

Level 1 - Explorer HAT Introduction - Python

The Explorer HAT from Pimoroni is an easy to use, general purpose addon board (HAT) for the Raspberry Pi.

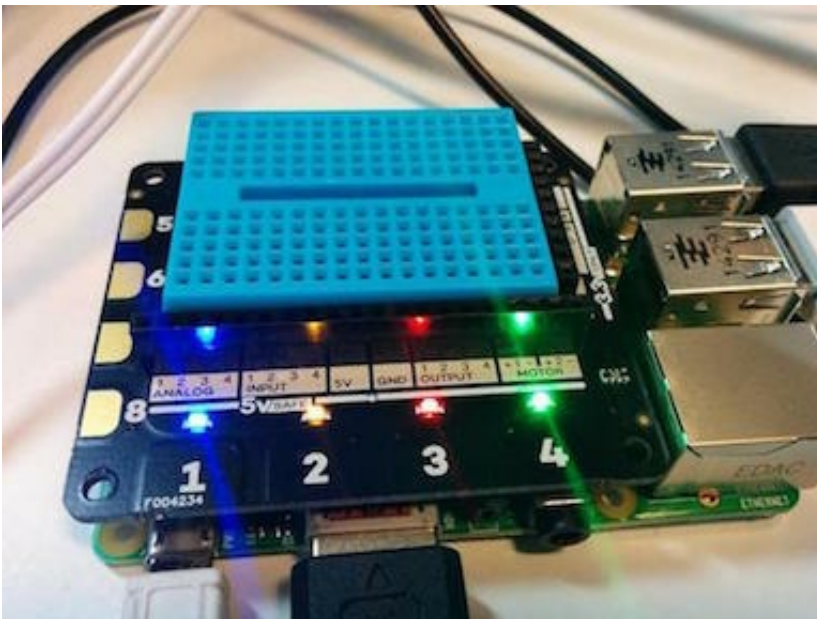
This activity includes an introduction to using its built in inputs (x8 capacitive touch inputs) and outputs (x4 coloured LEDs).

What you will need

- x1 Explorer Pro HAT (Hardware Attachment on Top).
- x4 crocodile clips.
- Some tinfoil or other conductive material (fruit, playdough etc).

Attaching your Explorer HAT

If your Explorer HAT isn't already attached, nab a volunteer to give you a hand to attach it as it can be a fiddle.



Opening up Python (IDLE)

For this project we will be using Python 3. To start working in Python, open up IDLE3 by opening a terminal and typing `sudo idle3 &`. Within a few seconds IDLE 3 should load up. Finally, click on file, new window and you are ready to go.

Lets get coding!

Code setup

At the start of each Python program for the Explorer HAT, we need to import the Explorer HAT library and the time library (to allow us to use `time.sleep()`). To do that, we need to add this section to the start of each of our programs

```
import explorerhat as eh
import time
```

LEDs

The Explorer HAT includes 4 coloured LEDs! We can play with these super easily. Try the code below

```
import explorerhat as eh
import time
eh.light.red.on()
time.sleep(1)
eh.light.red.off()
```

The program above simply imports the required libraries, turns the red LED on, waits 1 second then turns the red LED off.

The library also includes some extra functions, other than just `.on()` and `.off()`. Why don't you try

```
import explorerhat as eh
import time
eh.light.red.fade(100, 0, 3)
```

Can you figure out what each of the numbers means? Try changing them and see what happens.

There is also a few other available functions including

```
.blink( on_time, off_time ),
.pulse( fade_in_time, fade_out_time, on_time, off_time ) and
.stop()
```

LED Challenges

1. Can you turn each LED on and off one after each other?
2. Can you create an awesome light show using a combination of `.on()`, `.off()`, `.fade()`, `.blink()` and `.pulse()` ?

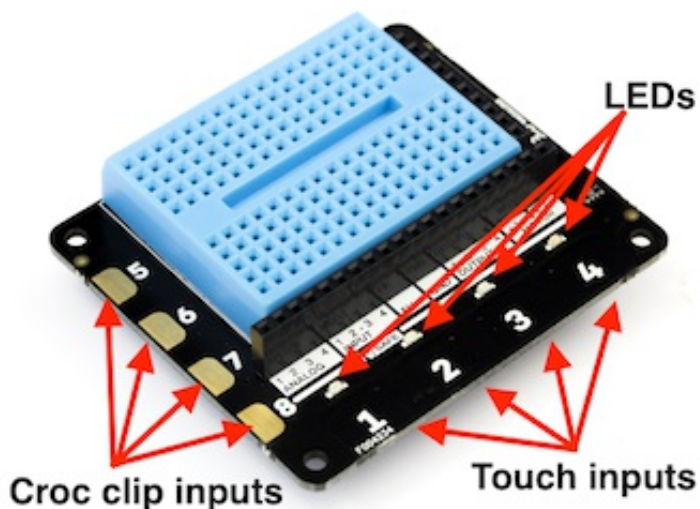
Inputs

The Explorer HAT also has 8 capacitive touch inputs. Capacitive touch is the same technology used in most touch screens.

It allows you to simply touch the pads, instead of having to physically press in a button.

The 8 capacitive touch inputs are split into 2 sections.

- 1-4 are for pressing directly with your fingers.
- 5-8 are for attaching to other "stuff" using crocodile clips, for example fruit, playdough or tinfoil.



Lets try using pad number 1. To use it, in Python we have 2 options. The simplest way is using what is called "Event Driven mode", basically it keeps an eye in the background to see if the pad is pressed, and if it does it triggers an event. In the case of Python, it runs whatever function we assign to it. The cool bit is, you can be waiting for that event and set off your own code when it is triggered!

```
import explorerhat as eh
import time

def awesome_function(channel, event):
    print("Awesomeness has been pressed!")

eh.touch.one.pressed( awesome_function )
```

In the code above, after importing the libraries as usual, we define what is called a **function** (called `awesome_function`). A function is simply a mini program that we are teaching the computer about. At any time, we can ask the computer to run that mini program from inside our main program.

We then register our new function as an **event handler**, basically which bit of code to run when the event is set off. In this case, the event will be set off when we press on pad 1.

You can put anything inside the function, but **make sure you follow the indentation!**

Functions are indented in one level with the tab key.

Input Challenges

1. Combining this and the LED activity, can you trigger an LED to come on when touch pad 2 is touched?
2. Can you connect some tinfoil to the Explorer HAT to crocodile pad 5 and use it as a button to trigger your light show from the outputs section?

What's next?

Now you have a basic understanding of using the built in inputs and outputs of the Explorer HAT, you are ready to move onto Level 2 Explorer HAT Python activities!