Exercise in Measuring Reaction Time.

We make 2 examples, with different layouts each capable of comparing reaction times of different participants.

Example 1

Press the button when the light goes on.

Parts

1 x led

1 x.220 ohm resistor

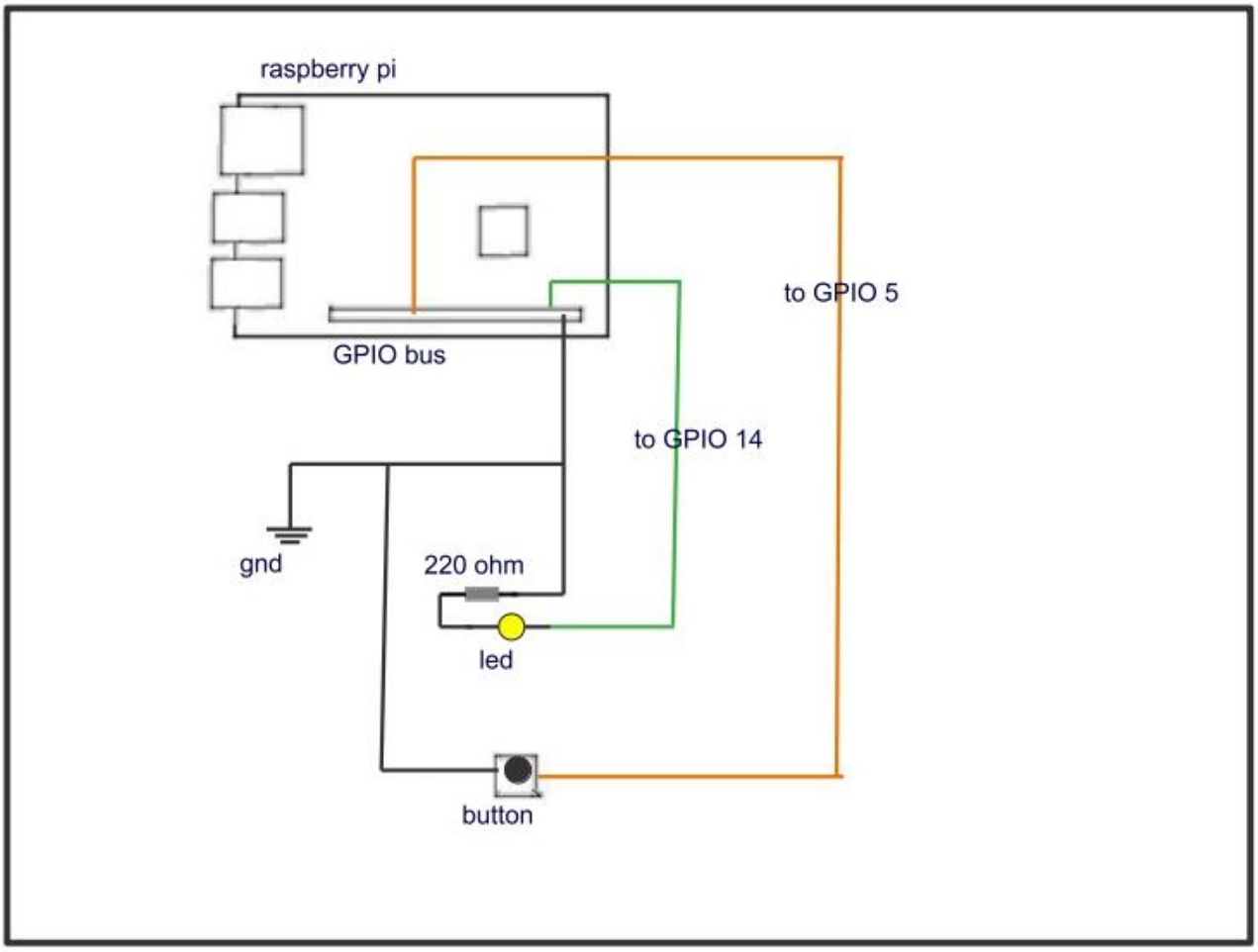
3 x Dupont male to female wires

1 x microswitch button

1 x breadboard wire

1 x breadboard

Schematic



Above details the connections of our circuit. Basically a random number is used to delay the time the led lights and the timer works out how long it takes the button to be pressed and reports it to the console. Very simple, but it is also the basics of our next example.

Caution:
when placing the button on the breadboard take care that the orientation is correct otherwise it will not function as a switch.

The program below associated with the circuit will perform the task, it may be copied into a specific directory and run from the same directory by the instruction ;
sudo python reactiontm.py
Alternatively it may be run from a Python IDE such as Thonny or Mu.

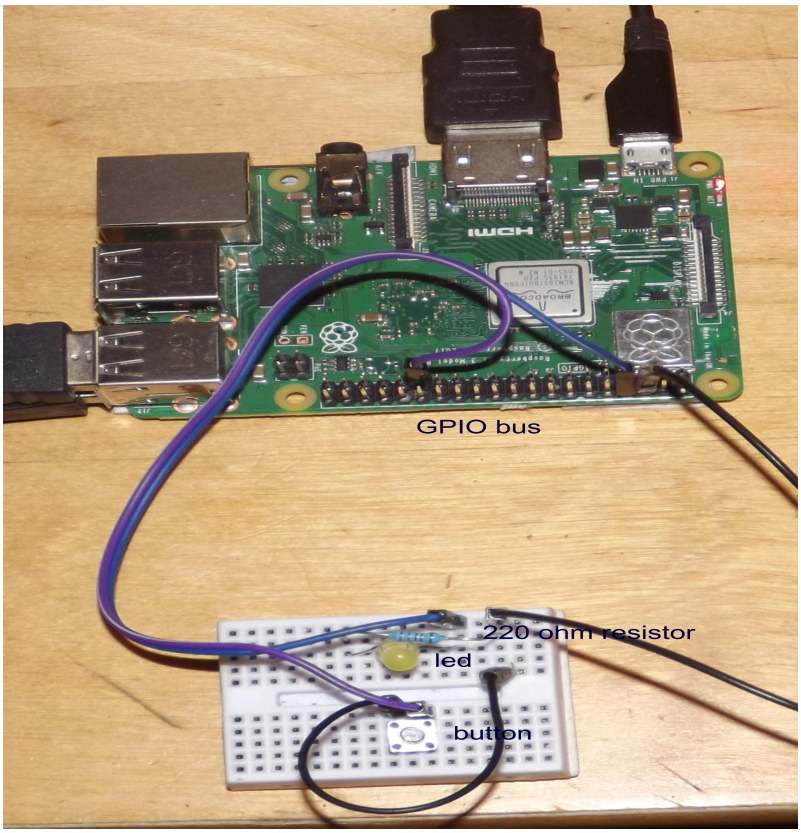
Reactiontm.py

```
“””
 reactiontm.py : to determine the lag time between the led lighting up and
 when button is pressed.
Tunde Adeyemo 01/21
License: Public Domain
“””
import time
```

```
import digitalio
import board
import random

led = digitalio.DigitalInOut(board.D14)
led.direction = digitalio.Direction.OUTPUT
button = digitalio.DigitalInOut(board.D5)
button.direction = digitalio.Direction.INPUT
button.pull = digitalio.Pull.UP


led.value = False  # Start with led off
time.sleep(random.random()*10)
t0 = time.monotonic()
led.value = True
while led.value == True:
    if button.value == False:
        t = time.monotonic() - t0
        print("reaction time =")
        print("%.5f " %(t))
        led.value = False
```

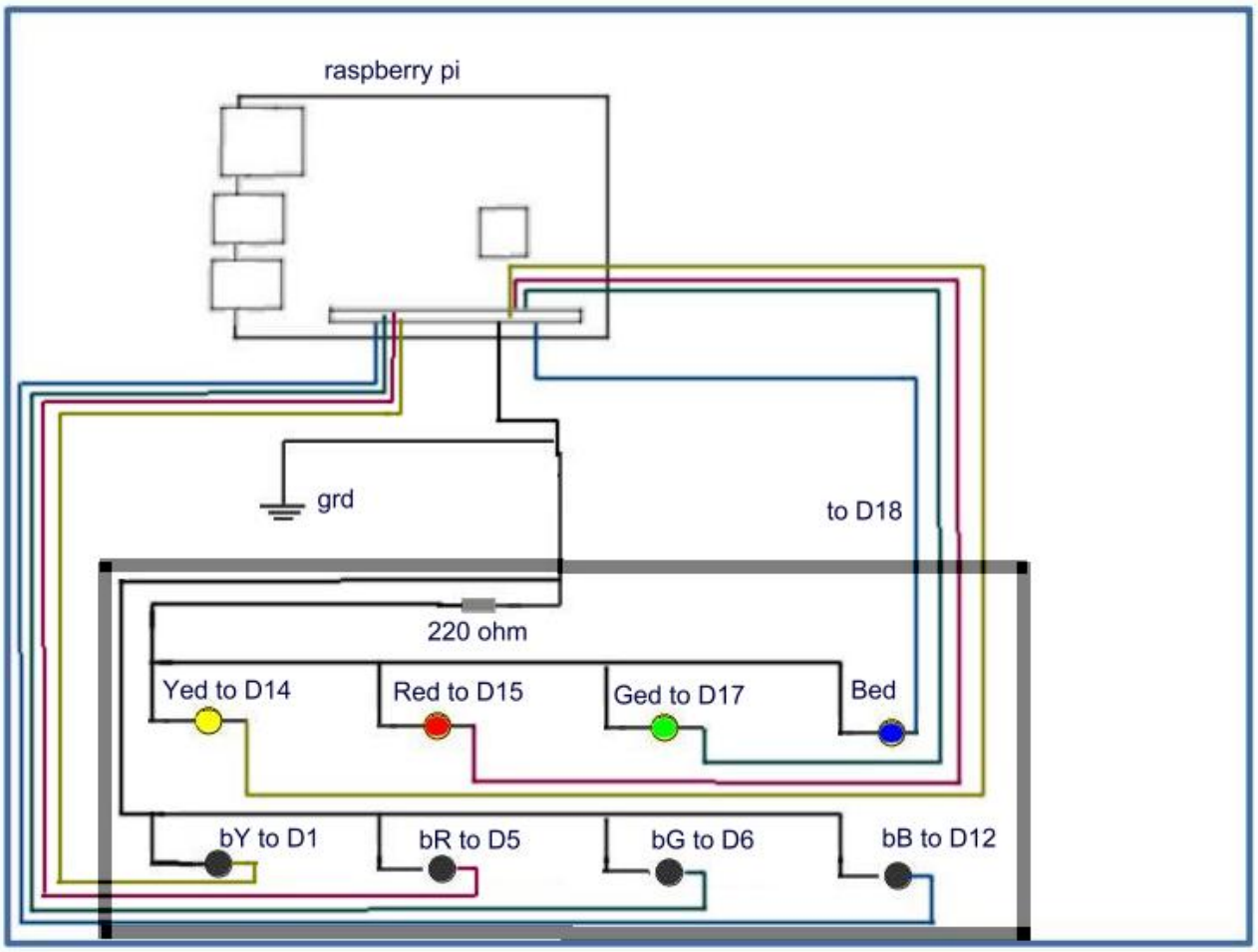


Now for the big kahuna. What about if we do this with 4 different colour lights each with it's own button and repeat the trial 10 times to get an average and fastest reaction. Surely this would be a worthwhile exercise resulting in measurable difference in individuals and or circumstance.

Example 2

Follow the Colour

Schematic



Parts

4 x led

1 x.220 ohm resistor

9 x Dupont male to female wires

4 x microswitch button

9 x breadboard wire

1 x breadboard

the function of this apparatus is much the same of the previous example. After a random time duration a random selection from 4 colours is made and the led associated is lit. The time to when it's associated button pressed is recorded. This is repeated 10 times. The console then shows the average time of reaction and the fastest time.

The program YRGB.py that performs the task is shown below. It may be copied and run from a terminal or an IDE such as Thonny or Mu.

YRGB.py

```
""" YRGB.py is derived from reaction.py :
 it performs a test of the type in reaction.py for 10 times
 it then relays the result as the average and the best
 Tunde Adeyemo 01/21
 License: Public Domain """

import time
import digitalio
import board
import random

light=[]
Yed = digitalio.DigitalInOut(board.D14)
light.append(Yed)
Red = digitalio.DigitalInOut(board.D15)
light.append(Red)
Ged = digitalio.DigitalInOut(board.D17)
light.append(Ged)
Bed = digitalio.DigitalInOut(board.D18)
light.append(Bed)
Yed.direction = digitalio.Direction.OUTPUT
Red.direction = digitalio.Direction.OUTPUT
Ged.direction = digitalio.Direction.OUTPUT
Bed.direction = digitalio.Direction.OUTPUT

button=[]
bY = digitalio.DigitalInOut(board.D1)
```

```
button.append(bY)
bR = digitalio.DigitalInOut(board.D5)
button.append(bR)
bG = digitalio.DigitalInOut(board.D6)
button.append(bG)
bB = digitalio.DigitalInOut(board.D12)
button.append(bB)
bY.direction = digitalio.Direction.INPUT
bY.pull = digitalio.Pull.UP
bR.direction = digitalio.Direction.INPUT
bR.pull = digitalio.Pull.UP
bG.direction = digitalio.Direction.INPUT
bG.pull = digitalio.Pull.UP
bB.direction = digitalio.Direction.INPUT
bB.pull = digitalio.Pull.UP
stt =[]

for i in range (10):

    Yed.value = False  # Start with led off
    Red.value = False
    Ged.value = False
    Bed.value = False
    time.sleep(random.random()*10)

    col = random.random()
    col *= 4.0
    #print (col)
    if col <1.0 :
        indx = 0
        clr = "yellow"
    elif col <2.0 and col >=1.0 :
```

```
        indx = 1
        clr = "red"
    elif col <3.0 and col >=2.0 :
        indx = 2
        clr = "green"
    elif 3.0<= col  :
        indx = 3
        clr = "blue"



    light[indx].value = True
    #print(indx,button[indx].value)

    t0 = time.monotonic()
    while light[indx].value == True:
        if button[indx].value == False:
            t = time.monotonic() - t0
            print("reaction time =")
            print("%.5f %s %i" %(t,clr,i+1))
            print("")
            light[indx].value = False
            stt.append(t)
stm = stt[0]
sts= 0.0
for i in stt:
    #print(i)
    if i <stm :
        stm = i
    sts+=i

print("your average reaction time is")
print("%.5f " %(sts/10))
```

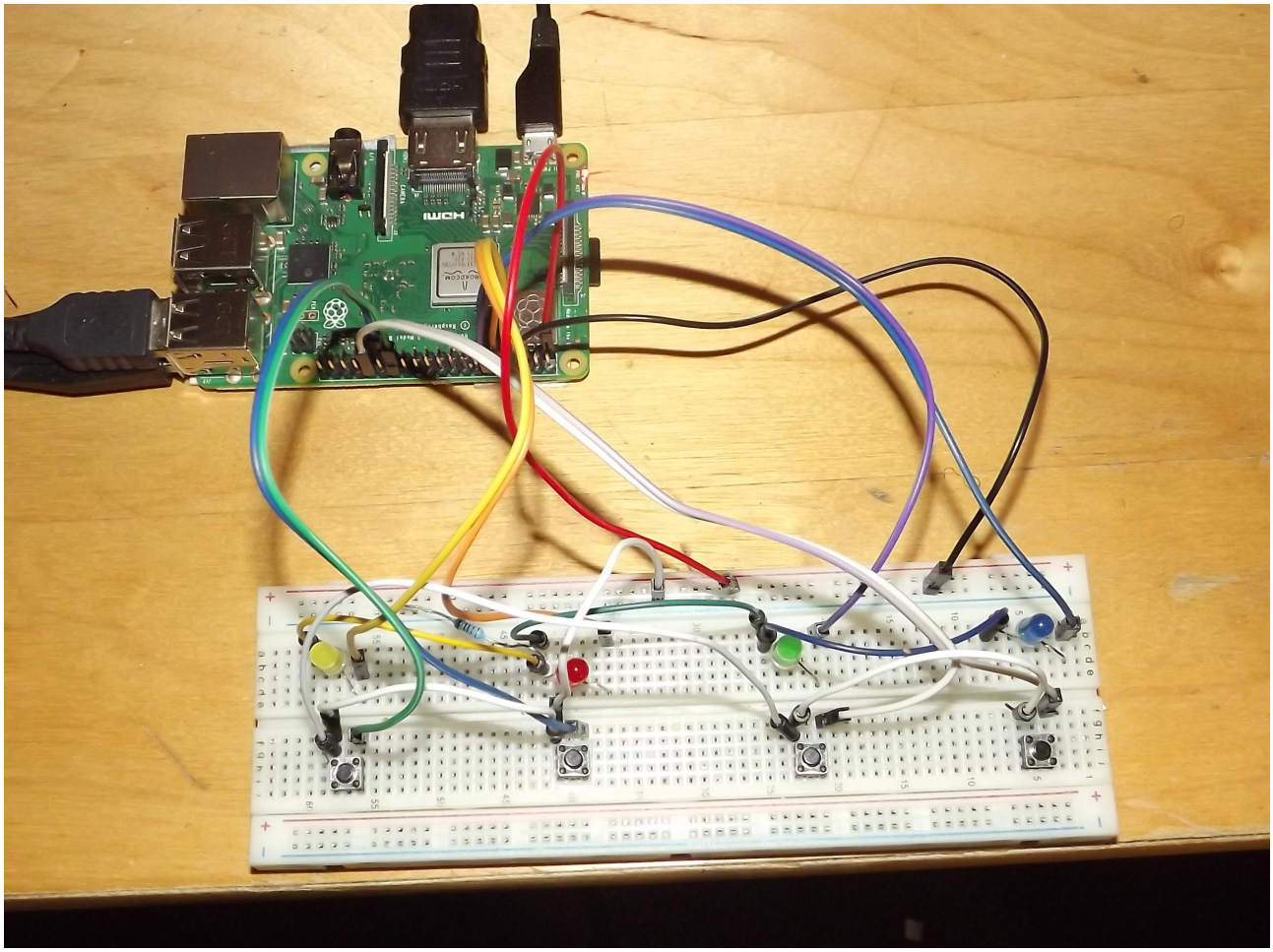
print("best reaction time is")

print("%.5f " %(stm))

Observations:

The buttons are delicate and intricate and require care when being pressed otherwise they become displaced but this prototype can demonstrate the practicability of a hardwired permanent circuit.

You can do repeated trials to observe improvements due to improved attention, concentration and technique. It can be quite addictive. Try changing the time factor from 10 to 2 in the line
“ time.sleep(random.random()*10)”



references.

The Python Language Reference

Release 3.7.4

Guido van Rossum and the Python development team

Learning to Program Using Python

Cody Jackson

Measurement of Rotation Rate of a Small Motor using a Raspberry Pi

Tunde Adeyemo