Tampines Regional Library
LearnX Community
Pi Python Introductory Course
Course Material
By
Goh Soon Seng

PLEASE DO NOT WRITE ANYTHING ON THE PAGES
OF THIS BOOKLET. THANK YOU.

Programme

Python Lesson – Session # 1

LIBRARIES

- Machine
- picozero
- time
- tm1637

VARIABLES

LOOPS

- while True:

PYTHON RULE

- Indentation
- Character casing

Python Lesson – Session # 2

FUNCTIONS

CONDITIONAL STATEMENTS

- **if else**

LOOPS

- while True:
- for

MESSAGING

- how humans and machine communicate with each other

Python Lesson – Session # 3

TROUBLESHOOTING

MORE PYTHON CODING

PASSWORD

INPUT STATEMENT

Python Lesson – Session # 4

REVISION & TEST

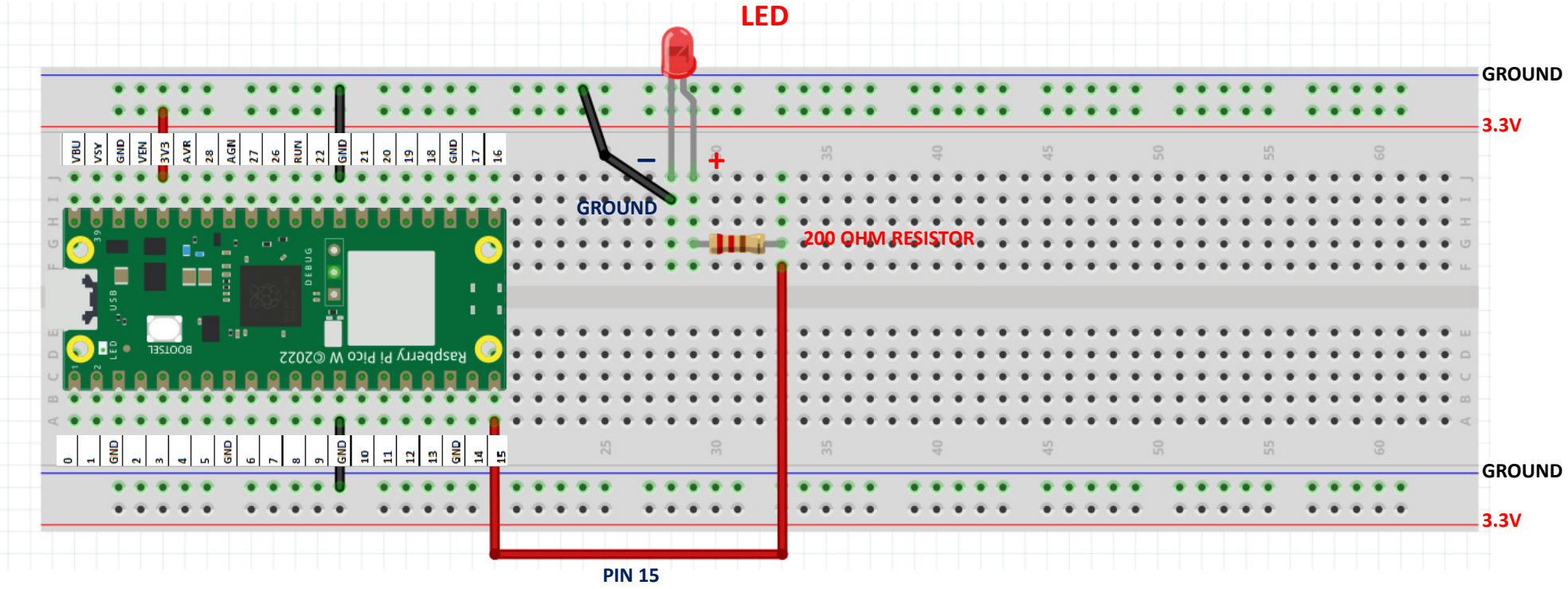
- programming the lights and siren of a patrol car

INTRODUCTION TO CHATGPT

- How to login to ChatGPT
- Getting ChatGPT to help us code
- Learning using ChatGPT

EXERCISE #1 Wiring a Light Emitting Diode (LED)

Please wire up the following circuit



TESTING OUR CIRCUIT USING THONNY SHELL

Ex 1a. Turning our Red LED on

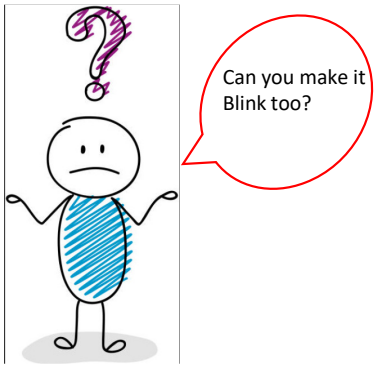
```
>>> from picozero import LED
>>> red_led = LED(15)
```

```
>>> red_led.on()
```

Ex 1b. Turning it off

```
>>> red_led.off()
```

Ex 1c. Blinking the red LED



ALGORITHM TO CODE

ALGORITHM

Start with red led off
Wait 1 second
Turn red led on
Wait 1 second
Turn red led off
Wait 1 second
Turn red led on
Wait 1 second
Turn red led off

CODE

<untitled> *

```
1  from picozero import LED
2  from time import sleep
3
4  red_led=LED(15)
5
6  red_led.off()
7  sleep(1)
8  red_led.on()
9  sleep(1)
10 red_led.off()
11 sleep(1)
12 red_led.on()
13 sleep(1)
14 red_led.off()
15
```

RUN and save this as ex1c.py

while True:

ex1c.py

```
1  from picozero import LED
2  from time import sleep
3  red_led = LED(15)
4
5  red_led.off()
6  sleep(1)
7  red_led.on()
8  sleep(1)
9  red_led.off()
10 sleep(1)
11 red_led.on()
12 sleep(1)
13 red_led.off()
```

Convert

```
1  from picozero import LED
2  from time import sleep
3  red_led = LED(15)
4  while True:
5      red_led.off()
6      sleep(1)
7      red_led.on()
8      sleep(1)
9      red_led.off()
10     sleep(1)
11     red_led.on()
12     sleep(1)
13     red_led.off()
14
```

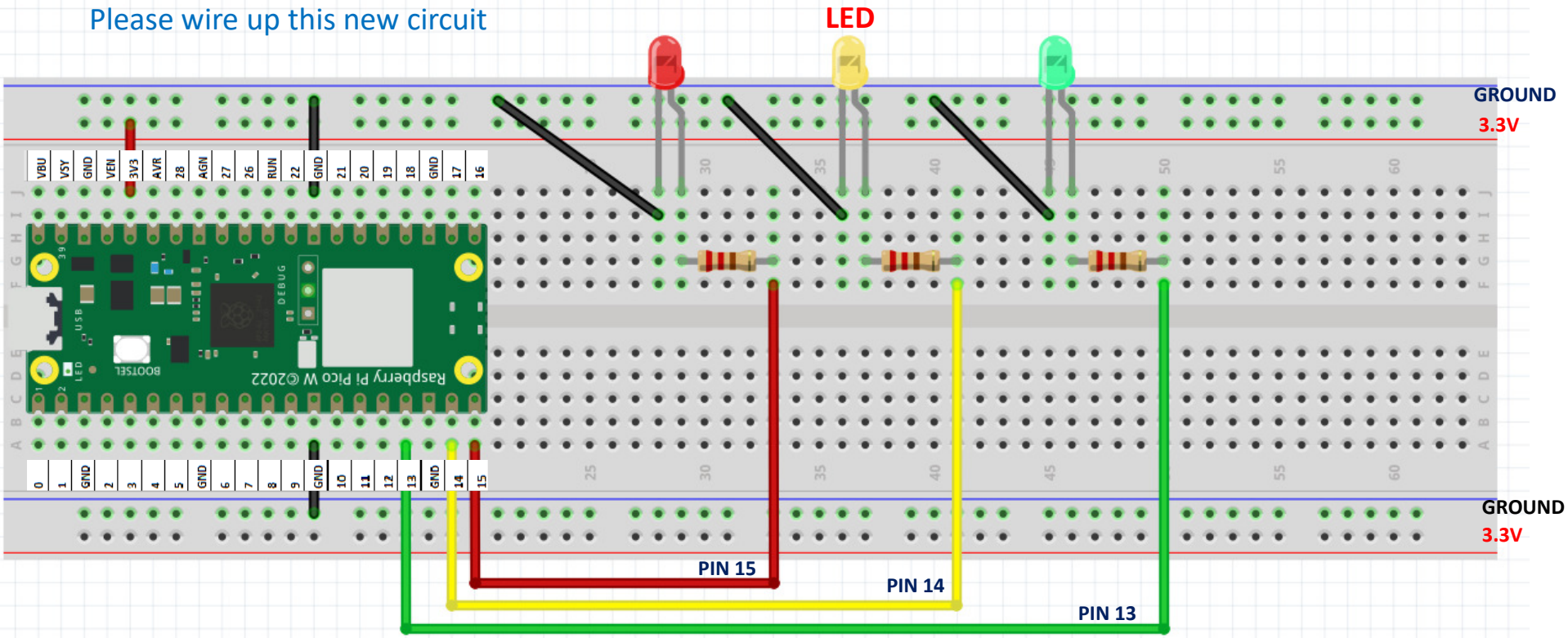
Runs once

Runs forever

Run this new program
Save it as ex1d.py

EXERCISE 2 – TRAFFIC LIGHT

Please wire up this new circuit

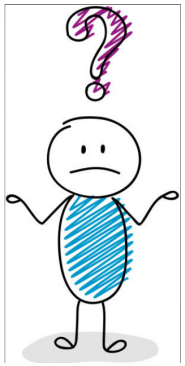


EXERCISE 2 – CONTROLLING MULTIPLE LEDs

Ex 2a. Test the Circuit. Turn each of the LEDs ON and OFF

```
>>> from picozero import LED
>>> red_led= LED(15)
>>> yellow_led = LED(14)
>>> green_led=LED(13)
```

Ex 2b. Turning the circuit into a Traffic Light System



How?

How does a traffic light work?

Green Light On
Give cars 10 seconds to cross
Green Light Off
Yellow Light On
Give 5 seconds warning
Yellow Light Off
Red Light On
Wait 10 seconds
Red Light Off

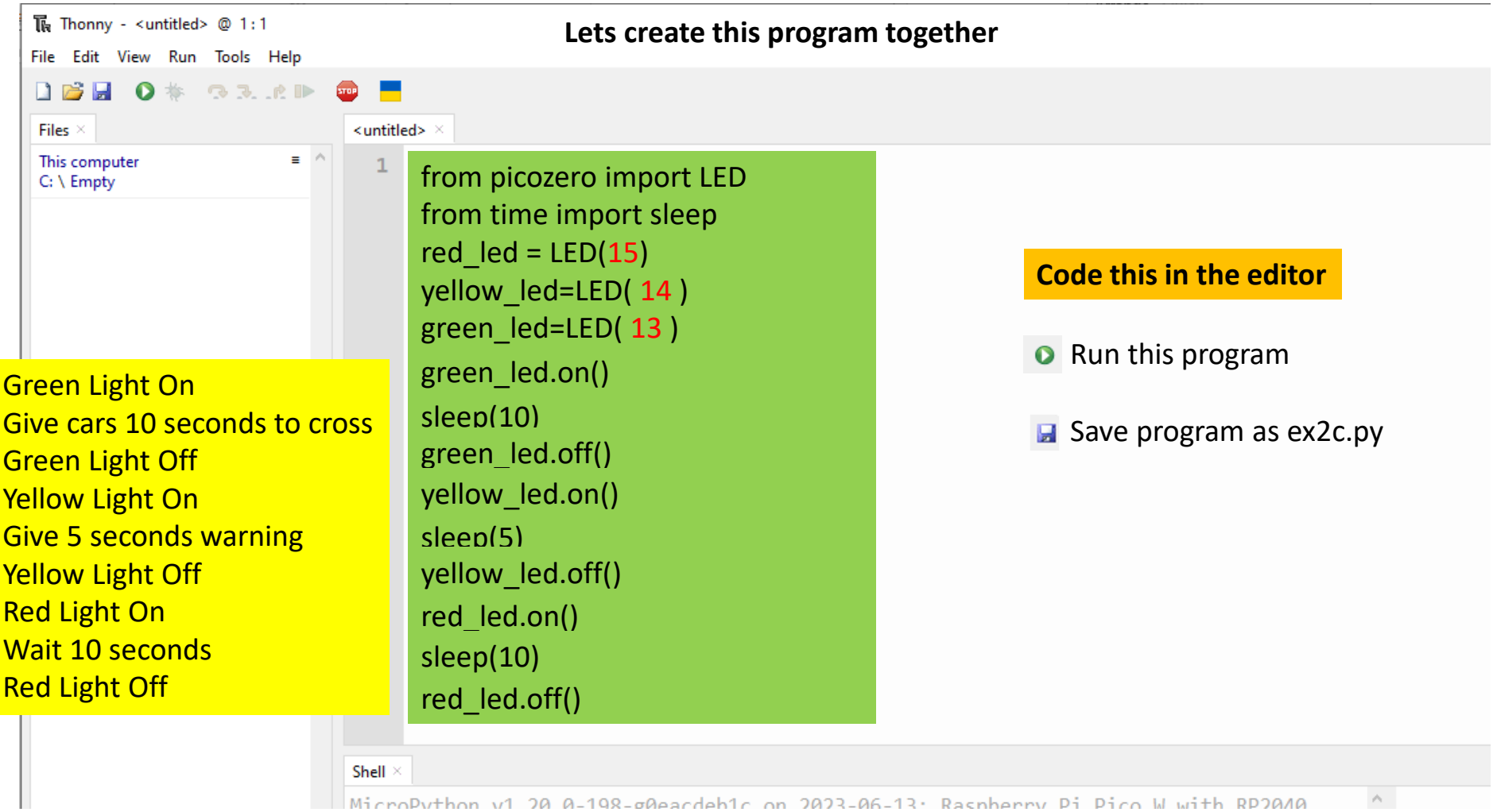
Algorithm

Code

```
green_led.on()
sleep(10)
green_led.off()
yellow_led.on()
sleep(5)
yellow_led.off()
red_led.on()
sleep(10)
red_led.off()
```

Code this in the editor
Run and save it as ex2b.py

Lets create this program together



Green Light On
Give cars 10 seconds to cross
Green Light Off
Yellow Light On
Give 5 seconds warning
Yellow Light Off
Red Light On
Wait 10 seconds
Red Light Off

Code this in the editor

Run this program

Save program as ex2c.py

ex2c.py

```
1  from picozero import LED
2  from time import sleep
3  red_led = LED(15)
4  yellow_led=LED(14)
5  green_led=LED(13)
6
7  green_led.on()
8  sleep(10)
9  green_led.off()
10 yellow_led.on()
11 sleep(5)
12 yellow_led.off()
13 red_led.on()
14 sleep(10)
15 red_led.off()
```

Runs once

convert

Runs forever

```
1  from picozero import LED
2  from time import sleep
3  red_led = LED(15)
4  yellow_led=LED(14)
5  green_led=LED(13)
6  while True:
7      green_led.on()
8      sleep(10)
9      green_led.off()
10     yellow_led.on()
11     sleep(5)
12     yellow_led.off()
13     red_led.on()
14     sleep(10)
15     red_led.off()
16
```

Run the Program
Save it as ex2d.py

Python Lesson – Session # 2

FUNCTIONS

CONDITIONAL STATEMENTS

- **if else**

LOOPS

- while True:
- for

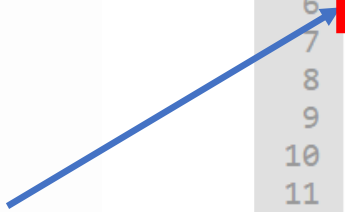
MESSAGING

- how humans and machine communicate with each other

FUNCTION – A SET OF CODES WITH A NAME

ex2d.py

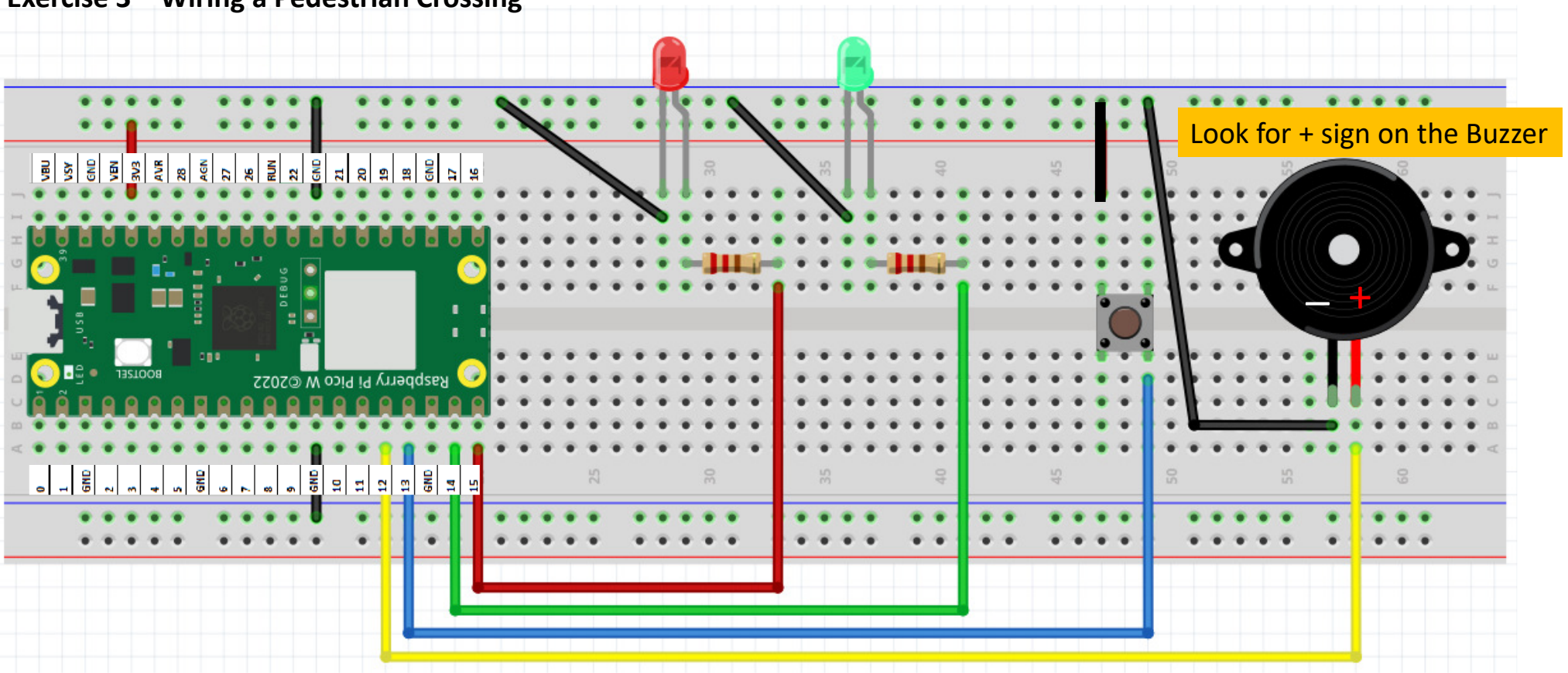
```
1  from picozero import LED
2  from time import sleep
3  red_led = LED(15)
4  yellow_led=LED(14)
5  green_led=LED(13)
6  while True:
7      green_led.on()
8      sleep(10)
9      green_led.off()
10     yellow_led.on()
11     sleep(5)
12     yellow_led.off()
13     red_led.on()
14     sleep(10)
15     red_led.off()
16
```



```
1  from picozero import LED
2  from time import sleep
3  red_led = LED(15)
4  yellow_led=LED(14)
5  green_led=LED(13)
6  def trafficlight():
7      green_led.on()
8      sleep(10)
9      green_led.off()
10     yellow_led.on()
11     sleep(5)
12     yellow_led.off()
13     red_led.on()
14     sleep(10)
15     red_led.off()
16
17 while True:
18     trafficlight()
19
```

Run the Program
Save it as ex2e.py

Exercise 3 – Wiring a Pedestrian Crossing



TESTING OUR CIRCUIT USING THONNY SHELL

Ex 3a. Turning our Buzzer on

```
>>> from picozero import Buzzer
>>> buzzer= Buzzer(12)
>>> buzzer.on()
```

Ex 3b. Turning it off

```
>>> buzzer.off()
```

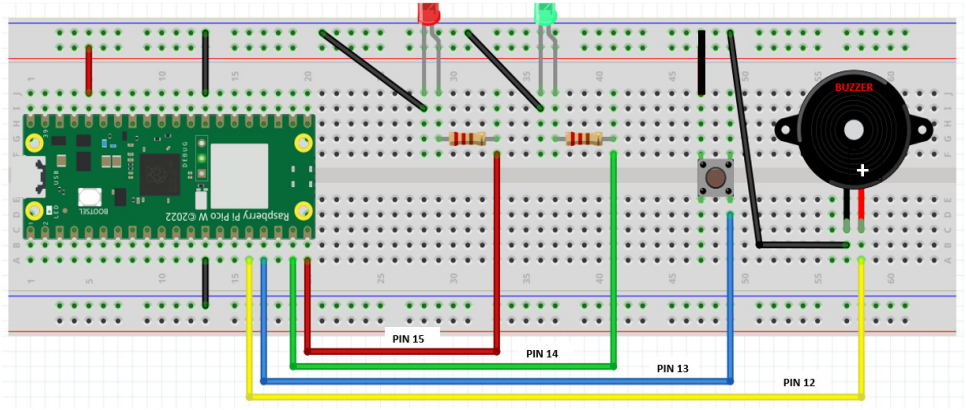
Ex 3c. Setting up multiple sensors : our circuit has Buzzer, Button, LED

```
>>> from picozero import Buzzer, Button, LED
>>> buzzer= Buzzer(12)
>>> button=Button(13)
>>> red_led=LED(15)
>>> button.when_pressed = red_led.on
>>> button.when_released = red_led.off

>>> button.when_pressed = buzzer.on
>>> button.when_released = buzzer.off
```

Let's say we want the LED
To light up when we press the button
And off when we release the button

Exercise 3 – Turning the circuit into a pedestrian crossing with code



HOW DOES A PEDESTRAIN CROSSING WORK?

RED LED IS ON / GREEN LED IS OFF
WAIT FOR 10 SECONDS
RED LED GOES OFF
GREEN LED GOES ON
GIVE 10 SECONDS FOR PEOPLE TO CROSS
GREEN LED GOES OFF
RED LED TURNS ON

convert

TO CODE

```
from picozero import  LED, Buzzer, Button
from time import sleep
red_led = LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)
red_led.on()
green_led.off()
sleep(10)
red_led.off()
green_led.on()
sleep(10)
green_led.off()
red_led.on()
```

RUN and
Save this program as ex3c.py

Exercise 3 – Turning the circuit into a pedestrian crossing with code

ex3c.py

```
from picozero import  LED, Buzzer, Button
from time import sleep
red_led = LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)
red_led.on()
green_led.off()
sleep(10)
red_led.off()
green_led.on()
sleep(10)
green_led.off()
red_led.on()
```

RUN ONCE

convert

CONVERT THE
RUN ONCE PROGRAM
TO RUN FOREVER

RUN AND SAVE
PROGRAM AS
ex3d.py

RUN FOREVER

ex3d.py

```
from picozero import LED, Buzzer, Button
from time import sleep
red_led = LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)

red_led.on()
green_led.off()

while True:
    sleep(10)
    red_led.off()
    green_led.on()
    sleep(10)
    green_led.off()
    red_led.on()
```

How to convert this into a functon

Give a name for your function. Let's call it greenman()

```
from picozero import LED, Buzzer, Button
from time import sleep
red_led = LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)
red_led.on()
green_led.off()
def greenman():
    sleep(10)
    red_led.off()
    green_led.on()
    sleep(10)
    green_led.off()
    red_led.on()
while True:
    greenman()
```

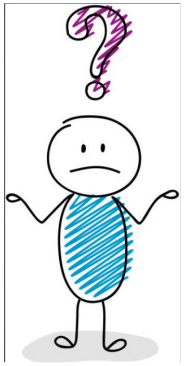
Run program
Save it as ex3e.py

ex3e.py

```
from picozero import LED, Buzzer, Button
from time import sleep
red_led = LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)
red_led.on()
green_led.off()
def greenman():
    sleep(10)
    red_led.off()
    green_led.on()
    sleep(10)
    green_led.off()
    red_led.on()
while True:
    greenman()
```

WAIT FOR 10 SECONDS
RED LED GOES OFF
GREEN LED GOES ON
GIVE 10 SECONDS FOR PEOPLE TO CROSS
GREEN LED STARTS TO BLINK 10 TIMES (10 SECONDS)
BUZZER ALSO BLINKS 10 TIMES (10 SECONDS)
GREEN LED GOES OFF
RED LED TURNS ON

GREEN LED STARTS TO BLINK 10 TIMES (10 SECONDS)
BUZZER ALSO BLINKS 10 TIMES (10 SECONDS)




```
def blinkblink():
    for x in range(1,11,1):
        green_led.on()
        buzzer.on()
        sleep(0.5)
        green_led.off()
        buzzer.off()
        sleep(0.5)
```

10 X

[ex3b.py]

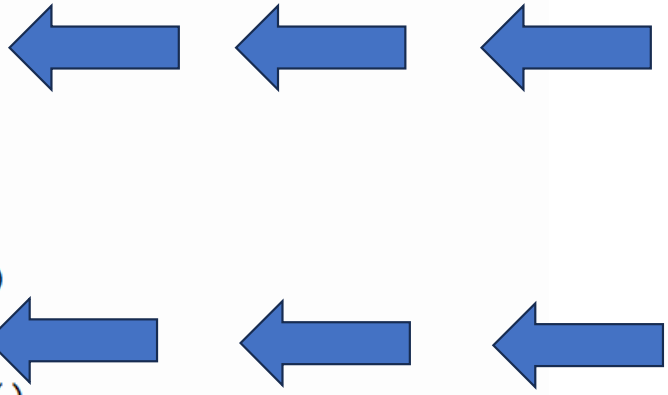
```
from picozero import LED, Buzzer, Button
from time import sleep
red_led=LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)
red_led.on()
green_led.off()



def greenman():
    sleep(10)
    red_led.off()
    green_led.on()
    sleep(10)

    green_led.off()
    red_led.on()


while True:
    greenman()
```



```
1  from picozero import LED, Buzzer, Button
2  from time import sleep
3  red_led=LED(15)
4  green_led=LED(14)
5  buzzer=Buzzer(12)
6  button=Button(13)
7  red_led.on()
8  green_led.off()
9
10 def blinkblink():
11     for x in range(1,11,1):
12         green_led.on()
13         buzzer.on()
14         sleep(0.5)
15         green_led.off()
16         buzzer.off()
17         sleep(0.5)
18
19 def greenman():
20     sleep(10)
21     red_led.off()
22     green_led.on()
23     sleep(10)
24     blinkblink()
25     green_led.off()
26     red_led.on()
27
28 while True:
29     greenman()
```

Run program
Save it as ex3f.py

ex3f.py

```
1  from picozero import LED, Buzzer, Button
2  from time import sleep
3  red_led=LED(15)
4  green_led=LED(14)
5  buzzer=Buzzer(12)
6  button=Button(13)
7  red_led.on()
8  green_led.off()
9  def blinkblink():
10     for x in range(1,11,1):
11         green_led.on()
12         buzzer.on()
13         sleep(0.5)
14         green_led.off()
15         buzzer.off()
16         sleep(0.5)
17 def greenman():
18     sleep(10)
19     red_led.off()
20     green_led.on()
21     sleep(10)
22     blinkblink()
23     green_led.off()
24     red_led.on()
25
26 while True:
27     greenman()
```

This program works fine.
But in the real world, it's not practical.
Why?

If no one wants to cross, the program will still continue to work

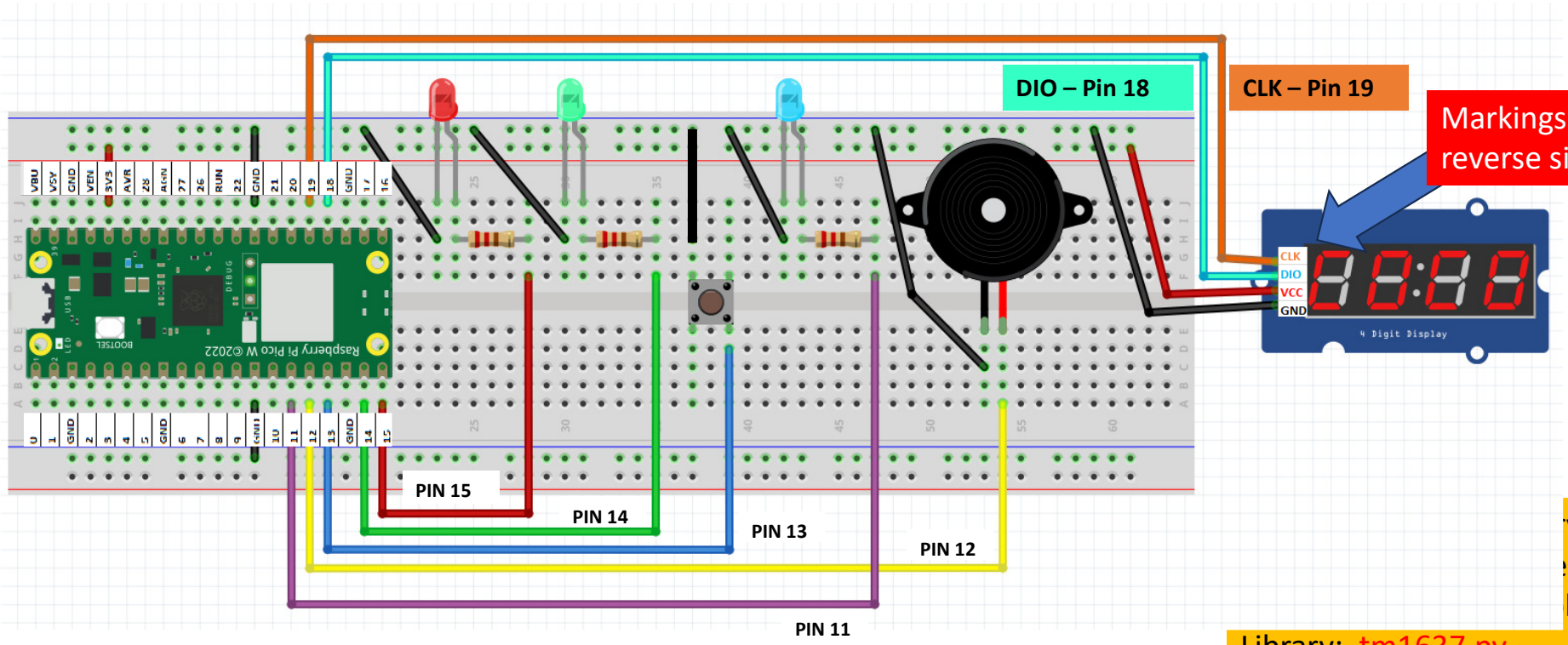
We want this program to run only
when someone presses the button. How can we make use of the button

```
26 while True:
27     greenman()
```

```
25
26 button.when_pressed = greenman
```

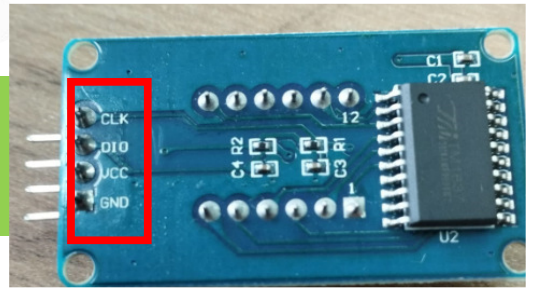
Make this change
Run program
Save it as ex3g.py

Exercise 4 – Full Fledge Pedestrian Crossing



Library: tm1637.py

EXERCISE 4a – THE 7 SEGMENT DISPLAY




```
>>> from machine import Pin
>>> import tm1637
>>> tm = tm1637.TM1637(clk=Pin(19), dio=Pin(18))
```

```
>>> tm.show("help")
```

```
>>> tm.number(1234)
```

```
>>> tm.temperature(24)
```

```
>>> tm.show(" " * 4)
```

In python " " * 4 means 4 SPACES
This will clear the display.
Make sure there is a SPACE in between the quotation marks

Exercise 4a. Display the following
PICO
95
70 Degrees Celcius
Clear the display

COUNT DOWN DISPLAY

```
from machine import Pin
from time import sleep
import tm1637
tm = tm1637.TM1637(clk=Pin(19), dio=Pin(18))

for x in range(10,-1,-1):
    tm.number(x)
    sleep(1)
```

Run program
Save as ex4b.py

Try Counting from 0 to 20.
Refer to ex3f.py if you are not sure how to count upwards

Modify ex3c.py to add Count down in blinkblink()

ADDING COUNTER TO PEDESTRIAN CROSSING blinkblink()

Besides the green man flashing and the buzzer beeping
the counter will show how much time left for crossing

```
1  from picozero import LED, Buzzer, Button
2  from machine import Pin
3  import tm1637
4  from time import sleep
5  red_led=LED(15)
6  green_led=LED(14)
7  buzzer=Buzzer(12)
8  button=Button(13)
9  tm = tm1637.TM1637(clk=Pin(19), dio=Pin(18))
10 red_led.on()
11 green_led.off()
12 def blinkblink():
13     for x in range(10,-1,-1):
14         tm.number(x)
15         green_led.on()
16         buzzer.on()
17         sleep(0.5)
18         green_led.off()
19         buzzer.off()
20         sleep(0.5)
21 def greenman():
22     sleep(10)
23     red_led.off()
24     green_led.on()
25     sleep(10)
26     blinkblink()
27     green_led.off()
28     red_led.on()
29
30 button.when_pressed =    greenman
```

Run this amended program
Save as ex4a.py

Modify ex4a.py to add antispam control

```
1  from picozero import LED, Buzzer, Button
2  from machine import Pin
3  import tm1637
4  from time import sleep
5  red_led=LED(15)
6  green_led=LED(14)
7  buzzer=Buzzer(12)
8  button=Button(13)
9  tm=tm1637.TM1637(clk=Pin(19),dio=Pin(18))
10 antispam_led=LED(11)
11 antispam_led.off()
12 red_led.on()
13 green_led.off()
14 def blinkblink():
15     for x in range(10,-1,-1):
16         tm.number(x)
17         green_led.on()
18         buzzer.on()
19         sleep(0.5)
20         green_led.off()
21         buzzer.off()
22         sleep(0.5)
23 def check():
24     if antispam_led.on():
25         pass
26     else:
27         antispam_led.on()
28         greenman()
```

```
32 def greenman():
33     sleep(10)
34     red_led.off()
35     green_led.on()
36     sleep(10)
37     blinkblink()
38     green_led.off()
39     red_led.on()
40     antispam_led.off()
41
43 button.when_pressed = check
```

Run Program
Save as ex4b.py

Python Lesson – Session # 3

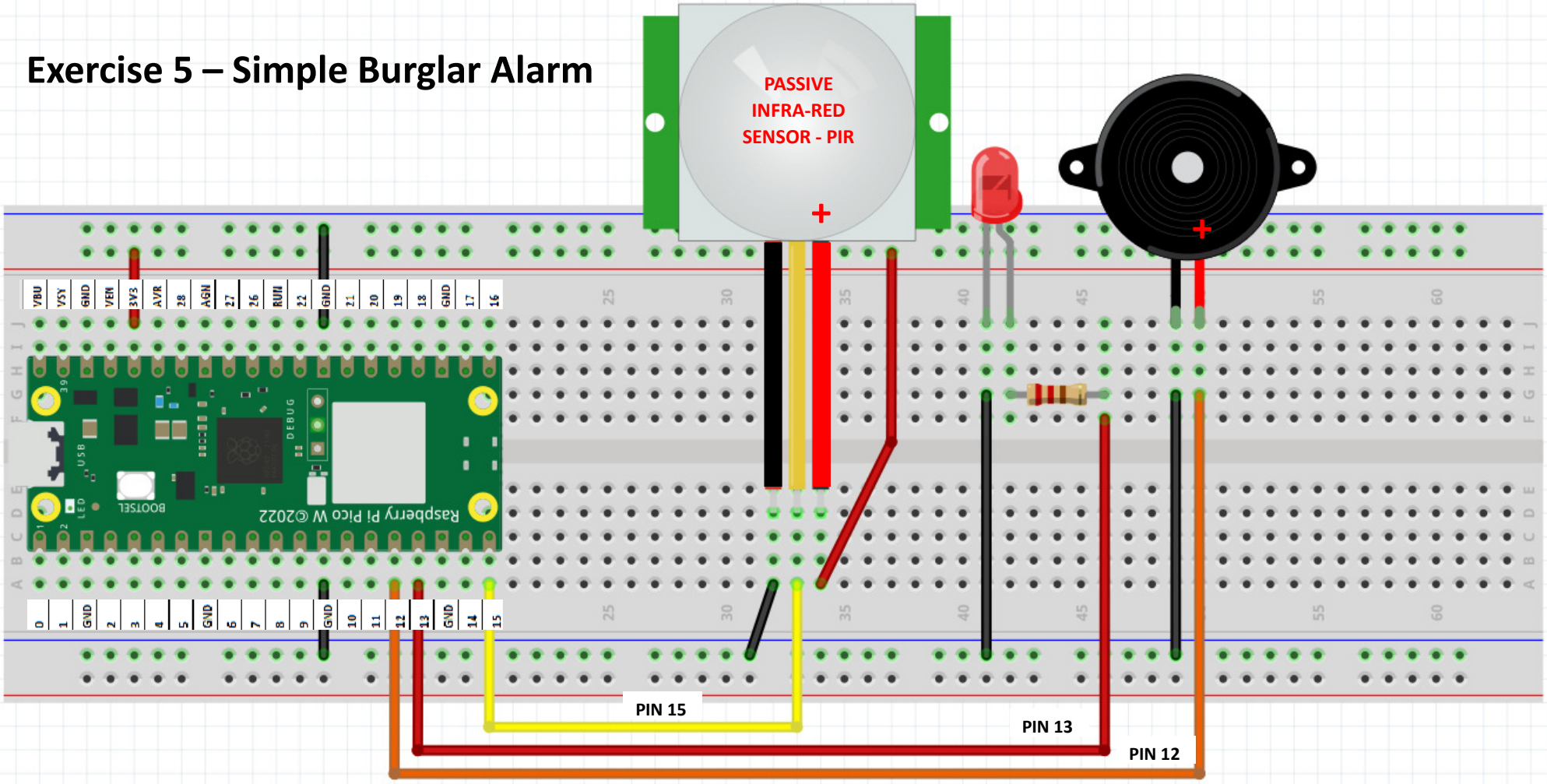
TROUBLESHOTTING – 4 BASIC ERRORS

PUTTING PYTHON KNOWLEDGE TO USE

PASSWORD

INPUT STATEMENT

Exercise 5 – Simple Burglar Alarm



HOW DOES A PIR WORK

- In summary, the HC-SR501 PIR sensor is a device that can detect motion by sensing changes in the infrared heat patterns around it.
  - It's like having a pair of eyes that can "see" heat, and when it detects movement, it can trigger other devices or actions based on that detection.

Actually, it works like a Button
Remember what happens when
A button is pressed in the
Earlier example

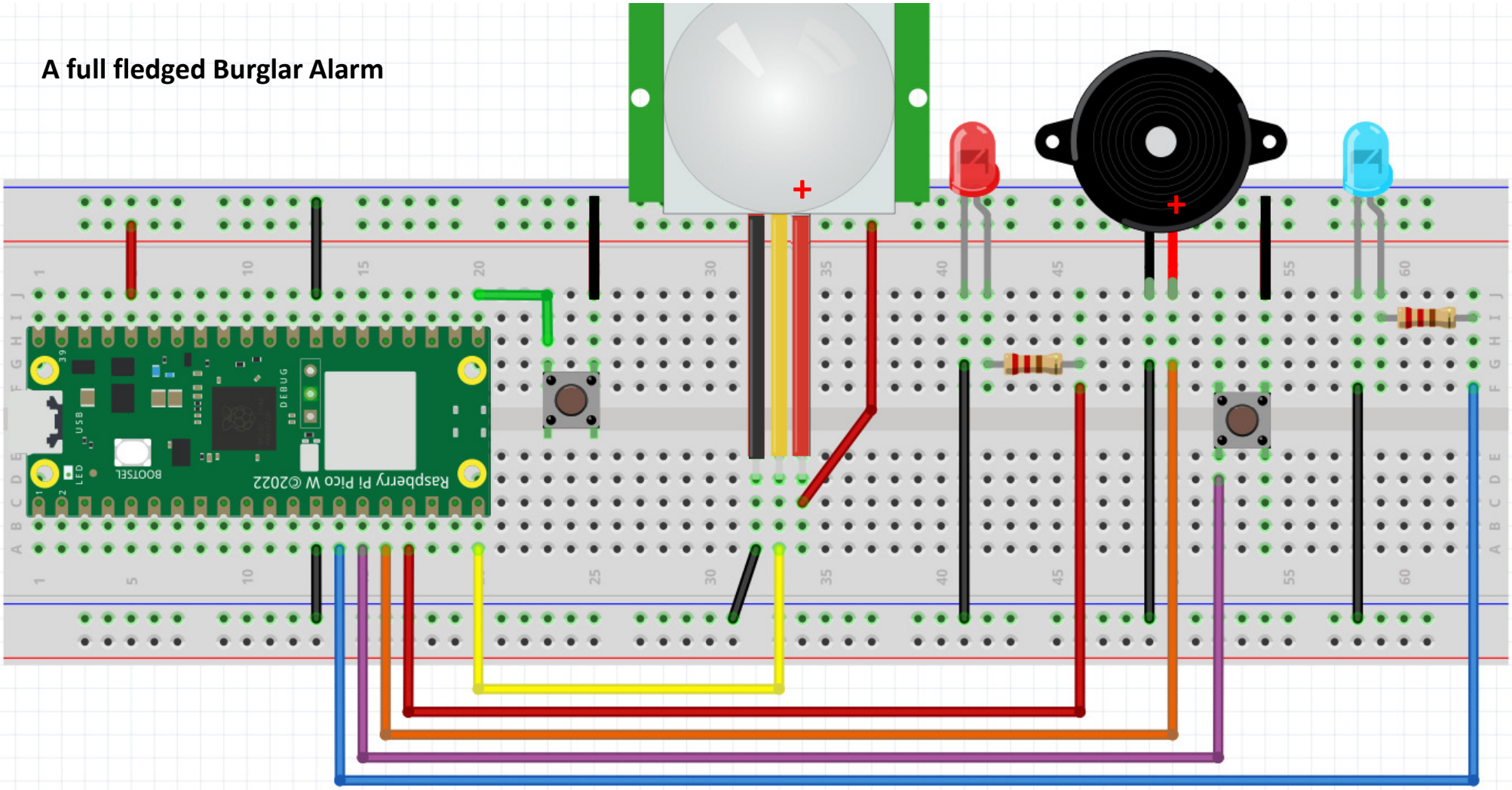
- We set the Burglar Alarm On
  - at Night (ARM)
- We turn it off in the morning (DISARM)
- To DISARM a password is needed

```
18  def greenman():
19      sleep(10)
20      red_led.off()
21      green_led.on()
22      sleep(10)
23      blinkblink()
24      green_led.off()
25      red_led.on()
26
27  red_led.on()
28  green_led.off()
29  while True:
30      if button.value()==1:
31          greenman()
```

Run program
Save as ex5a.py

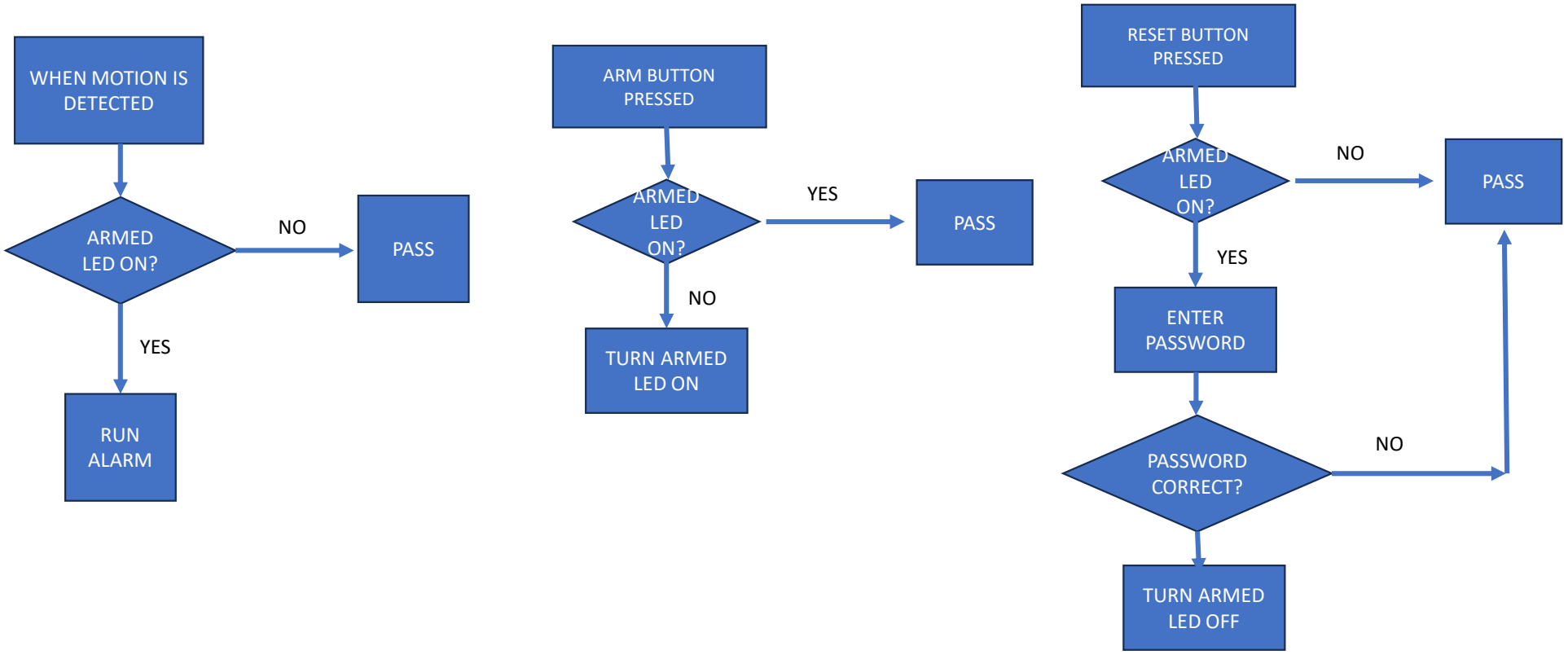
```
1   from  machine import Pin
2   from picozero import LED, Buzzer
3   from time import sleep
4   led=LED(13)
5   buzz=Buzzer(12)
6   sensor_pir=Pin(15,Pin.IN)
7   def alarm():
8       led.on()
9       buzz.on()
10      sleep(.5)
11      led.off()
12      buzz.off()
13      sleep(.5)
14
15  while True:
16      if sensor_pir.value()==1:
17          alarm()
```

A full fledged Burglar Alarm



Ex 5b. A full fledged Burglar Alarm

- A Burglar alarm should work only when it is activated
- And should be deactivated when it is not required



CODE FOR THE FULL FLEDGED BURGLAR ALARM

```
1  from machine import Pin
2  from picozero import Button,Buzzer,LED
3  from time import sleep
4
5  sensor_pir = Pin(15,Pin.IN)
6  led = LED(13)
7  buzz = Buzzer(12)
8
9  arm_button=Button(16)
10 armed_led = LED(10)
11 disarm_button=Button(11)
12
13 password='12345'
```

```
15 def alarm():
16     for x in range(1,11,1):
17         led.on()
18         buzz.on()
19         sleep(0.5)
20         led.off()
21         buzz.off()
22         sleep(0.5)
23
```

```
24 def arm_burglar_alarm():
25     if armed_led.value == 0 :
26         armed_led.on()
27     else:
28         pass
29
```

```
30 def disarm_burglar_alarm():
31     if armed_led.value == 0:
32         pass
33     else:
34         pwd = input('Enter Disarm Password: ')
35         if pwd == password:
36             armed_led.off()
37         else:
38             pass
39
```

```
40 armed_led.off()
41 buzz.off()
42
43 arm_button.when_pressed = arm_burglar_alarm
44 disarm_button.when_pressed = disarm_burglar_alarm
45
46 while True:
47     if sensor_pir.value() == 1 and armed_led.value == 1:
48         alarm()
```

Run program
Save as ex5b.py

Some perculiarity to take note off

Patrol Car Ex – Putting knowledge gained to use

Use your blue , red LED and a Buzzer to make a noisy flashing lights on a patrol car

Choose any pin you wish

Wire up the circuit and write the python code to make it work

HINT

Material

1 x Red LED

1 x Blue LED

1 x Buzzer

Python Code

machine and time library

while True:

Red LED on and off

Blue LED on and off

Sleep in between on and off

Use this example

```
from picozero import LED, Buzzer, Button
from time import sleep
red_led=LED(15)
green_led=LED(14)
buzzer=Buzzer(12)
button=Button(13)
red_led.on()
green_led.off()

def greenman():
    sleep(10)
    red_led.off()
    green_led.on()
    sleep(10)

    green_led.off()
    red_led.on()

while True:
    greenman()
```

Save your program
As **patrolcar.py**
And RUN

Solution

```
from picozero import LED, Buzzer
from time import sleep
red_led=LED(17)
blue_led=LED(11)
buzz=Buzzer(12)

def flashing():
    red_led.on()
    buzz.on()
    sleep(0.3)

    red_led.off()
    buzz.off()

    blue_led.on()
    buzz.on()
    sleep(0.3)

    blue_led.off()
    buzz.off()

while True:
    flashing()
```

Python Lesson – Session # 4

REVISION

INTRODUCTION TO CHATGPT

Refer to Traffic Light Wiring Diagram

- **How to use CHATGPT to write us the program**
- **Keywords in prompt**
  - **Micropython**
  - **Pico W**
  - **Use picozero library**
  - **Specify components and pin numbers**
  - **Provide the algorithm if available**