## 5.2 Rpi Making a GUI

Questions 1) and 2):

<https://github.com/TheCollo67/RPiGUI>

## Operating LEDs

Let’s program our LEDs to turn on and off based on our commands. Using gpiozero library to control LEDs, we need to import them first. Type the following code to do that.

from gpiozero import LED

Now we need to assign variables for our 3 LEDs that we wired.

led17 = LED(17)

led18 = LED(18)

led23 = LED(23)

Now to turn them on, type the following

led17.on()

led18.on()

led23.on()

To turn them off, type the following

led17.off()

led18.off()

led23.off()

Let’s make these LEDs blink. When there is blinking, there is some timing involved. To allow python to use we need to import a small part of another library.

from time import sleep

To make the LEDs blink we need to create a loop. We will use a while loop.

while True:

led17.on()

sleep(1)

led17.off()

sleep(1)

To break the loop, press CTRL + C

We can also blink LEDs using gpiozero’s blink function.

led17.blink()

We can also change the on and off times.

led18.blink(1, 0.5)

Turn the LEDs off by using the following codes

led17.off()

led18.off()

Make the 3 LEDs blink at different rates using blink()

## Releasing pins

Pin 17, 18, 23 are now locked with led17, led18, led23 variables and cannot be reused till they have been released. Attempting to reuse them will cause an error. To release them we type the following code

led17.close()

led18.close()

led23.close()

## Using Button input

We need to import Button from gpiozero library.

from gpiozero import Button

Now assign a variable and pin to our button by typing

btn5 = Button(5)

Let’s create a loop which will tell us whether the button is being pressed or not.

while True:

if btn5.is\_pressed:

print(“Button is pressed”)

else:

print(“Button is not pressed”)

btn5.wait\_for\_press()

print(“Button was pressed”)

This program stops as soon as you press the button and it also shows True, which also means that the button was pressed. If you don’t put a message, all you will see is True. Let’s associate our button to an LED. Type the following code

led17 = LED(17)

btn5.when\_pressed = led17.on

btn5.when\_released = led17.off

Now the LED 17 will be on as long as your button is on.

def btnpressed():

print(“Button was pressed”)

We still need to associate the button to this function.

btn5.when\_pressed = btnpressed

We could toggle an LED every time we press the button.

led12 = LED(12)

def ledtoggle():

led12.toggle()

btn5.when\_pressed = ledtoggle

Now every time you press the button, the LED 12 will toggle its state.

Toggle all 3 LEDs when the button is pressed

## Control a Collection of LEDs

If we would like to control multiple LEDs together, we will need to use LEDBoard. First we will import this from the gpiozero library.

from gpiozero import LEDBoard

Now assign all 3 LEDs as an LEDBoard. (make sure you have release all the LEDs using close() command before you program the LEDBoard)

leds = LEDBoard (17, 18 , 23)

To turn them all on or off we use the same command as we did for the LED.

leds.on()

and

leds.off()

We can make them blink as well

leds.blink()

If you wish to control them individually, you do the following

leds.value = (1, 0, 1)

Just like LED, you can release the LEDBoard by using close() command.

Question 3)

An example application is Traffic lights

A full traffic lights system.

Using a [**TrafficLights**](https://gpiozero.readthedocs.io/en/stable/api_boards.html#gpiozero.TrafficLights) kit like Pi-Stop:

from gpiozero import TrafficLights

from time import sleep

lights = TrafficLights(2, 3, 4)

lights.green.on()

while True:

sleep(10)

lights.green.off()

lights.amber.on()

sleep(1)

lights.amber.off()

lights.red.on()

sleep(10)

lights.amber.on()

sleep(1)

lights.green.on()

lights.amber.off()

lights.red.off()

Alternatively:

from gpiozero import TrafficLights

from time import sleep

from signal import pause

lights = TrafficLights(2, 3, 4)

def traffic\_light\_sequence():

while True:

yield (0, 0, 1) # green

sleep(10)

yield (0, 1, 0) # amber

sleep(1)

yield (1, 0, 0) # red

sleep(10)

yield (1, 1, 0) # red+amber

sleep(1)

lights.source = traffic\_light\_sequence()

pause()

Using [**LED**](https://gpiozero.readthedocs.io/en/stable/api_output.html#gpiozero.LED) components:

from gpiozero import LED

from time import sleep

red = LED(2)

amber = LED(3)

green = LED(4)

green.on()

amber.off()

red.off()

while True:

sleep(10)

green.off()

amber.on()

sleep(1)

amber.off()

red.on()

sleep(10)

amber.on()

sleep(1)

green.on()

amber.off()

red.off()

Question 4)

A collection of LEDs can be accessed using [**LEDBoard**](https://gpiozero.readthedocs.io/en/stable/api_boards.html#gpiozero.LEDBoard):

from gpiozero import LEDBoard

from time import sleep

from signal import pause

leds = LEDBoard(5, 6, 13, 19, 26)

leds.on()

sleep(1)

leds.off()

sleep(1)

leds.value = (1, 0, 1, 0, 1)

sleep(1)

leds.blink()

pause()

Using [**LEDBoard**](https://gpiozero.readthedocs.io/en/stable/api_boards.html#gpiozero.LEDBoard) with pwm=True allows each LED’s brightness to be controlled:

from gpiozero import LEDBoard

from signal import pause

leds = LEDBoard(5, 6, 13, 19, 26, pwm=True)

leds.value = (0.2, 0.4, 0.6, 0.8, 1.0)

pause()

from gpiozero import LED, Button, LightSensor, RGBLED

from time import sleep

import random

import os

from guizero import App, Text, TextBox, PushButton, Box, Waffle, Picture

from sense\_hat import SenseHat

led17 = LED(17)

led27 = LED(27)

led22 = LED(22)

btn18 = Button(18)

btn12 = Button(12)

btn16 = Button(16)

btn20 = Button(20)

ldr5 = LightSensor(5, 5, 0.01, 0.6)

colorled = RGBLED(red=6, green=19, blue=13)

sense = SenseHat()

def allon():

led17.on()

my\_waffle.set\_pixel(0,0,"blue")

#led17\_status.color("blue")

led27.on()

my\_waffle.set\_pixel(1,0,"red")

#led27\_status.color("red")

led22.on()

my\_waffle.set\_pixel(2,0,"orange")

#led22\_status.color("orange")

playsound2()

def alloff():

led17.off()

#led17\_status.color("black")

led27.off()

#led27\_status.color("black")

led22.off()

#led22\_status.color("black")

my\_waffle.set\_all("grey")

colorled.color = (0,0,0)

playsound1()

def playsound1():

os.system('omxplayer /home/pi/Downloads/example.mp3')

def playsound2():

os.system('omxplayer /home/pi/Downloads/pingas.mp3')

def playsound3():

os.system('omxplayer /home/pi/Downloads/billygoat.mp3')

def randomcolors():

r = round(random.random(),2)

g = round(random.random(),2)

b = round(random.random(),2)

colorled.color=(r,g,b)

playsound3()

def seq():

led17.off(), led27.off(), led22.off()

my\_waffle.set\_all("grey")

#led17\_status.color("black")

#led27\_status.color("black")

#led22\_status.color("black")

led17.on()

my\_waffle.set\_pixel(0,0,"blue")

#led17\_status.color("blue")

sleep(0.5)

led27.on()

my\_waffle.set\_pixel(1,0,"red")

#led27\_status.color("red")

sleep(0.5)

led22.on()

my\_waffle.set\_pixel(2,0,"orange")

#led22\_status.color("orange")

sleep(0.5)

playsound2()

def seqoff():

led17.on(), led27.on(), led22.on()

#led17\_status.color("blue")

#led27\_status.color("red")

#led22\_status.color("orange")

my\_waffle.set\_pixel(0,0,"blue")

my\_waffle.set\_pixel(1,0,"red")

my\_waffle.set\_pixel(2,0,"orange")

led17.off()

my\_waffle.set\_pixel(0,0,"grey")

#led17\_status.color("black")

sleep(0.5)

led27.off()

my\_waffle.set\_pixel(1,0,"grey")

#led27\_status.color("black")

sleep(0.5)

led22.off()

my\_waffle.set\_pixel(2,0,"grey")

#led22\_status.color("black")

sleep(0.5)

playsound3()

def show\_name():

sense.show\_message(my\_textbox.get())

my\_textbox.set("")

def get\_temp():

temp\_data = round(sense.get\_temperature(), 2)

return(temp\_data)

def get\_pres():

pres\_data = round(sense.get\_pressure(), 2)

return(pres\_data)

def get\_hum():

hum\_data = round(sense.get\_humidity(), 2)

return(hum\_data)

def update\_data():

temp\_data.set(get\_temp())

temp\_data.after(1000, update\_data)

pres\_data.set(get\_pres())

hum\_data.set(get\_hum())

def ledblue():

led17.toggle()

if led17.is\_lit:

my\_waffle.set\_pixel(0,0,"blue")

else:

my\_waffle.set\_pixel(0,0,"grey")

def ledred():

led27.toggle()

if led27.is\_lit:

my\_waffle.set\_pixel(1,0,"red")

else:

my\_waffle.set\_pixel(1,0,"grey")

def ledorange():

led22.toggle()

if led22.is\_lit:

my\_waffle.set\_pixel(2,0,"orange")

else:

my\_waffle.set\_pixel(2,0,"grey")

btn18.when\_pressed = allon

btn12.when\_pressed = alloff

btn16.when\_pressed = seq

btn20.when\_pressed = randomcolors

app = App(title="Open Day Demo", height=1080, width=980)

#app.attributes("-fullscreen", True)

welcome\_message = Text(app, text="Welcome to Electrotechnology Open Day", size=40, font='Times New Roman', color="blue")

my\_textbox = TextBox(app, width=20)

scroll\_text = PushButton(app, command = show\_name, text="Show the name in scroll", pady=20, padx=20)

my\_waffle = Waffle(app, height=1, width=3, dim=40, dotty=True)

my\_waffle.set\_all("grey")

box1 = Box(app, layout='grid')

pb1 = PushButton(box1, command=ledblue, text = "Blue LED", grid=[0,0])

pb2 = PushButton(box1, command=ledred, text = "Red LED", grid=[0,1])

pb3 = PushButton(box1, command=ledorange, text = "Orange LED", grid=[0,2])

box2 = Box(app, layout='grid')

temp\_img = Picture(box2, image='/home/pi/Pictures/temp150.gif', grid=[0,0])

#temp\_text = Text(box2, text="Temperature", grid=[0,0])

temp\_data = Text(box2, text=get\_temp(), size=30, grid=[0,1])

temp\_data.after(1000, update\_data)

pres\_img = Picture(box2, image='/home/pi/Pictures/pres150.gif', grid=[1,0])

#pres\_text = Text(box2, text="Pressure", grid=[1,0])

pres\_data = Text(box2, text=get\_pres(), size=30, grid=[1,1])

hum\_img = Picture(box2, image='/home/pi/Pictures/hum150.gif', grid=[2,0])

#hum\_text = Text(box2, text="Humidity", grid=[2,0])

hum\_data = Text(box2, text=get\_hum(), size=30, grid=[2,1])

app.display()