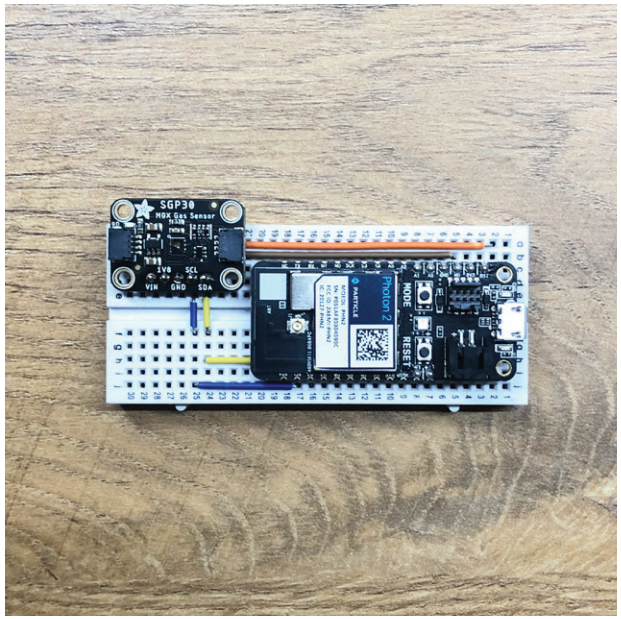
Mounty the LED in place on the side panel and solder two wires on that will connect to the breadboard later **Fig 4.3**.

Tidy up the connections as much as possible. I used solid core wires placed on the breadboard to save some space and make it look a little nicer **Fig 4.4**. Be sure to follow the wiring diagrams from earlier.









Step 4.2

Installing into the enclosure

Mount the screen in place **Fig.4.5**.

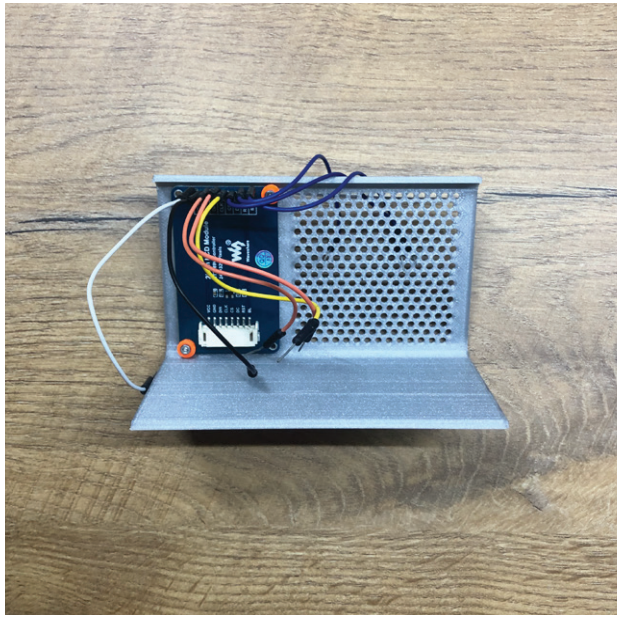
Next mount the breadboard in place. Make sure to line up the USB port on the Photon2 with the cutout in the side panel to allow USB power/charging **Fig.4.6**.

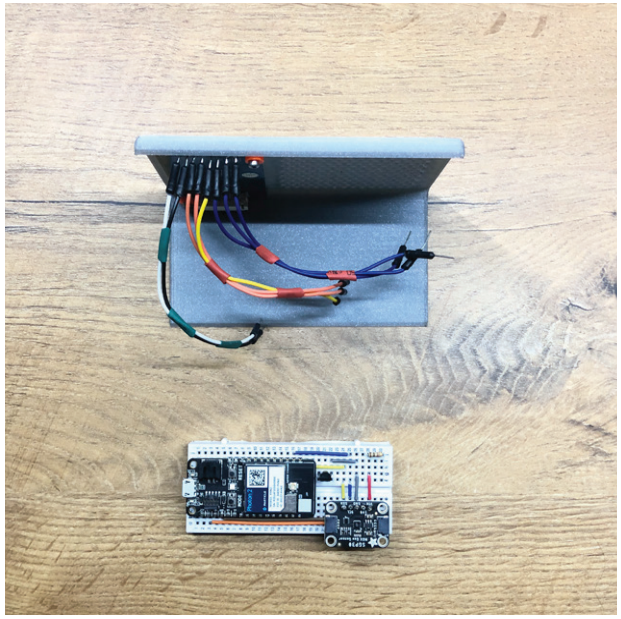
We will use a MOSFET to control the fan speed.

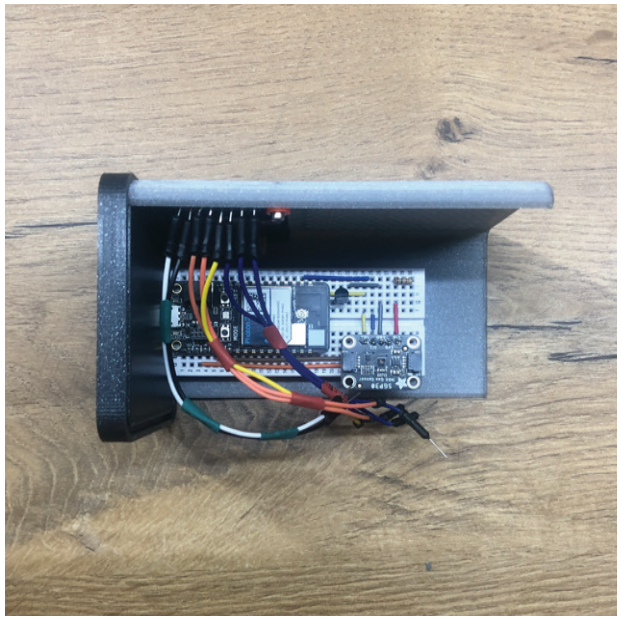
Connect the n-MOSFET and Fan as per **Fig.4.7**. Make sure of the pinout of your n-MOSFET as they can differ. You want to connect it as follow:

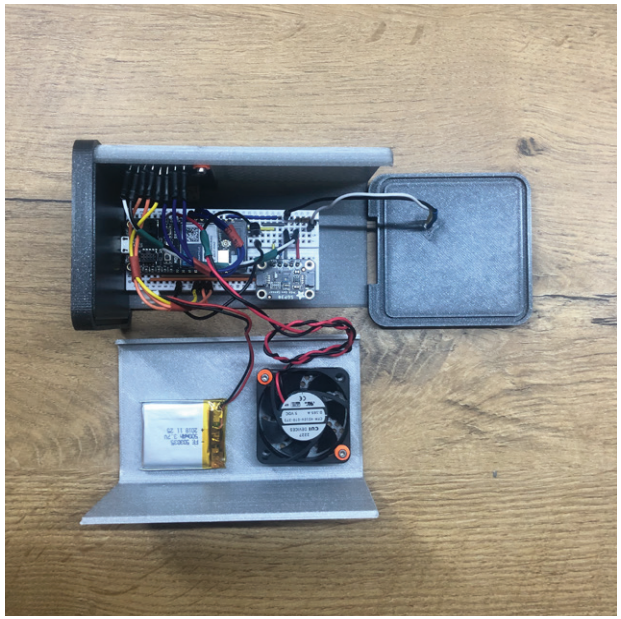
Li+	→	Fan V+
A1	→	MOSFET Gate
GND	→	MOSFET Source
MOSFET Drain	→	FAN V-

You should have something that resembles the last picture **Fig.4.8**.









Step 5

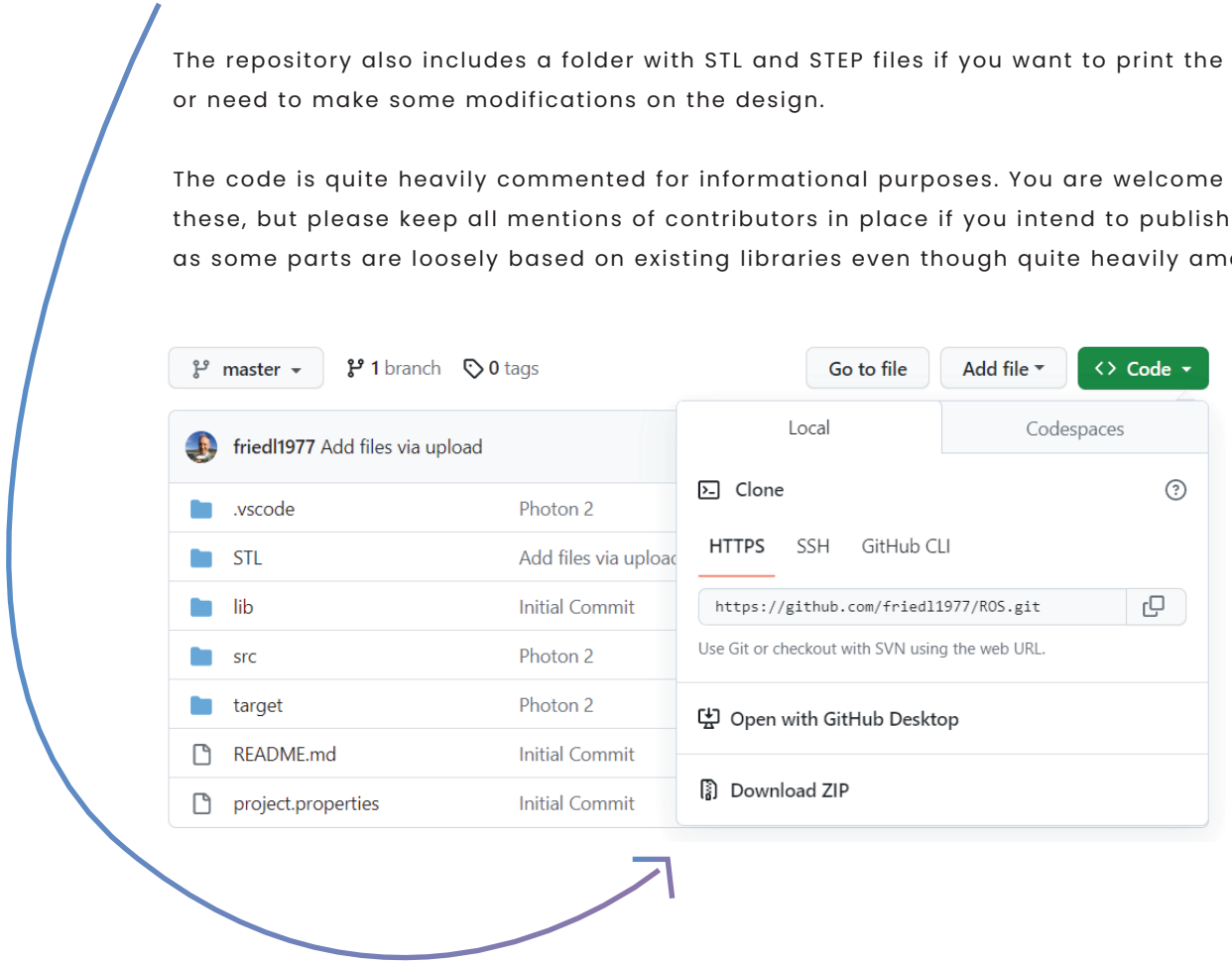
Code

Everything has been done for you. Simply follow the link below and download the entire project from the Github repository.

https://github.com/friedl1977/Air_Quality_Sensor

The repository also includes a folder with STL and STEP files if you want to print the enclosure or need to make some modifications on the design.

The code is quite heavily commented for informational purposes. You are welcome to remove these, but please keep all mentions of contributors in place if you intend to publish this code as some parts are loosely based on existing libraries even though quite heavily amended.



NOTE

If you are using Visual Studio Code, make sure to use a USB cable to flash. If you use cloud flash, the libraries hosted on the sever will be used and your display will not function as intended.

The library in this project has been amended to accommodate the 2" display

Step 6

Working principal

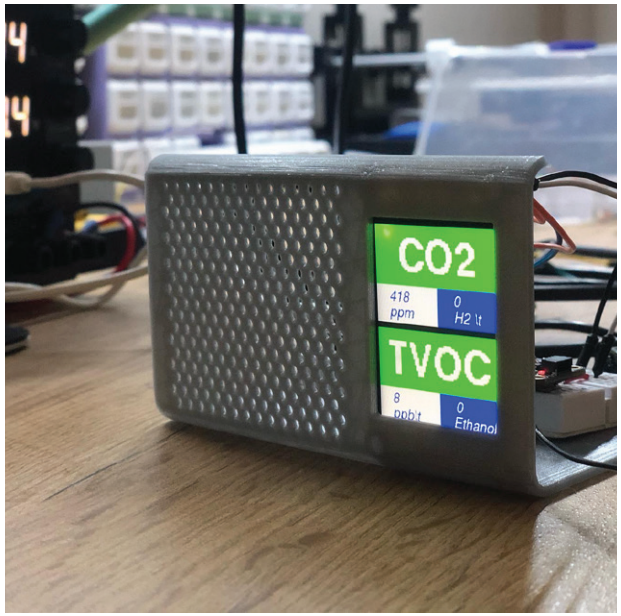
The monitor can be placed in the area you want to monitor the air quality of.

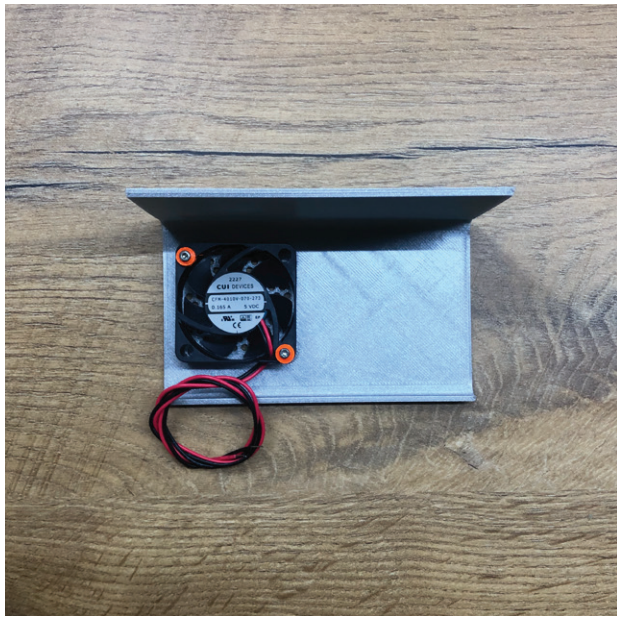
The screen will flash when starting up, this is normal due to the screen being refreshed.

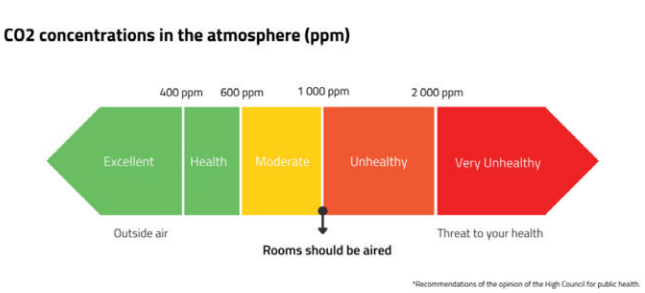
The sensor displays the CO2 and TVOC status by means of coloured block which will change as thresholds are exceeded.

The predefined limits follow the suggested limits indicated in **Fig 6.2** and **Fig 6.3**.

Due to the power consumption of the SPG30 sensor, I recommend using a battery with a higher capacity rating or leave the device powered by USB







Index Category	Index Value	TVOC (ppb)
Good	0 - 50	0 - 220
Moderate	51 - 100	221 - 660
High	101 - 150	661 - 1430
Very High	151 - 200	1431 - 2200
Very High	201 - 300	2201 - 3300
Very High	301 - 500	3301 - 5500

Room for imporovement

this is NOT a production ready design :)

Add a power switch (simple ON/OFF) in case you want to use an internal battery

Refine the code - A LOT. There is definitely room for improvement on the algorithm that determines the direction of the person travelling.

The VL53L1X sensor has it's limitations in terms of reading speed. Increasing the scan speed to much will reduce the accuracy and cause false readings.

Two sensors would be more accurate in my opinion, but installation would be more challenging as you would need to install on sensor each side of the door. The there is no physical door, this would be easy enough.

If you intend to use this product , DESIGN A PCB instead of using a breadboard. While breadboards are acceptable means for rapid prototyping, the present many challenges.

Enjoy the project!!