



TrafficHAT

Beginners GPIO

Blinking an LED

This worksheet will guide you through turning an LED on and off using Python & the GPIO Pins on the Raspberry Pi.

To begin load up the geany IDE by going to the Start menu --> Programming --> Geany

Next confirm you are working on a blank file by checking the tab for this program says untitled.

Begin by adding the following to the new file replacing `Your Name Here` with your name.

```
#TrafficHAT - Blink LED  
#By Your Name Here
```

These two lines are comments which only we can read. In python any line which has a `#` at the beginning is a comment and is never read or interpreted by the computer.

Next save the program either by pressing `Ctrl & S` or the save icon in geany.

Call this program `1-BasicGPIO.py`

Confirm that the program is set to save in your home folder by clicking on the house icon which has your name next to it and click save.

Your comments should now appear in the colour red.

Now we can continue on with the rest of the program. Copy out the following code.

Each line here has been commented with what the line does. You don't need to copy these as they are for you to understand what each line does.

Don't forget - Python is case sensitive!

```
```import RPi.GPIO as IO #This imports the RPi.GPIO Library as IO
```

```
from time import sleep # This imports only the sleep function from time.
```

# Next we need to setup the Raspberry Pi's GPIO Pins

---

```
IO.setmode(IO.BCM) #This sets the Pi to BCM / GPIO Numbering
```

## Next we want to setup the Green LED as an output

---

```
green = 22
```

```
IO.setup(green,IO.OUT) # This sets the green LED to be an output.
```

## By using the try loop we can then add an except at the end

---

## to figure out what the bugs are in our code.

---

try:

```
#In python we indent one for code to be inside a block.
```

```
#All code with 1 indent will be inside the try block
```

```
#To do an indent you can either press space 4 times
```

```
#Or press the Tab button once which will insert 4 spaces.
```

```
#You can do either but you can only do one or the other
```

```
#in the same program.
```

```
#Geany also usually does this automatically for you
```

while True:

```
#Code in the while loop will repeat forever.
```

```
#As you can see we're using 2 indents for this.
```

```
#To begin we're going to turn the LED On
```

```
#The RPi.GPIO Library accepts a binary input to decide if it is on or off.
```

```
#Binary is either 1 for On / True or 0 for Off / False
```

```
#To control an output we do the following
```

```
IO.output(green,1)
```

```
#This will turn the LED On
```

except KeyboardInterrupt:

#While this isn't actually an error python sees this as an exception.

#This code is ran when we quit the program using the keyboard.

#We want to clean up all of the GPIO Pins by running  
IO.cleanup()

except:

#If there are any other errors we want to know what they are.

print "Error:", sys.exc\_info()[0]

raise``

Now you can press F5 to run your program.

But there's an error!

Using GPIO pins on the Raspberry Pi requires us to run the python program as root. To make it so our program launches as root we need to make a change to a setting in geany.

Start by clicking Build at the top toolbar at the top and then Set Build Commands and a box should pop up like the following.

We want to change the execute command to use sudo python instead.

Add sudo to the beginning and your box should now look like this.

Now try re-running the program. The green LED should turn on.

To close your program press Ctrl & C into the terminal window. This will close the program and reset the GPIO pins so the green should go off.

And that is it for worksheet 1!