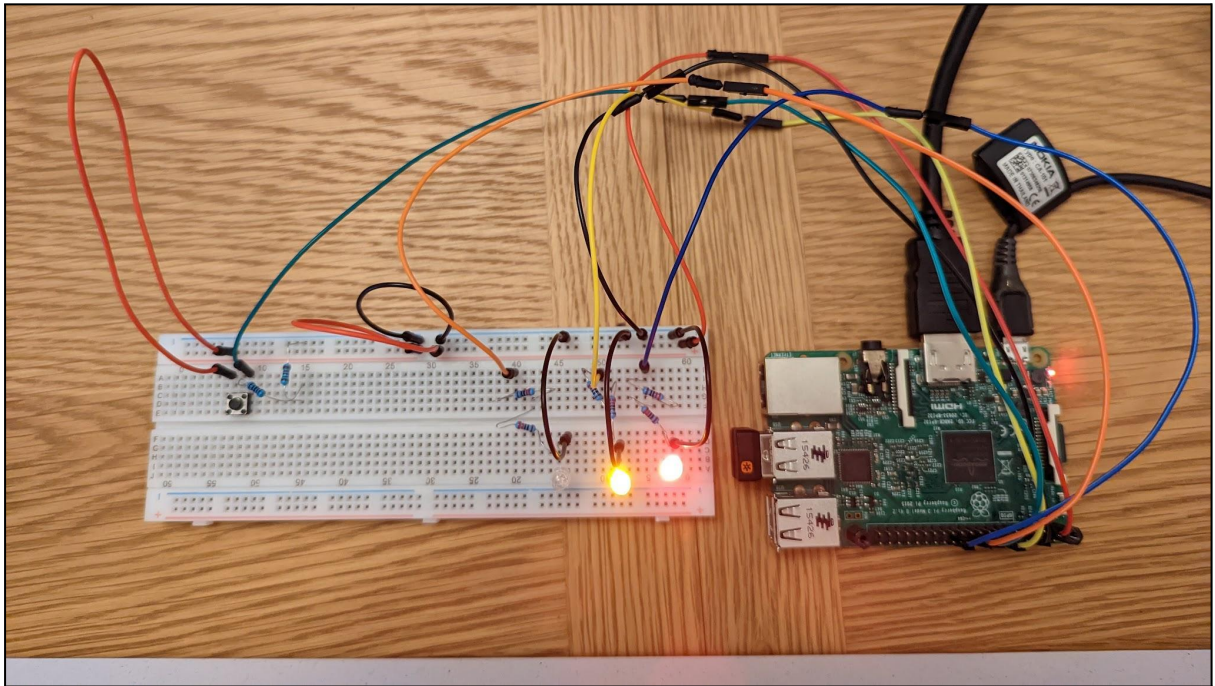
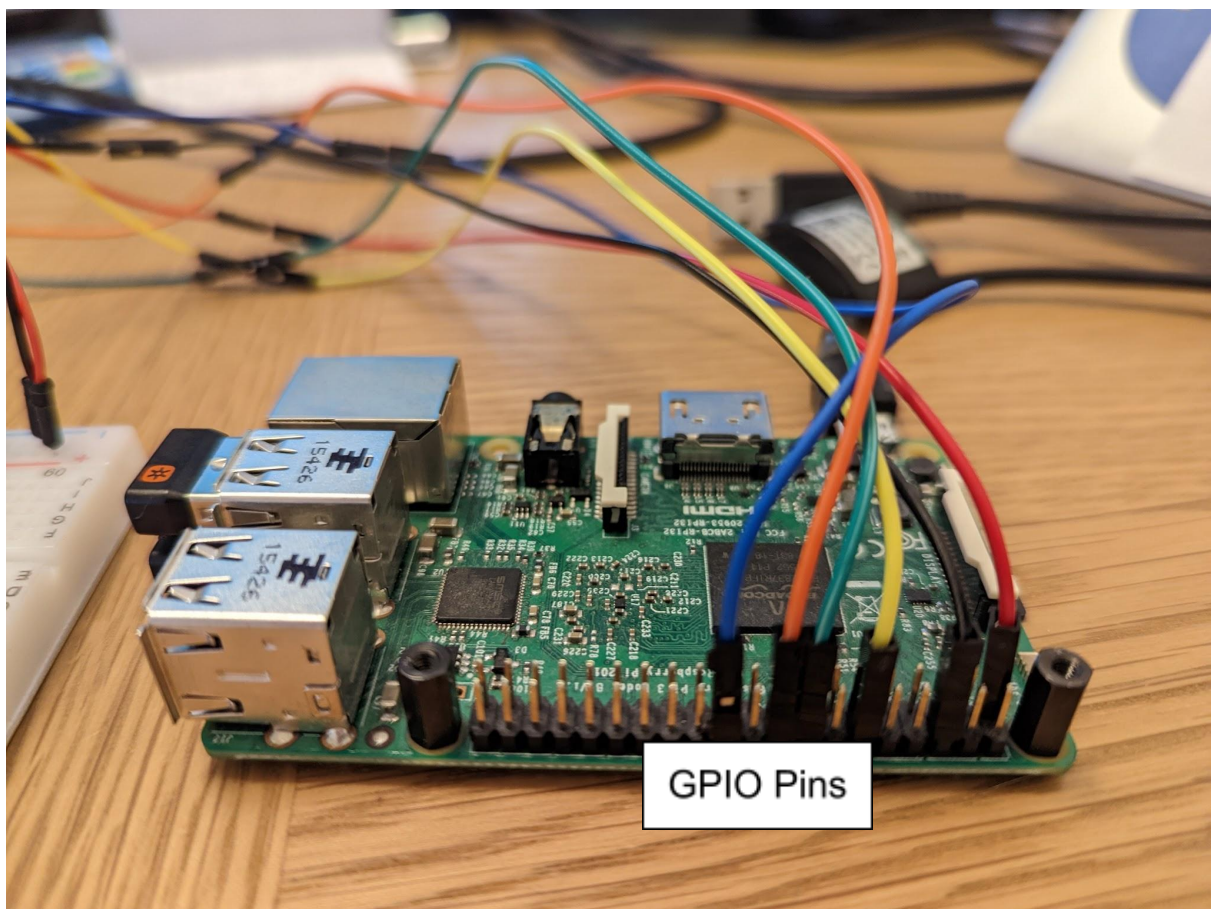


Traffic Light Circuit

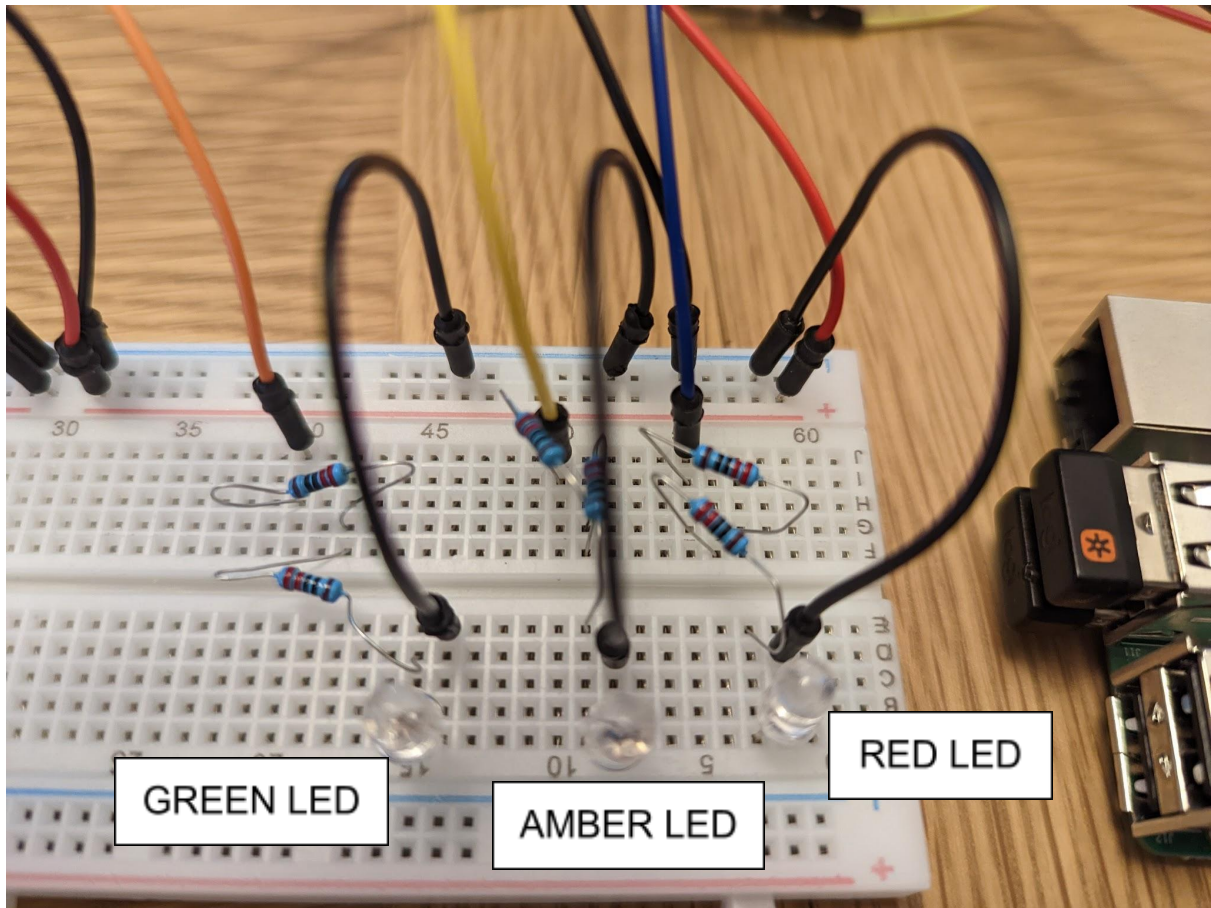


Overall Circuit Connections

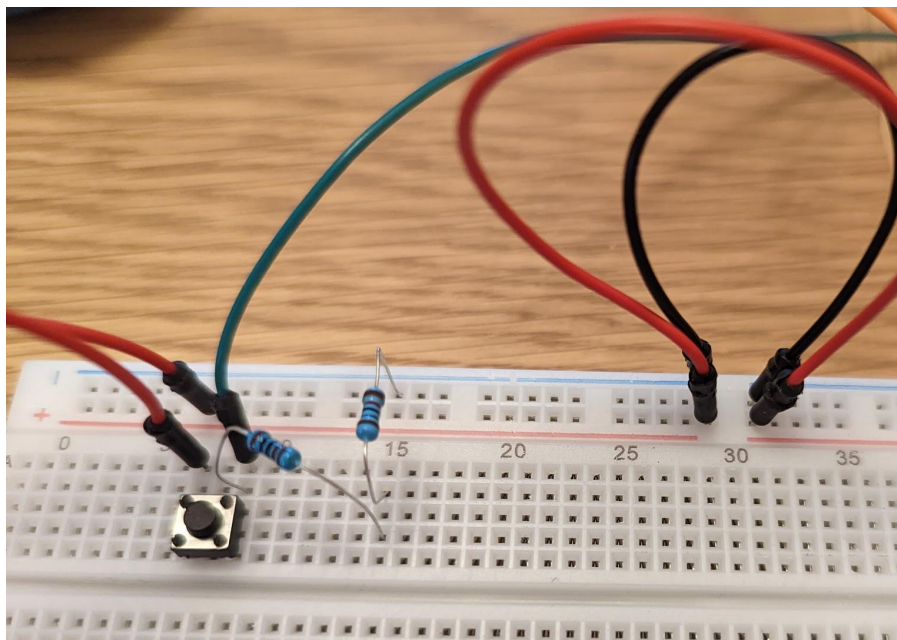
Wiring Key: **Red** +3.3v **Black** GND **Blue** RED LED **Yellow** AMBER LED **Green** GREEN LED



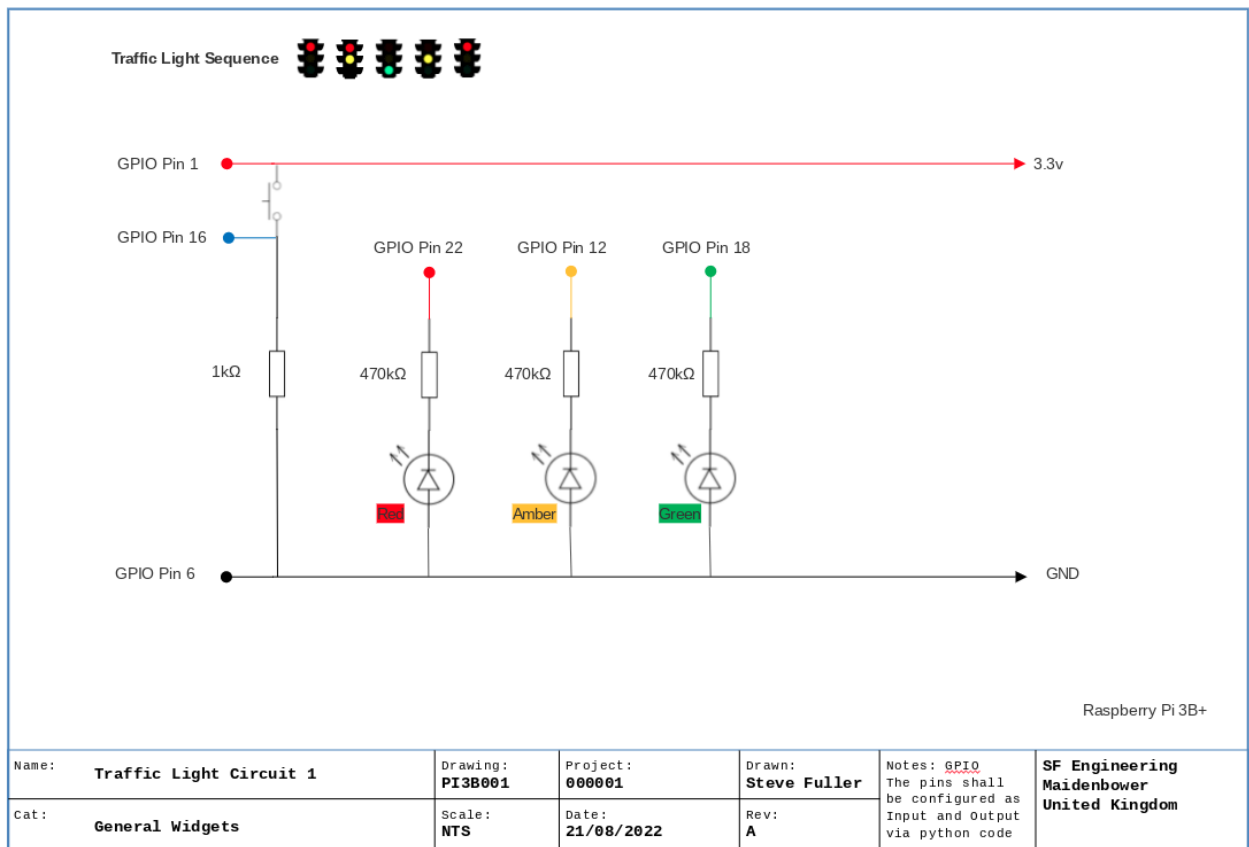
Raspberry Pi GPIO Connections



LED Wiring Connections



Push Button Wiring



Circuit Diagram

Link to circuit diagram [here](#)

The Python 2.7 code © SGF 2022 Approved

```
#####
# Traffic_Light.py
# Code sequences through traffic light
# until the push button is pressed
#####

# import libraries
import RPi.GPIO as GPIO
from time import sleep

# set constants for GPIO pins
# 3.3v Pin 1 GND Pin 6
push_button = 16
red_LED = 22
amber_LED = 12
green_LED = 18

# time delays for traffic lights. Get user entry. Entering nothing will use the defaults
stay_on_red = int(raw_input("Enter the stop time in seconds [10]: "), or "10")
stay_on_redamber = int(raw_input("Enter the get ready time in seconds [3]: "), or "3")
```

```

stay_on_green = int(raw_input("Enter the go time in seconds [10]: "), or "10")
stay_on_amber = int(raw_input("Enter the ready to stop time in seconds [3]: "), or "3")

# other times
start_time = 3
quit_prompt_time = 3

# set up GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
# set inputs
GPIO.setup(push_button,GPIO.IN)
# set outputs
GPIO.setup(red_LED,GPIO.OUT)
GPIO.setup(amber_LED,GPIO.OUT)
GPIO.setup(green_LED,GPIO.OUT)

# function to individually switch each LED
# red, amber and green are passed as Boolean, value sleep_time passed as int as seconds
def switch_LEDS(red, amber, green, sleep_time):

    GPIO.output(red_LED, red)
    GPIO.output(amber_LED, amber)
    GPIO.output(green_LED, green)

    sleep(sleep_time)

# end of function

#####
# Program starts here
#####
print "To stop program push and hold button until code quits"
txt = "Checking all LED's for {} seconds"
print (txt.format(start_time))
print ""

switch_LEDS(True, True, True, start_time) # set LEDS and wait

print "Starting Traffic Light Sequence"
print ""

# Start Sequence
while GPIO.input(push_button) == 0: #if button not pressed

    # Red LED
    txt = "Red LED on for {} seconds"
    print (txt.format(stay_on_red))

    switch_LEDS(True, False, False, stay_on_red)

```

```

if GPIO.input(push_button) == 1: # quit
    break
# Red End

# Red and Amber LED
txt = "Red and Amber LED's on for {} seconds"
print (txt.format(stay_on_redamber))

switch_LEDS(True, True, False, stay_on_amber)

if GPIO.input(push_button) == 1: # quit
    break
# Red Amber End

# Green LED
txt = "Green LED on for {} seconds"
print (txt.format(stay_on_green))

switch_LEDS(False, False, True, stay_on_green)

if GPIO.input(push_button) == 1: # quit
    break
# Green End

# Amber LED
txt = "Amber LED on for {} seconds"
print (txt.format(stay_on_amber))
print ""
switch_LEDS(False, True, False, stay_on_amber)

if GPIO.input(push_button) == 1: # quit
    break
# Amber End

# if not a break then the While loop starts again

# Code restarts here when While loop breaks
print ""

print "Push button pressed to quit"
print "Resetting LED's"
txt = "Quitting in {} seconds"
print (txt.format(quit_prompt_time))

print ""

# turn on all LEDS momentarily to show push button pressed
switch_LEDS(True, True, True, quit_prompt_time)

GPIO.cleanup()
#####

```

Program Ends here

#####