

Wheel of Fortune – Powered by Raspberry Pi and Explorer HAT Pro

Introduction

Explorer HAT Pro is an add on board from the lovely people at Pimoroni. The board enables anyone to easily learn physical computing using Python. Physical computing is an exciting and innovative area of technology and provides children with massive incentive to learn computing.

Learning objective

In this project we will build a motorised wheel which will rotate for a random amount of time and when it stops it will select a question / person from the wheel.

Resources

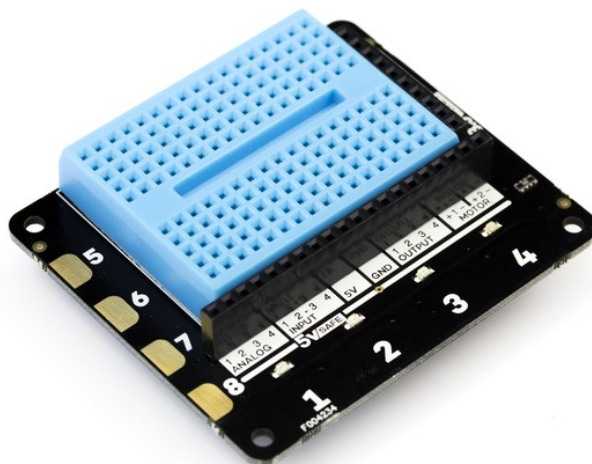
Game wheel pieces and code can be downloaded from

Lesson Summary

- An introduction to Explorer HAT Pro
- Build our hardware and wheel
- Program our project using Python

An introduction to Explorer HAT Pro

Explorer HAT Pro is an £18 add on board for the Raspberry Pi that enables anyone to learn physical computing and electronics. Fixing over the GPIO pins Explorer HAT Pro provides a flexible platform for tinkering.



Build our hardware and wheel

- Turn off your Raspberry Pi and firmly attach your Explorer HAT Pro.
- Connect the motor to Motor 1
- Print off and cut out the wheel and selector.
- Affix wheel to motor using blutack and also secure your motor to the desk using more blutack.
- Set up your Raspberry Pi and power it up.

Program our project using Python

Open the IDLE application as root, you can do this via LXterminal by typing

```
sudo idle
```

When IDLE3 has finished loading click on File >> New to open a blank editor window.

Let's get coding!

Import Modules

Our first line will be to import the Explorer HAT module into our code. This enables us to use the Explorer HAT Pro with Python.

```
import explorerhat
```

Next we will import two more modules one which will control the speed of our game and another to enable our project to select a random number.

```
from time import sleep  
from random import randint
```

You can see that these modules have been imported differently. Rather than import the whole library we only import the libraries that we need.

Create a function

Functions are awesome! Using functions we can create a sequence of code that can be used whenever we need it. Our function controls the spin of our wheel which is triggered by pressing the number 1 button on our Explorer HAT Pro

.

We start by defining the name of the function, in this case `wheel`,

```
def wheel(channel, event):
```

You will see that in brackets the words `channel` and `event`, these are arguments that ExplorerHAT Pro needs in order to use the button with our project.

Now we will create a for loop which will iterate five times...but why? Well in order to make sure

that our numbers are as random as can be the for loop will overwrite the value of the variable that we will create next.

```
for i in range(5):
```

We next create a variable called `duration` and the data that we will store inside of our variable will be the random integer chosen using the following code.

```
duration = randint(1,10)
```

We now turn on the power to `motor 1` of our Explorer HAT Pro, this starts the motor turning. The motor will continue to turn while the project sleeps, and this duration is the randomly chosen number that we earlier created.

You will notice that there is a `print(duration)` function in the code, this will output the random number to the Python shell for debugging purposes.

The last step in the function turns off the power to `motor 1` of our Explorer HAT Pro.

Here is the code for the function for your reference.

```
def wheel(channel, event):  
    for i in range(5):  
        duration = randint(1,10)  
        print(duration)  
        explorerhat.motor.one.forward(100)  
        sleep(duration)  
        explorerhat.motor.one.stop()
```

Now that we have completed the function, we create a line that will handle the button press event. When the button is pressed, this event is True and the code created in the function will be executed.

```
explorerhat.touch.one.pressed(wheel)
```

With our code complete, save your work and when ready click on **Run >> Run Module** to start the project. Go ahead and press the button labelled 1 on your Explorer HAT Pro. We should see the wheel move for a random number of seconds. The exact duration will be output to the Python shell.

Your code should look like this

```
import explorerhat  
from time import sleep  
from random import randint
```

```
def wheel(channel, event):
    print("Button 1 Pressed!")
    for i in range(5):
        duration = randint(1,10)
        print(duration)
        explorerhat.motor.one.forward(100)
        sleep(duration)
        explorerhat.motor.one.stop()

explorerhat.touch.one.pressed(wheel)
```

Congratulations, you've made your own version of the Wheel of Fortune.

Extension Activity

The Explorer HAT Pro add on board also comes with four LEDs that we can easily use in our code.

To turn on a single LED, for example the red LED we type

```
explorerhat.light.red.on()
```

And to turn off

```
explorerhat.light.red.off()
```

To turn on all of the LEDs

```
explorerhat.light.on()
```

And to turn off

```
explorer.light.off()
```

You can also blink the LEDs for example the red LED.

```
explorerhat.light.red.blink(ON_TIME, OFF_TIME)
```

Adapt the function that we created earlier so that the LED are used.

Plenary

Explorer HAT Pro comes with a rich and powerful library of functionality and you can find out

more from <https://github.com/pimoroni/explorer-hat>

- How could you adapt this project for use in your school?
- Could this project be built with Scratch GPIO?

Outcome

All students

- Understand what the Explorer HAT Pro is and what it can do.
- Begin to understand physical computing and electronics.
- Used Python to create a randomly rotating wheel.