

## Getting started with Enviro pHAT

Enviro pHAT is an environmental sensing board that lets you measure temperature, pressure, light, colour, motion and analog sensors. It's the perfect board for building a tiny monitoring station with Pi Zero that you can stick on a shelf and keep track of conditions in the room.

Here, we'll go through putting the board together, installing the software, using the software, and a quick example of how to log values from all of the sensors.

For this tutorial, you'll need:

- A Raspberry Pi Zero with soldered male header
- An Enviro pHAT
- A USB wifi dongle (we recommend the official Raspberry Pi one)
- A micro USB to full-size USB adaptor cable or shim
- A 5V, 2A micro USB power supply

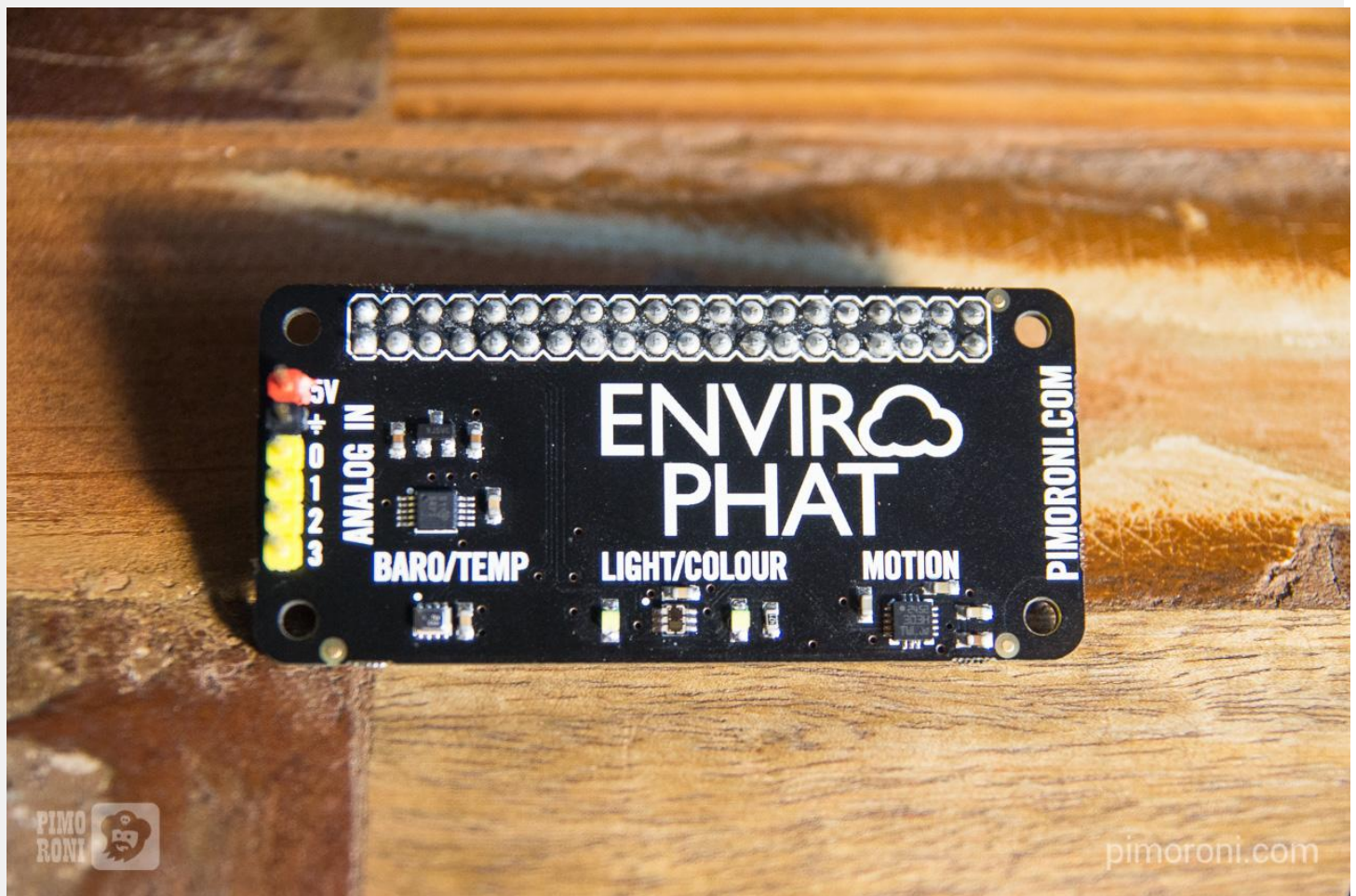
## Assembling the board

Your Enviro pHAT will come un-soldered, with a strip of 2x20 pin female header provided. You'll need to solder that on to Enviro pHAT, to allow you to connect it to your Pi Zero's male header.

We've got a handy guide [here](#) on how to solder pHATs, so check that out if you're new to soldering.

Another option is to solder the Enviro pHAT straight onto the male header of the Pi Zero for a super-slim solution, although this means that you'll not be able to use any other pHATs with your Pi Zero.

We also provide a small strip of male header for the analog input pins, but you only need to solder this if you'll be using the analog inputs. If you are soldering it, make sure that you have the long ends of the male header pointing up the way!



# Installing the software

We always recommend using the most up-to-date version of Raspbian, as this is what we test our boards and software against, and it often helps to start with a completely fresh install of Raspbian, although this isn't necessary.

As with most of our boards, we've created a really quick and easy one-line-installer to get your Enviro pHAT set up. We'd suggest that you use this method to install the Enviro pHAT software.

Open a new terminal, and type the following, making sure to type 'y' or 'n' when prompted:

```
curl -sS get.pimoroni.com/envirophat | bash
```

Once that's done, it's probably a good idea to reboot your Pi to let the changes propagate.

# Using the software

Open a new terminal and type `sudo python` to open a new Python prompt.

The Python library is partitioned into separate modules for `light`, `motion`, `weather`, `analog` and `leds`.

## Light

To import the light module, that we can then use to read the overall light level and RGB colour values, type the following:

```
from envirophat import light
```

The TCS3472 sensor reads four different values - clear, red, green and blue - with the clear value being the overall light level and the background against which the red, green and blue values are measured.

The `light` module has two methods (well, more than two, but you'll probably not need the others) that you can use.

The first is `.light()`, which gives the overall light level from the clear reading. Type the following to print the light level value:

```
print light.light()
```

The second is `.rgb()`, which gives the RGB colour values. Type the following to print out the RGB values:

```
print light.rgb()
```

You'll notice that this prints out a tuple of the RGB values and we can easily unpack this to three separate variables `r`, `g` and `b` in a single line:

```
r, g, b = light.rgb()
```

To get a more accurate colour reading, we've added two white LEDs either side of the light/colour sensor that should reflect more light back to the sensor. You can turn these on by typing:

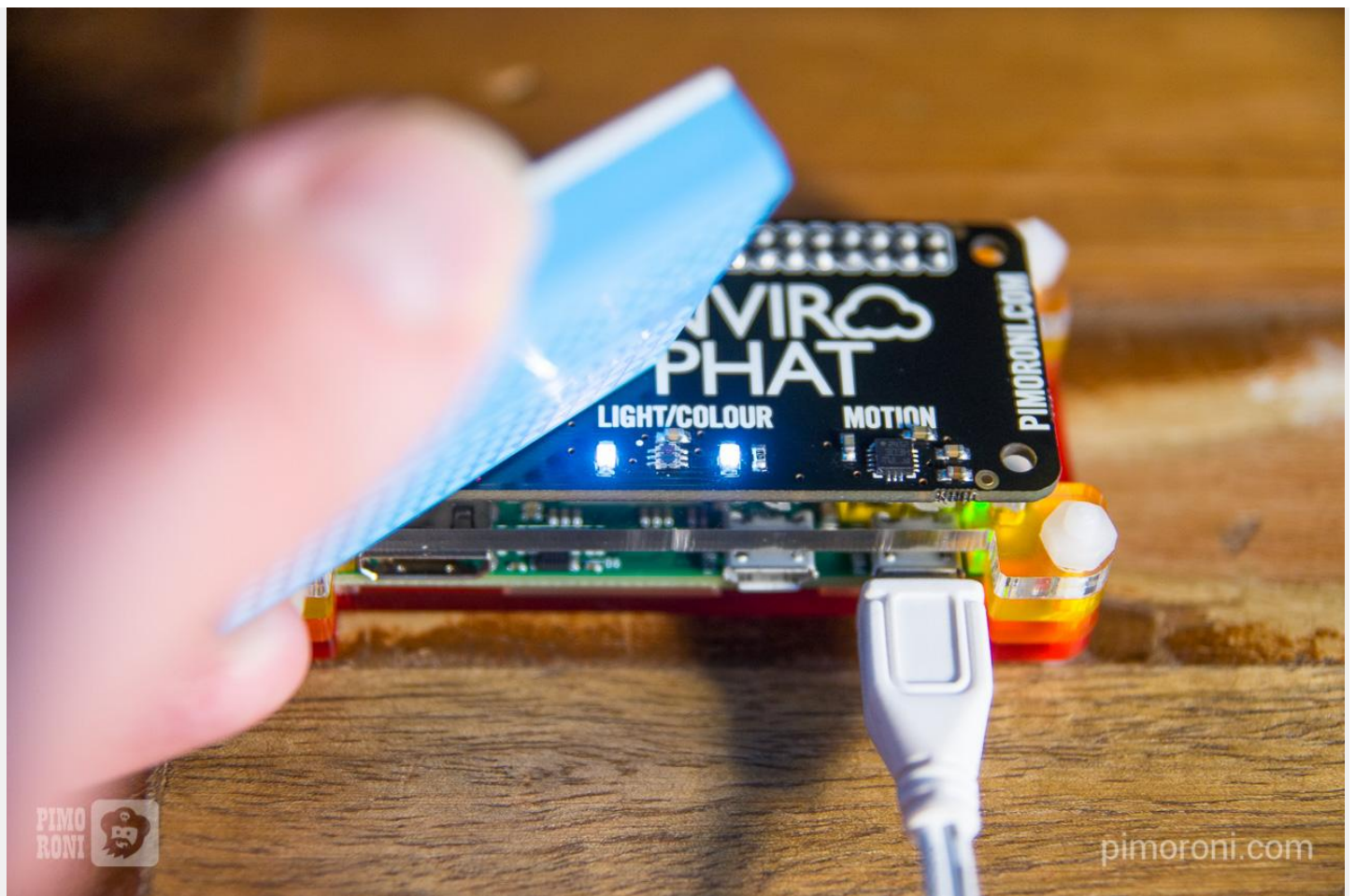
```
from envirophat import leds
```

```
leds.on()
```

And to turn them off again:

```
leds.off()
```

Try holding something coloured above the colour sensor, above the text reading 'LIGHT/COLOUR' and see how the RGB values change.



## Weather

Just like the `light` module, we can import the `weather` module by typing:

```
from envirophat import weather
```

The BMP280 sensor can read temperature and pressure, and the `weather` module provides two methods for reading these values.

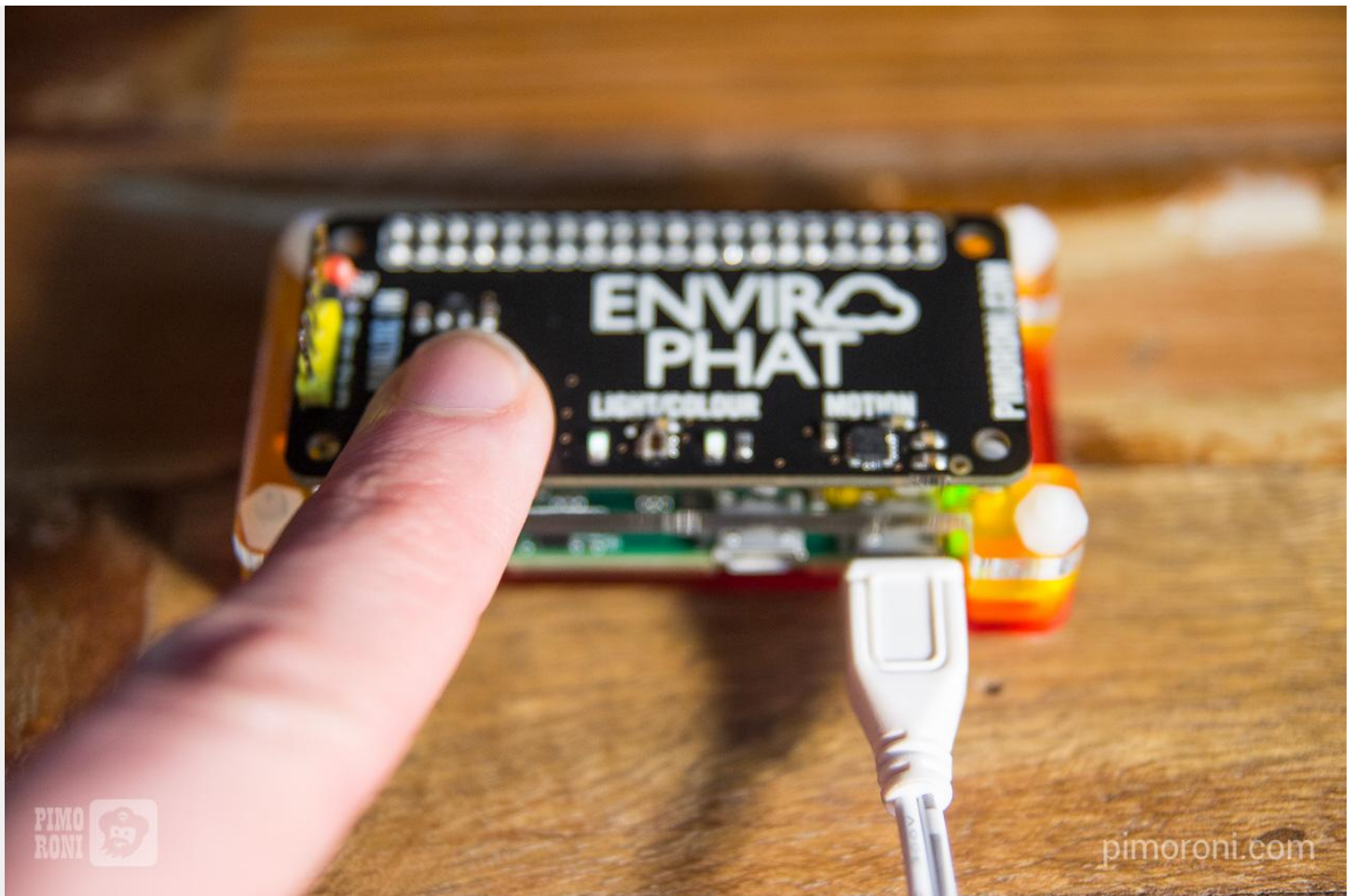
To read the temperature in degrees Celsius, type the following:

```
print weather.temperature()
```

And to read the pressure in hPa (hecto Pascals), type the following:

```
print weather.pressure()
```

Try touching the BMP280 on Enviro pHAT, just below the text that reads 'BARO/TEMP' and then print the temperature again to see how it changes.



## Motion

The LSM303D accelerometer/magnetometer sensor can detect orientation, motion and heading (compass position) of the board.

To import the motion module, type:

```
from envirophat import motion
```

Similar to the `light.rgb()` method, `motion.accelerometer()` returns a tuple of `x`, `y` and `z` variables for the three axes of movement.

To read the accelerometer values, assign them to the variables `x`, `y` and `z`, and then print them, type the following:

```
x, y, z = motion.accelerometer()
print x, y, z
```

Try using a `while` loop to constantly print the `x`, `y` and `z` variables, and then move Enviro pHAT around and see how the values change:

```
import time

while True:
    print motion.accelerometer()
    time.sleep(0.1)
```

This will print out the accelerometer values every 0.1 seconds.

The heading is equivalent to a compass position, although some fiddling is required to convert it into numbers where north is at 0 degrees and south at 180 degrees.

To read and print the heading, type:

```
print motion.heading()
```

If you'd like to set north to 0 and then calculate accurate compass headings, then you can do the following:

Point the Enviro pHAT with the text the right way up (relative to you) towards north, then take a heading reading:

```
north = motion.heading()
```

Now, if you do the following, you can calculate the correct compass heading:

```
corr_heading = (motion.heading() - north) % 360
```

```
print corr_heading
```

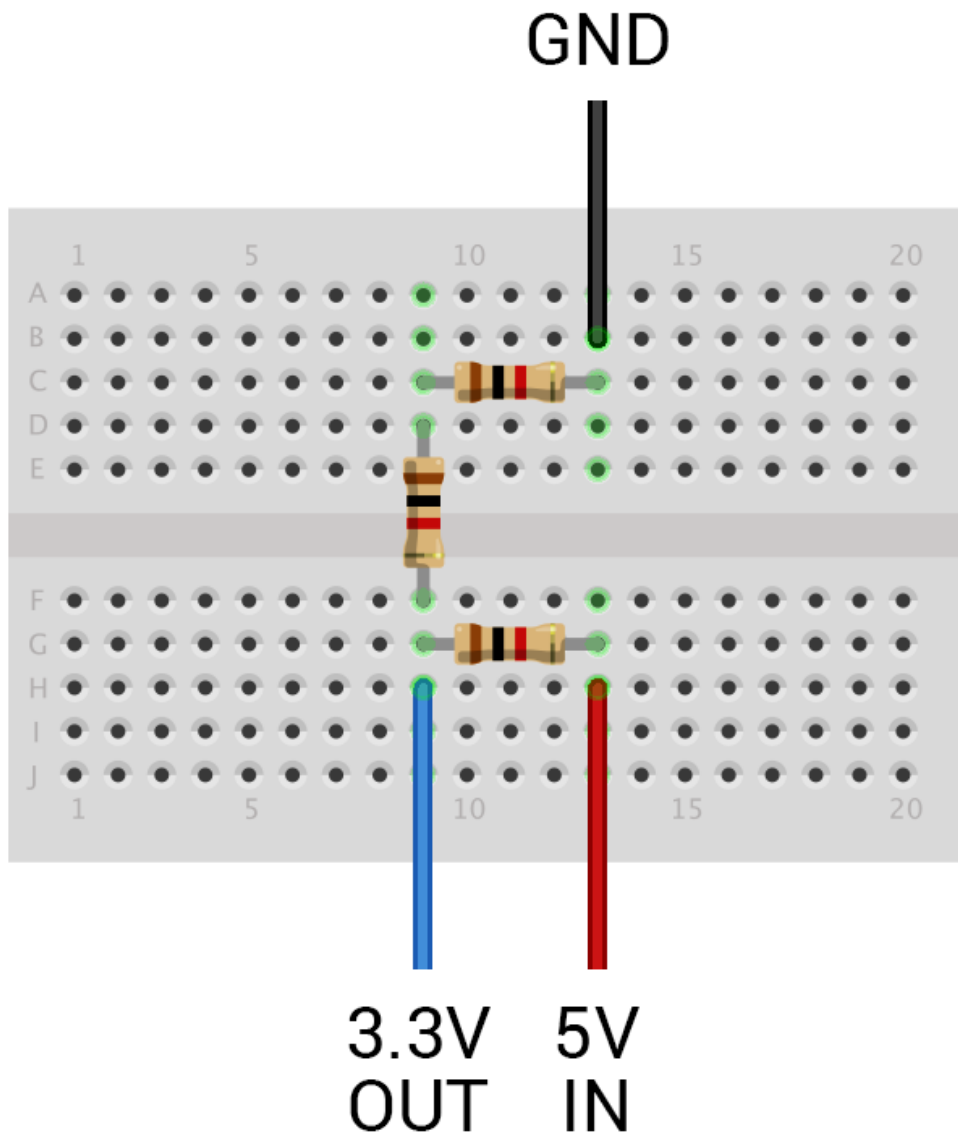
We just subtract our north reading from the current reading and then take the modulo 360, so that the values wrap around the compass correctly.

## Analog

The ADS1015 sensor on Enviro pHAT is an analog to digital converter (ADC) that can read analog signals with 3.3V logic, over 4 channels. This means that to use analog sensors that use 5V logic, we need to use a voltage divider to drop the 0-5V output from the analog sensor down to 0-3.3V.

An easy way to drop 5V down to 3.3V is to use three resistors of equal value and then, by tapping the right junction, you can get  $2 \times \frac{1}{3}$  of 5V which equals 3.3V!

In the example below, we're using three 1 k $\Omega$  resistors, and your analog output (from the sensor) would be connected to the red wire, ground to the black wire and the Enviro pHAT analog input to the blue wire.



We have a range of analog sensors [in our shop](#) that could be used with this voltage divider setup, like analog joysticks and gas sensors, as well as some that use 3.3V logic and could be used without the voltage divider.

To read the analog values from the ADC, just type:

```
from envirophat import analog
print analog.read_all()
```

Or to read from just one channel, e.g. channel 0, type:

```
print analog.read(0)
```

## Logging values

Logging the values from the sensors on Enviro pHAT is super-easy!

Here, we'll just poll the sensors at a given interval - every second - and then write those values to a tab-separated file. Tab-separated files are great for doing some analysis of your data further down the line but, if you wanted to do this over a longer period of time and with large amounts of data, you'd probably be better off setting up a SQLite3 database or similar in which to store the data.

Here's all of the code we'll use to log our data, and then we'll go through, line-by-line, what it all does:

```
import time
from envirophat import light, motion, weather, leds

out = open('enviro.log', 'w')
out.write('light\trgb\tmotion\theadings\ttemp\tpress\n')

try:
    while True:
        lux = light.light()
        leds.on()
        rgb = str(light.rgb())[1:-1].replace(' ', '')
        leds.off()
        acc = str(motion.accelerometer())[1:-1].replace(' ', '')
        heading = motion.heading()
        temp = weather.temperature()
        press = weather.pressure()
        out.write('%f\t%s\t%s\t%f\t%f\t%f\n' % (lux, rgb, acc, heading, temp, press))
        time.sleep(1)

except KeyboardInterrupt:
    leds.off()
    out.close()
```

First, we import `time` to allow us to add a one second delay between readings, and the modules from the `envirophat` library that we need.

```
import time
from envirophat import light, motion, weather, leds
```

Next, we open an output file, into which we can write our data, called `enviro.log`, and write the column headers to the file.

```
out = open('enviro.log', 'w')
out.write('light\trgb\tmotion\theadings\ttemp\tpress\n')
```

We're going to put our `while True:` loop into a `try` clause, so that we can clean things up later in the `except KeyboardInterrupt` clause, which will activate when you control-c exit the script.

```
try:
    while True:
        lux = light.light()
        leds.on()
        rgb = str(light.rgb())[1:-1].replace(' ', '')
        leds.off()
        acc = str(motion.accelerometer())[1:-1].replace(' ', '')
        heading = motion.heading()
        temp = weather.temperature()
        press = weather.pressure()
        out.write('%f\t%s\t%f\t%f\t%f\n' % (lux, rgb, acc, heading, temp, press))
        time.sleep(1)
```

We're taking readings from each of the sensors and, for the `rgb` and `acc` readings, we're doing a little fiddling to reformat the tuple converted into a string to remove the brackets at the beginning and end, and remove the spaces.

We switch the LEDs on before the colour reading is taken, and then off again after, because if we had them on while the `lux` reading was being taken, it might affect the reading taken and not reflect the true ambient light level.

The values are written to a new line in the output file, using the `%` placeholder notation, to make the syntax a little neater. The tabs are represented by `\t` and new lines by `\n`.

`time.sleep(1)` introduces a 1 second delay before the next reading is taken, although each iteration of the `while` loop will take some time, so the readings won't be taken exactly a second apart.

You may also want to add a timestamp to your readings. You can do this with something like the following:

```
import datetime

timestamp = datetime.datetime.now().isoformat()
```

Last of all, we have our `except KeyboardInterrupt` clause that activates when we exit the script with control-c:

```
except KeyboardInterrupt:
    leds.off()
    out.close()
```

This switches off the LEDs and closes the output file.

Save the code above to a file named something like `enviro_logger.py` and then run it by typing `sudo python enviro_logger.py`.

## Taking it further

Why not try to store the sensor data in a SQLite3 database, and then create a web front-end to display the data?

There are loads of possibilities with Enviro pHAT, especially when coupled with some of our other pHATs and HATs like Scroll pHAT, Unicorn pHAT and Display-O-Tron HAT.