DIY weather station build

Introduction

I first thought about using a raspberry pi for a weather station back in September 2012, my initial plan was that I would make all of the sensors myself. It turned out that the project was simply too big for the spare time I had available so it never got started. In March 2016 I saw the blog post from Clive titled "Big Brown Boxes", all about the Raspberry Pi weather station kit shipped to schools. A little bit of researched revealed that the available information was sufficient for me to build my own while making use of the software provided.

The goals

- Compatible with the Raspberry Pi weather station software.
- Easy for others to build using readily available parts.
- Minimal cost.

Raspberry Pi weather station kit

The first step is to identify what the official kit uses and how it connects to the pi. Although there are no schematics available most of the required information can be found on the weather station website. A summary of which can be found in the table below:

Sensor/IC Function	Manufacturer & Part No.	Quantity	Comments
Anemometer	Maplin N09QR or N82NF	1	Connects between GPIO5 and 0V
Wind vane	Maplin N81NF	1	Connects to ADC input (MCP3427 I2C address 0x69)
Rain gauge	Maplin N77NF	1	Connects between GPIO6 and 0V
Ground temperature	Maxim DS18B20	1	Connects to GPIO4 (One wire interface)
Real Time Clock (RTC)	NXP PCF8253T	1	Connects to I2C address 0x68 (SDA1 and SCL1)
Analog to Digital Converter (ADC)	Microchip MCP3427	2	Connect to I2C address 0x69 and 0x6A
Humidity and air temperature	Measurement Specialist HTU21D	1	Connects to I2C address 0x40
Barometric pressure	Bosch BMP180	1	Connects to I2C address 0x77
Air quality	Figaro TGS2600	1	Connects to ADC input (MCP3427 on address 0x6A)

Design Simplification

The PC8253T RTC seems to be superfluous to requirement, especially as the Pi is connected to a wired network, the Pi can simply use ntp to obtain the time just like all other Raspberry Pi do. Because of this and the fact that the part chosen is not the easiest to obtain the rtc will not be included in this design.

The air quality sensor is said to be un-calibrated, requires a separate ADC chip, produces about 210mW of heat and hold little interest to me. Removing this from the design saves around £6 and simplifies the design.

Providing Power

The official Raspberry Pi weather station incorporates a 24V to 5V dc-dc converter to power the Pi through the GPIO connector, the intention is that you use what is termed passive Power over Ethernet (PoE) to provide the power. Passive PoE can be used because the Pi only supports 100mb ethernet which in turn only uses two of the four ethernet cable pairs (passive PoE makes use of the two unused ethernet cable pairs to provide power). This will not work with commercial active PoE devices.

I wanted to use a simpler neater solution that would give the possibility of using active PoE with commercial PoE devices. To accomplish this I chose to make use of the 802.3af standards compliant TP-Link TL-POE10R splitter, available for around £10.

This is all well and good if you already have active PoE or a planning to include it, if not then this turns out to be a rather expensive option. Fortunately TP-Link offer a lower cost solution known as the TP-Link TL-POE200, this kit comes with everything necessary to provide PoE to the weather station and only costs around £13. The downside of this solution is that the specification makes no mention of being 802.3af compliant so you are unlikely to be able to use this with commercial PoE devices should you choose to at a later date.

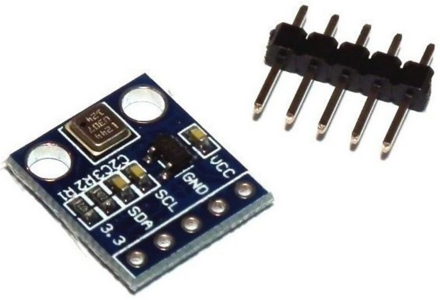
Choosing and buying the parts

So we know that we need the an anemometer, wind vane, rain gauge, ground temperature sensor, humidity sensor, air temperature sensor and barometric pressure sensor so where and how do we get them?

The anemometer, wind vane and rain gauge can be obtained from Maplin in the UK, the Maplin part numbers are N82NF or N09QR, N81NF and N77NF respectively.

The ground temperature sensor can be obtained from Adafruit (product ID 381), ebay or amazon (search for DS18B20 and look for waterproof in stainless housing with cable of desired length). Note: they usually come with wire colours of yellow, black and red, the red being +V, black 0V and the yellow the signal.

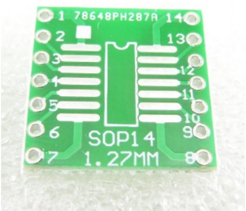
The barometric pressure sensor is a surface mount device and not suitable for DIY, fortunately it can be obtained from ebay or amazon (search for BMP180) already mounted on a small board, ideally the connection layout should be the same as the one shown on the picture below with the pin header pre-soldered on the component side or loose so that you can solder this on the correct side yourself:



You can use similar items from Adafruit or Sparkfun but you may need to modify the strip board to suit the connection layout.

Humidity/air temperature is provided by a HTU21D, again this is only available in surface mount, it is available already mounted on a small board from Adafruit, Sparkfun, ebay or Amazon. This connects directly to a flying lead so in this case the connection layout is not important. Searching for HTU21D on Ebay or Amazon should reveal the right part, in this design we do not need 5V to 3V3 conversion/level shifting so the boards that appear to have fewer components should be fine here.

The wind vane outputs an analogue voltage that is proportional to wind direction, the Raspberry Pi does not include any analogue inputs as standard. To overcome this the official weather station uses an analogue to digital convert chip known as a MCP3427. This chip is only available in surface mount and while someone experienced in soldering surface mount components may be able to solder one of these it is not compatible with strip board. Because we want to make our weather station compatible with official station software our options are limited. The solution is to buy a small adapter board to convert the surface mount chip to the 0.1 inch pitch require by the strip board and to use the larger pitch MCP3428 14 pin SOIC package to make soldering easier. Here’s a picture of the one I bought from ebay with and without the MCP3428 mounted:



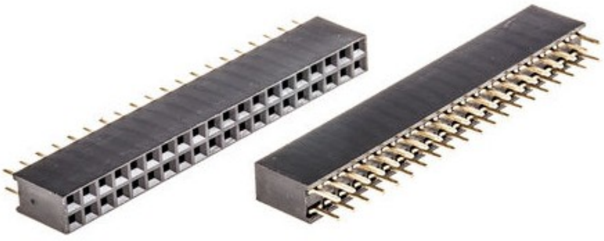
To mount the adapter board to the strip board you will need two seven pin 2.54mm pitch single row headers, these are easy to trim to the correct length so you just need to make sure you buy one long enough to make two seven pin lengths:

This design uses 2 way 0.1 inch (2.54mm) pitch screw terminal connectors to connect the external sensors. I could only find these on ebay direct from China so allow plenty of time for delivery, I bought 20 for £2.85 delivered:

We need several 1/4W through hole resistors, carbon or metal film are fine (metal film are considered better quality). Buying single resistors is an expensive way to buy resistors, it is far better to buy a kit of resistors containing the values you want, the advantage of buying a kit is that you will then have a selection of resistors that can be used for other projects. There are many resistor kits available from several sources, these will often not include the exact values used in this design. In most cases choosing the closest value your kit has will be fine, with the exception of R4 which should be close to 75K Ohms. Unfortunately this is one value that is most likely to be missing from the lower cost kits. This can be solved in a number of ways like using two resistors in series (68K + 6.8K) or two resistors in parallel, for example two 150K resistors in parallel or even 100K in parallel with 300K, you will need to choose the option which best suits what you have.

Several capacitors are used, the most important being C1, C2 and C3, again exact values are not that important and the circuit may well work even without C4-C7 being fitted (these are fitted to help prevent electrical noise from giving erroneous results). Ideally C1 should be a through hole tantalum bead capacitor with a voltage rating of at least 6V although an electrolytic capacitor should work as well. The other capacitors can be 2.54mm pitch low cost ceramic capacitors of any voltage rating greater than 3.3V.

You need to connect the strip board to Raspberry Pi GPIO pins, for this we use a 40 way socket. I used a standard 40 way PCB mount socket and bent the pins we need to connect as right angles to make soldering easier:



You will need a piece of strip board with 0.1inch hole spacing to mount the components on, it is unlikely that you will be able buy a piece of the exact size required so just make sure you buy a that is larger than 6.35cm x 6.83cm (make sure the copper traces run parallel to the short dimension).

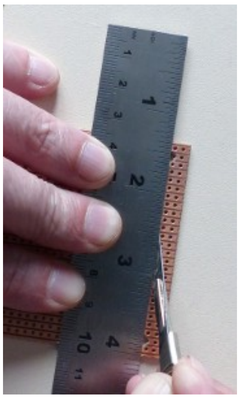
The strip board contains a number of link wires, I have used 22swg tinned copper wire.

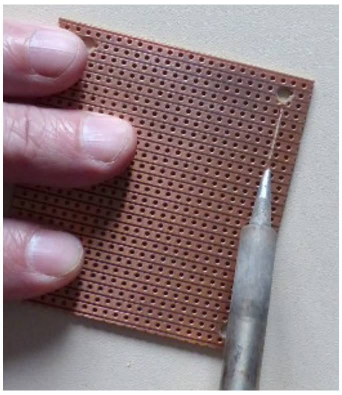
You will also need some stand-offs to support the strip board when mounted to the Raspberry Pi, an enclosure to house the project, one M20 cable gland for ethernet cable entry, and four M12 cable glands for the remaining cable entry points and air vent.

Building the main weather station board

Assuming that you now have all the necessary parts to build your weather station start by cutting the strip board to the correct size. Basically the strip board needs to have 24 complete holes along the copper track by 26 complete holes.

Next cut the tracks where SK1 (40 way GPIO connector) is located. This requires the use of a sharp knife, a straight edge and a soldering iron. Use the sharp knife against the straight edge to create two parallel score lines on the copper trace in between the two rows of pins where SK1 will be placed. Now use the hot soldering iron tip to remove the copper trace between the score lines by gently pushing the soldering iron away from you. The pictures below may help understand the steps required:

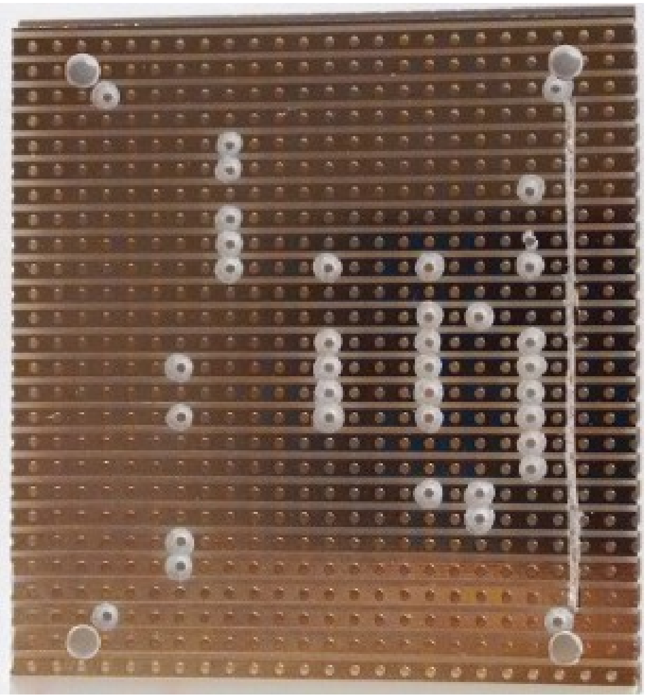




Carefully inspect the cut traces to make shore there are no short circuits.

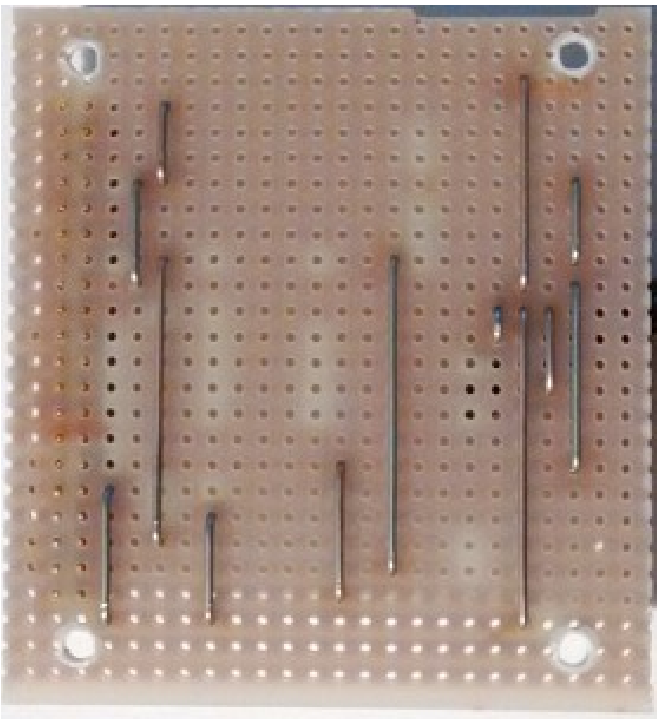
Next we need to cut the remaining traces using a tool known as a strip board cutter, if you do have one of these to hand you can make your own by placing a 3.5mm drill bit in a handle:

You then need to cut the rest of the traces so that your strip board looks like the one below:



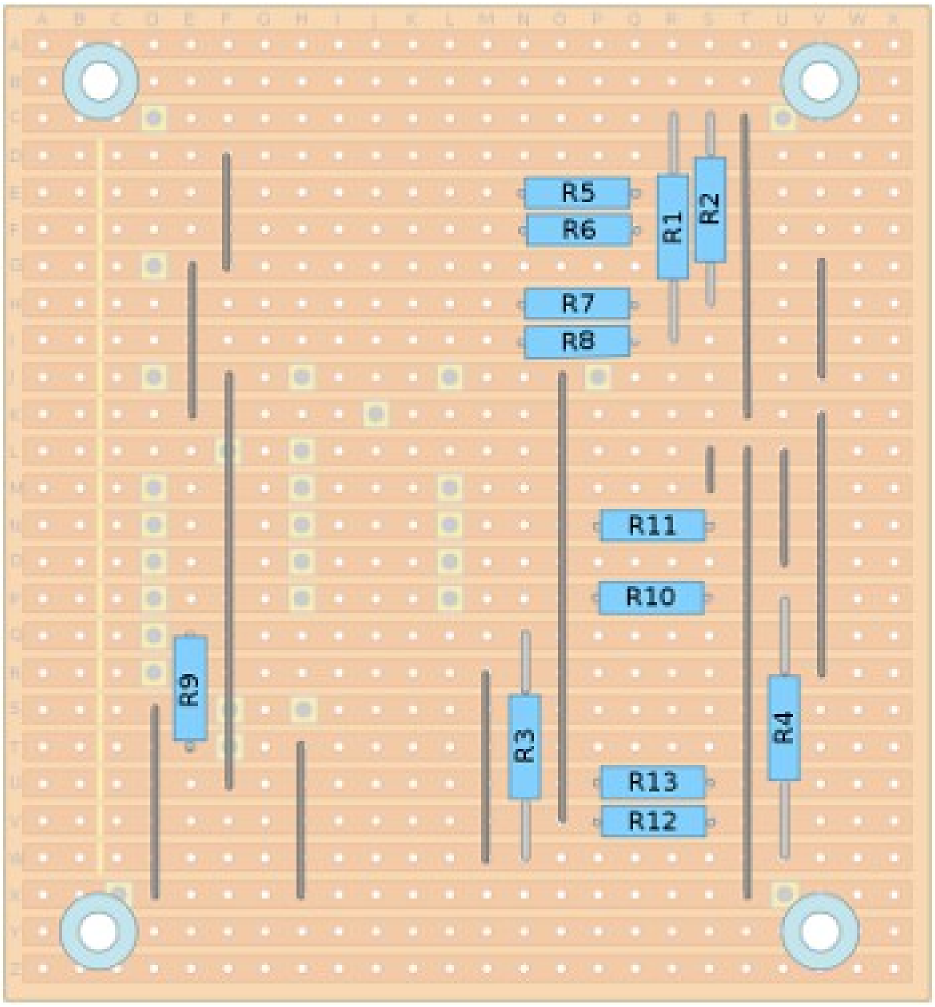
Carefully inspect the cut traces to make shore there are no short circuits, preferably with a magnifying glass or eye loop. This is also the time to drill the 3mm fixing holes shown nearest the corners on the picture above.

The next step is to fit the tinned copper wire links from the component side (the side without copper traces) of the strip board. I like the links to look straight so I normally take hold of the reel of wire and pull the end of the wire with a pair of pliers until the wire is straight. It is best to fit these one at a time by first putting right angled bend at both ends so the that wire fits through the correct strip board holes, turn the strip board over, solder both ends and then trim the wires. Repeat until the strip board looks like the one below:



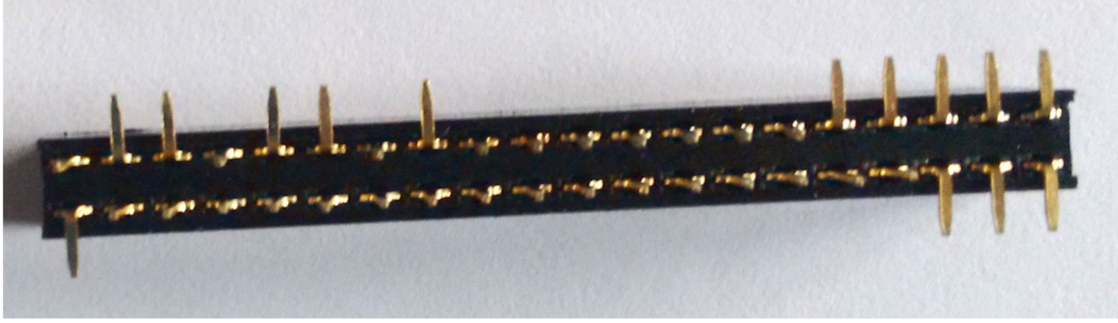
Now fit the resistors to the component side using the following table and image to identify which resistor goes where. The resistor value is coded on the resistor body using coloured bands, if you do not no how to read the resistor value using the coloured bands there are plenty of online tools and guides that can help. Alternatively if you have a multimeter you can measure the value of each resistor prior to fitting.

Reference	Value	Reference	Value
R1	3K3	R8	1K
R2	3K3	R9	100R
R3	3K3	R10	1K
R4	75K	R11	1K
R5	100R	R12	33R
R6	100R	R13	33R
R7	1K		



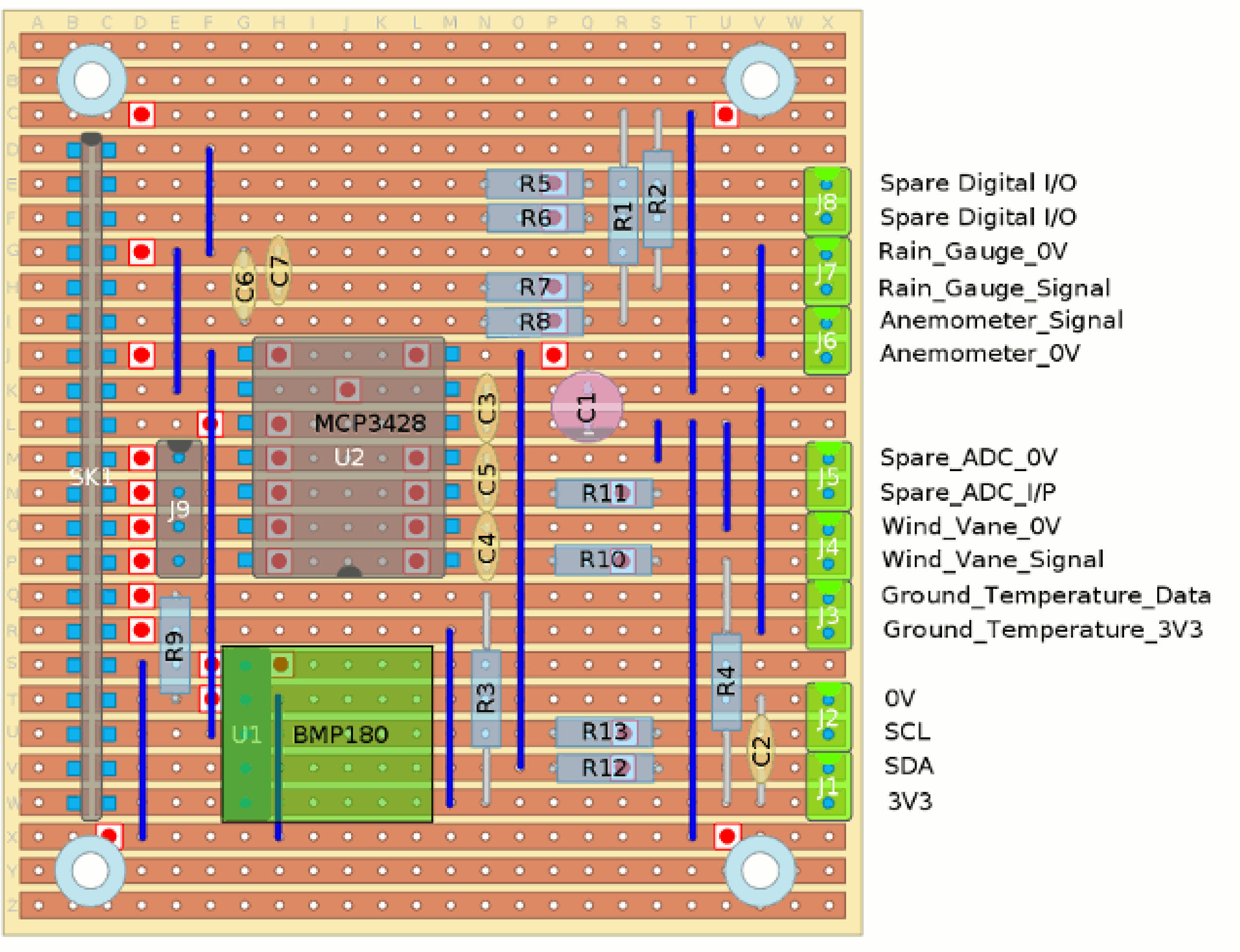
After fitting each resistor you will need to trim the legs close to the solder joint with wire cutters.

We now need to fit the 40 way connector to the solder side of the strip board (the opposite side to the resistors). This is probably the most difficult part to solder so you need to be very careful and patient when soldering this. Fortunately we do not need to solder all 40 pins and for the pins that need soldering we can make life easier by bending the connector pins at 90 degrees. The 40 way connector should look like this when done:



Now mount this to the solder side of the strip board taking care to get the connector the right way round.

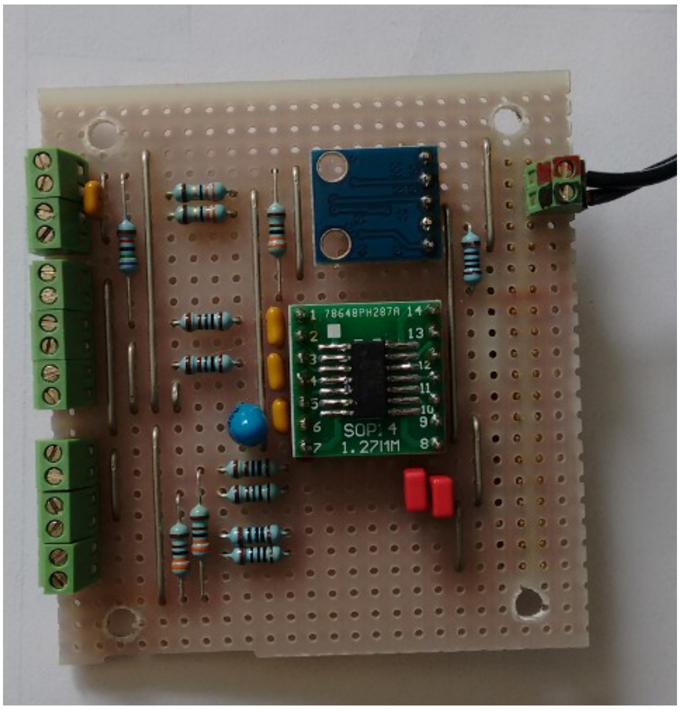
Finally fit the remaining components using the image below as a guide to where each component fits on the strip board:



Connecting Power

There are two possible methods of connecting power to the Pi weather station.

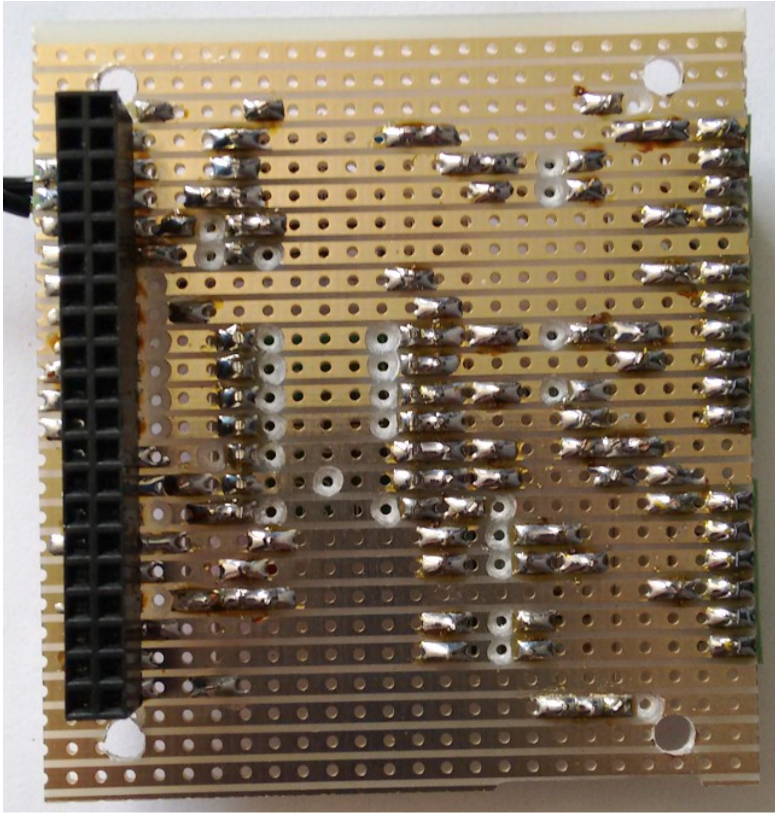
1. Through the normal Raspberry Pi power socket. To do this you will need an adapter cable to convert the TP-Link POE barrel plug to the microUSB connector used on the Pi. This is the safest method of connecting power.
2. You can connect the power directly to the weather station board. To do this you need to cut the TP-Link barrel connector off and connect the wires to pins 4 & 6 of the 40 way connector. I used this method by adding an additional 2 way screw terminal connector on the the weather station board as shown on the right hand edge of the image below, you will also see that I have marked the terminals with a marker pen to try and prevent me from accidentally connecting wires the wrong way round:



This method is far more prone to error and should only be used as a last resort.

Testing the weather station board

At some stage you need to plug the weather station board into the Raspberry Pi and if the board contains build errors there is a real chance that the Raspberry Pi may be damaged. If you have a multi-meter you can use the following steps to help minimise the risk.

2	1		3V3
4	3		SDA
6	5		SCL
8	7		0V
10	9		
12	11		Ground_Temperature_3V3
14	13		Ground_Temperature_Signal
16	15		Wind_Vane_Signal
18	17		Wind_Vane_0V
20	19		Spare_ADC_I/P
22	21		Spare_ADC_0V
24	23		
26	25		
28	27		Anemometer_0V
30	29		Anemometer_Signal
32	31		Rain_Gauge_Signal
34	33		Rain_Gauge_0V
36	35		Spare Digital I/O
38	37		Spare Digital I/O
40	39		

The first step is to check the resistance between the pins on the 40 way connector, use the table below to confirm your results are within the correct range, the cell colour indicates the multi-meter probe colour to be used, no test is required for white cells:

Pin #			Pin #		Expected Results
			2	1	
2	1	to	4	3	4450 – 4950 Ohm
4	3	to	6	5	8900 – 9900 Ohm
6	5	to	8	7	7700 – 8500 Ohm
8	7	to	10	9	Open circuit
10	9		12	9	Pins 9 – 21 are unused. A worthwhile check is to test each pin for continuity to the other side of the cut trace. In each case the test must show open circuit.
12	11		14	11	
14	13		16	13	
16	15		18	15	
18	17		20	17	
20	19		22	19	
22	21		24	21	
24	23	to	26	25	Open circuit
26	25	to	28	27	Open circuit
28	27	to	30	29	Open circuit
30	29	to	32	31	8150 – 9050 Ohm
32	31	to	34	33	Open circuit
34	33	to	36	35	Open circuit
36	35	to	38	37	Open circuit
38	37	to	40	39	Open circuit
40	39				

You can also double check that you have no shorts between the odd and even number pins.

Next check the screw terminal connections:

Screw Terminal Name		**Screw Terminal Name**	**Expected Result**
3V3	to	3V3	
SDA		SDA	4450 – 4950 Ohm
SCL		SCL	4450 – 4950 Ohm
0V		0V	Varying reading
Ground_Temperature_3V3		Ground_Temperature_3V3	<1 Ohm
Ground_Temperature_Signal		Ground_Temperature_Signal	3200 – 3500 Ohm
Wind_Vane_Signal		Wind_Vane_Signal	67000 – 83000 Ohm
Wind_Vane_0V		Wind_Vane_0V	Varying reading
Spare_ADC_I/P		Spare_ADC_I/P	
Spare_ADC_0V		Spare_ADC_0V	
Anemometer_0V		Anemometer_0V	Varying reading
Anemometer_Signal		Anemometer_Signal	3200 – 3500 Ohm
Rain_Gauge_Signal		Rain_Gauge_Signal	3200 – 3500 Ohm
Rain_Gauge_0V		Rain_Gauge_0V	Varying reading
Spare Digital I/O		Spare Digital I/O	
Spare Digital I/O		Spare Digital I/O	

Finally check the connections from the 40W to the screw terminals:

Pin #			**Screw Terminal Name**	**Expected Result**
2	1	to	3V3	<1 Ohm
4	3	to	SDA	30 – 36 Ohm
6	5	to	SCL	30 – 36 Ohm
8	7	to	Ground_Temperature_Signal	94 – 106 Ohm
10	9	to	0V, Wind_Vane_0V & Spare_ADC_0V	<1 Ohm
12	11			
14	13			
16	15			
18	17			
20	19			
22	21			
24	23			
26	25	to	Anemometer_0V & Rain_Gauge_0V	<1 Ohm
28	27			
30	29	to	Anemometer_Signal	950 – 1050 Ohm
32	31	to	Rain_Gauge_Signal	950 – 1050 Ohm
34	33			
36	35	to	Spare Digital I/O	94 – 106 Ohm
38	37	to	Spare Digital I/O	94 – 106 Ohm
40	39			

That's it, if your results match those above you are now ready to connect the weather station board to a Raspberry Pi. Assuming you already have a Raspberry Pi with Raspian installed and working, with the power disconnected fit the weather station board to the Raspberry Pi, make sure that nothing on the solder side of the weather station board comes into contact with the Raspberry Pi.

Apply power to the Pi through the normal power connector and make sure that the Pi boots up and runs as normal. If all is well you can now proceed to the next section.

Connecting the sensors

Software installation and modification

This weather station is designed to make use of the official Raspberry Pi weather station software with as few changes as possible. You will therefore need to install the software as described here:

https://www.raspberrypi.org/learning/weather-station-guide/software-setup.md

You need to follow the "MANUAL INSTALLATION" section, skipping the section "Setting up the real-time clock". This is because this design does not include a real-time clock, it gets the time from the internet just like a normal Raspberry Pi.

Follow the official guide until you reach the section "AUTOMATE UPDATING OF THE DATABASE" then follow the steps below.
Note: The results you will see in the "TESTING THE SENSORS" section will be slightly different (UU and 6a will be missing), this is because this design does not include the real time clock and air quality sensor.

Before proceeding with "AUTOMATE UPDATING OF THE DATABASE" we need to modify the files "log_all_sensors.py" and "crontab.save" by following the steps below. The modifications will disable the air quality sensor, this design does not include this, and stop data from being sent to the Oracle Apex database as this is reserved for schools using the official weather station.

First lets make a back-up of the files we want to modify:

```
cd ~/weather-station
cp crontab.save crontab.save.orig
cp log_all_sensors.py log_all_sensors.py.orig
```

Check that the backups exist:

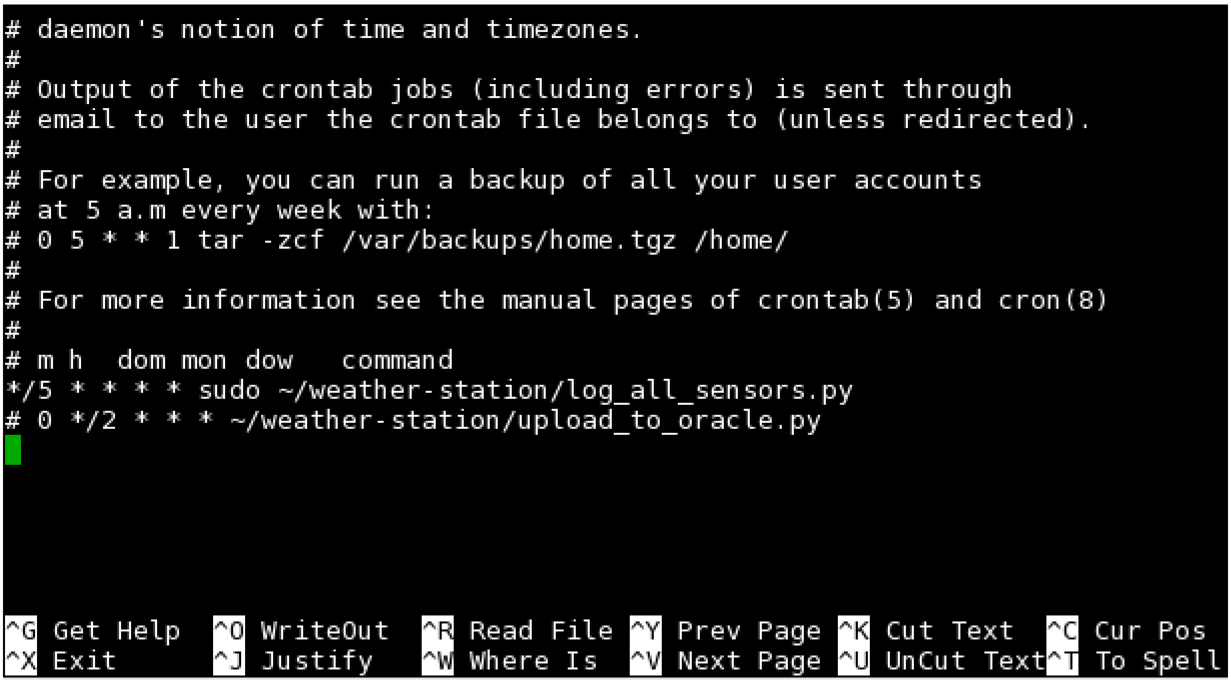
```
ls *.orig
```

You should see the back-up files "crontab.save.orig" and "log_all_sensors.py.orig" listed, if not then repeat the steps above and try to find were this step failed.

To prevent data from being sent to the Oracle Apex database we need to comment out the last line in the file "crontab.save". To do this open "crontab.save" in the text editor nano and place a # at the beginning of the last line:

```
nano ~/weather-station/crontab.save
```

Move the cursor to the last line and enter #, your terminal screen should now look like this:



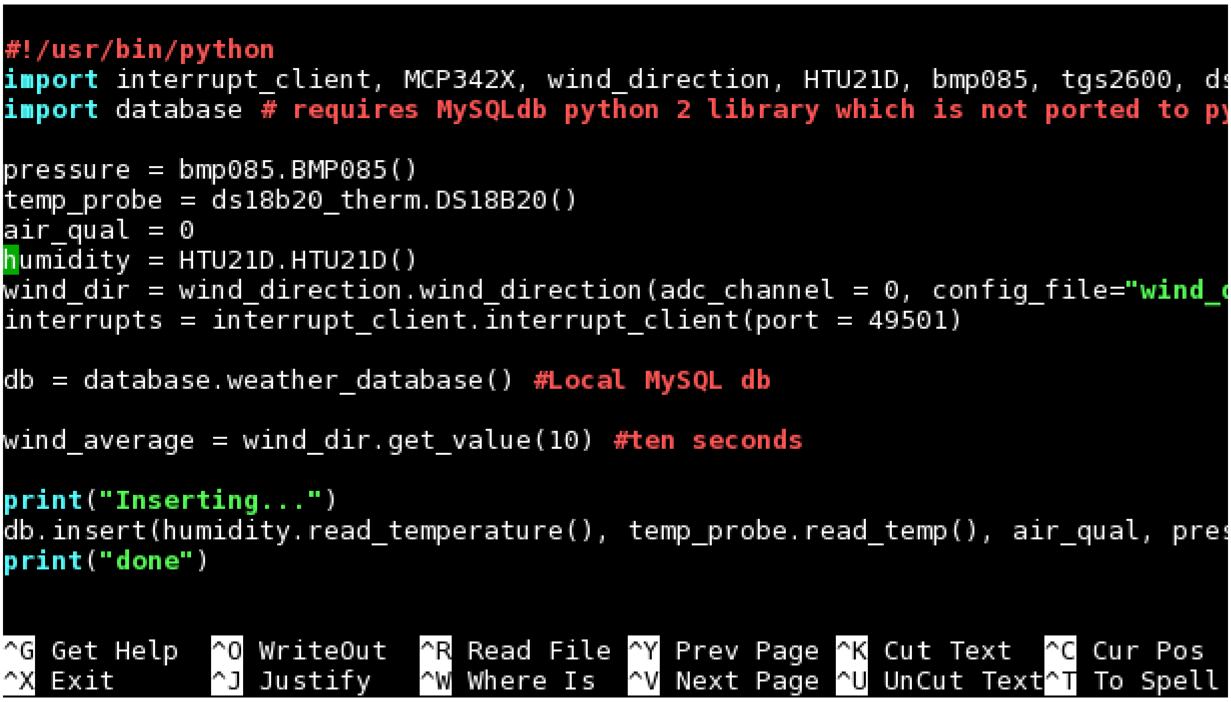
```
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/5 * * * * sudo ~/weather-station/log_all_sensors.py
# 0 */2 * * * ~/weather-station/upload_to_oracle.py
```

Press ctrl-O to save the modifications followed ctrl-X to exit nano.

We need to stop air quality sensor readings by modifying “log_all_sensors.py”. Open “log_all_sensors.py” in nano and modify the line that starts with “air_qual =” to read “air_qual = 0”. You also need to modify “air_qual.get_value()” to read “air_qual” in the line that starts “db.insert”:

nano ~/weather-station/log_all_sensors.py

The modified file should look like this:



```
#!/usr/bin/python
import interrupt_client, MCP342X, wind_direction, HTU21D, bmp085, tgs2600, ds
import database # requires MySQLdb python 2 library which is not ported to py

pressure = bmp085.BMP085()
temp_probe = ds18b20_therm.DS18B20()
air_qual = 0
humidity = HTU21D.HTU21D()
wind_dir = wind_direction.wind_direction(adc_channel = 0, config_file="wind_c
interrupts = interrupt_client.interrupt_client(port = 49501)

db = database.weather_database() #Local MySQL db

wind_average = wind_dir.get_value(10) #ten seconds

print("Inserting...")
db.insert(humidity.read_temperature(), temp_probe.read_temp(), air_qual, pres
print("done")
```

Press ctrl-O to save the modifications followed ctrl-X to exit nano.

You can now go back to the official guide and proceed with “AUTOMATE UPDATING OF THE

DATABASE”. You should skip the section titled “UPLOAD YOUR DATA TO THE ORACLE APEX DATABASE”.

Enclosure

Finding and buying a suitably large IP rated enclosure can be surprisingly difficult, especially when you add additional requirements such as UV resistant and fixing points outside the sealed area.