Morse Code Lab 12

Materials needed

- 5x male to female jumper wires
- Breadboard/Raspberry Pi
- An LED (of any kind)
  - I used a Keyestudio Stoplight
- A touch sensor
  - I used a Keyestudio touch sensor

Code (Look at #'s for explanation)

```
#Import the Time and GPIO libraries in order to use the libraries' code
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
#Sets the script mode to input/output
GPIO.setwarnings(False)
#Turns off warnings
#Continually loops the while statement until a KeyboardInterrupt occurs
x = True
while x==True:

#S
#I used GPIO pin #26 to turn the LED on/off
        GPIO.setup(26,GPIO.OUT)
#sets up the GPIO pin output
        print ("LED on")
#prints "LED on" to the terminal so I know the code is working
        GPIO.output(26,GPIO.HIGH)
        time.sleep(1)
#time.sleep() keeps the light at its current state (on/off) for however many seconds... 1 second in this line
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(1)
#You don't need this setup line of code every time you want to turn the LED on/off
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(1)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(1)
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(1)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(3)

#O
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(2)
#The amount of time is changed
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(1)
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(2)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(1)
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(2)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(3)

#S
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(1)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(1)
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(1)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(1)
        GPIO.setup(26,GPIO.OUT)
        print ("LED on")
        GPIO.output(26,GPIO.HIGH)
        time.sleep(1)
        print ("LED off")
        GPIO.output(26,GPIO.LOW)
        time.sleep(3)
```

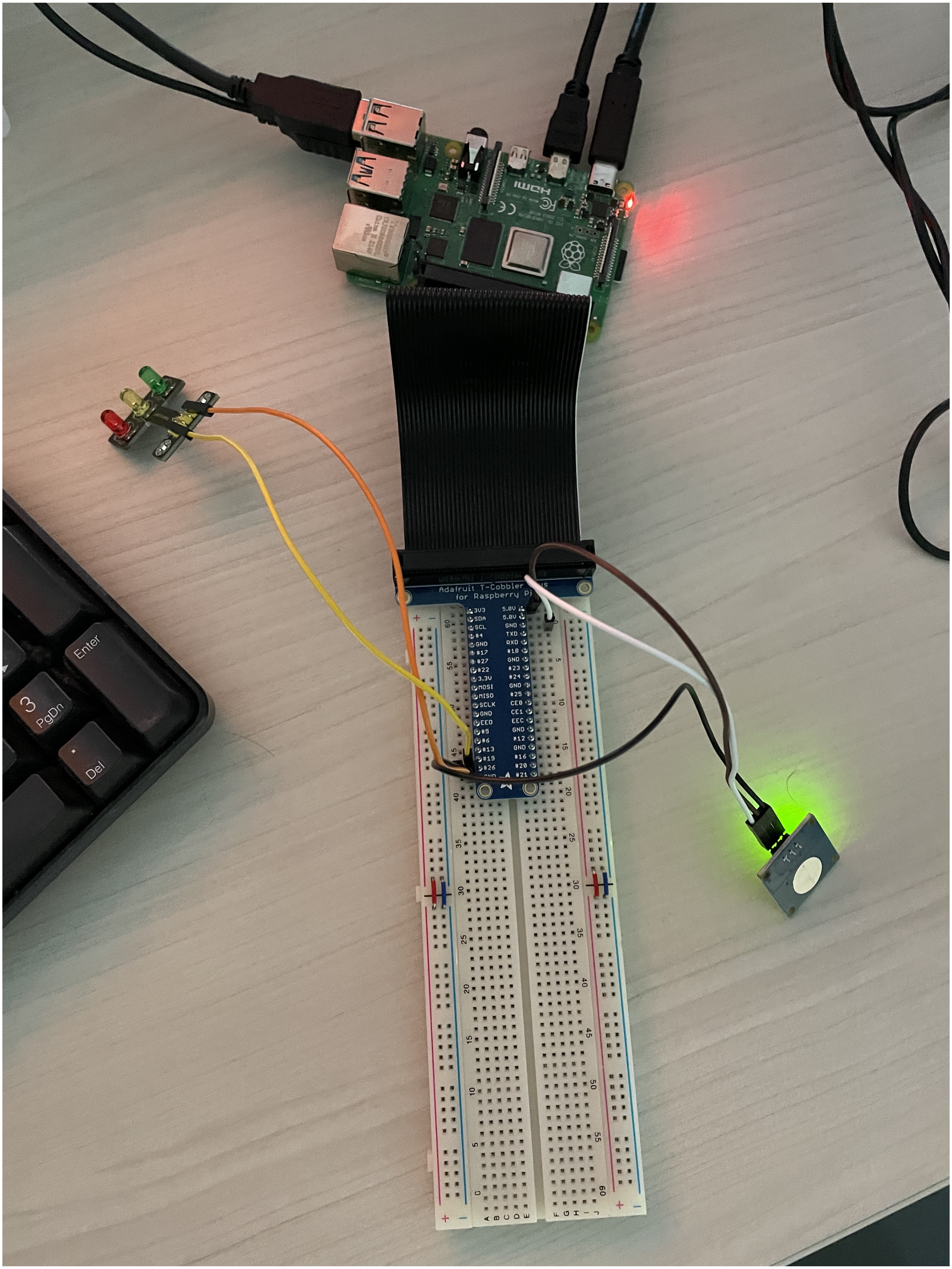
Running the Code

To run the code, open Visual Studio and make a new Python file. You may have to install Python3 through a VS Code extension (Documentation about installing Python extensions *https://code.visualstudio.com/docs/python/python-tutorial*).

Then, install the Raspberry Pi GPIO module using pip (`pip install RPi.GPIO`). You don't need to pip install the time module because it's already installed with Python.

Lastly, click the "run" button or run it through a VS Code directory in the terminal (`/usr/local/bin/python3 /Users/username/VSCode/morse.py`). The username would be the username of whichever user is logged into the computer.

Circuit


HDMI
Adafruit T-Cobbler Plus
for Raspberry Pi
3V3
SDA
SCL
#4
GND
#17
#27
#22
3.3V
MOSI
MISO
SCLK
GND
EED
#5
#6
#13
#19
#26
5.0V
5.0V
GND
TXD
RXD
#18
GND
#23
#24
GND
#25
CE0
CE1
EEC
GND
#12
GND
#16
#20
#21
Enter
3
PgDn
Del

LED

For a regular LED, the short leg (anode, -) connects to a GND port on breadboard using a male-female jumper wire (or connected directly to the breadboard with a resistor, which is safer. In that case, read this documentation to circuit: *https://thepihut.com/blogs/raspberry-pi-tutorials/27968772-turning-on-an-led-with-your-raspberry-pis-gpio-pins*. Be aware of what GPIO pins you're using!). The long leg (cathode, +) connects to GPIO port #26.

For the Keyestudio Stoplight, connect R to port #26 and GND to the GND port.

Touch Sensor

For the touch sensor, connect SIS to GPIO port. Then, connect VCC to 5.0 V (this is your voltage). Lastly, connect GND to the GND port (same GND as the one used for the LED).