***Aim of the Experiment:***

Physical Computation of different Embedded System Projects using push-button switch interfacing with Raspberry Pi and LEDs.

***Objective:***

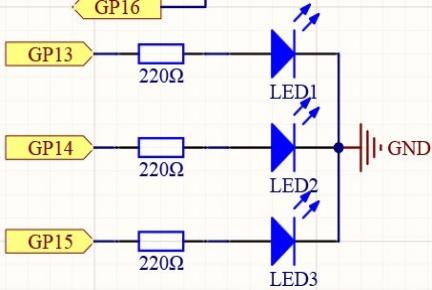
1. Implementation of **mini pedestrian crossing system** (simple traffic light control system) using multiple LEDs without push-button.
2. Study the **effect of Multithreading concept** on processing time of an embedded system using Micro-python script (e.g. Comparing the processing time to calculate the **square and cube of a series of numbers** with and without multithreading concept).
3. Implementation of mini pedestrian crossing system with proper control (**Puffin Crossing System**) based on the concept of **multi-threading** using multiple LEDs with a push-button and a buzzer.
4. Implementation of **Single player** based **Simple Reaction Timing Game** (Mental Chronometry) based on the concept of Interrupt Requests, or IRQs using a push-button.
5. Implementation of **Two player** based Simple Reaction Timing Game (**Mental Chronometry**) based on the concept of **Interrupt Requests, or IRQs** using a push- button.
6. Implementation of **changing the LED flashing rate** using pushbutton as external interrupts with external and internal Pull-up resistors.
7. Implementation of a **Digital Dice Game** using LEDs and Push-buttons.
8. Implementation of **Lucky Day of the Week** using LEDs and Push-buttons.

***Components/Equipment/items Required:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl**  **No.** | **Name of the**  **Component/Equipment** | **Specification** | **Quantity** |
| **1** | Raspberry Pi Pico | RP2040 microcontroller  chip, 125MHz | 1 |
| **2** | Raspberry Pi Pico cable | USB Type A to Micro-B | 1 |
| **3** | Resistors (carbon type) | ¼ watt (330 Ω) | 10 |
| **4** | LED | 3mm, Red | 8 |
| **5** | Tactile Push Button  Switches | 6 x 6 x 6 mm | 2 |
| **6** | Breadboard | 840 Tie points | 1 |
| **7** | Jumper Wire | --------------------------- | As per requirement |

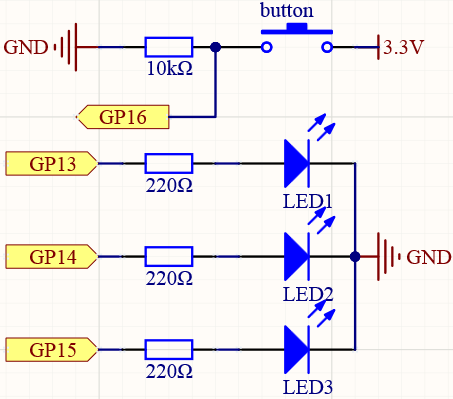
***Circuit/Schematic Diagram:***

# Objective 1



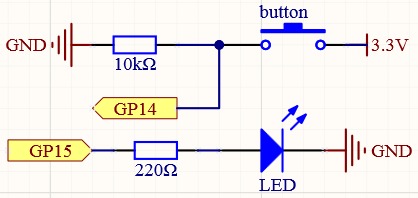
(Figure 1: Circuit diagram for implementation of mini pedestrian crossing system.)

# Objective 3



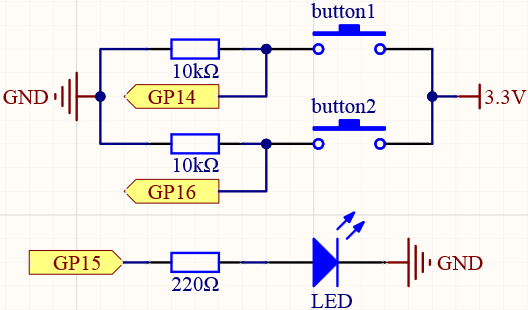
(Figure 2: Circuit diagram for implementation of Puffin crossing system.)

# Objective 4



(Figure 3: Circuit diagram for implementation of Single player based Simple Reaction Timing Game.)

# Objective 5

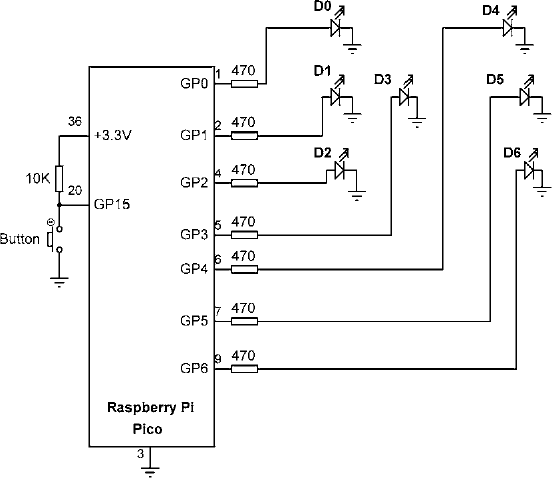


(Figure 4: Circuit diagram for implementation of Two player based Simple Reaction Timing Game.)

# Objective 6

(Figure 5: Circuit diagram for implementation of changing the LED flashing rate using pushbutton with Pull-up resistors.)

# Objective 7



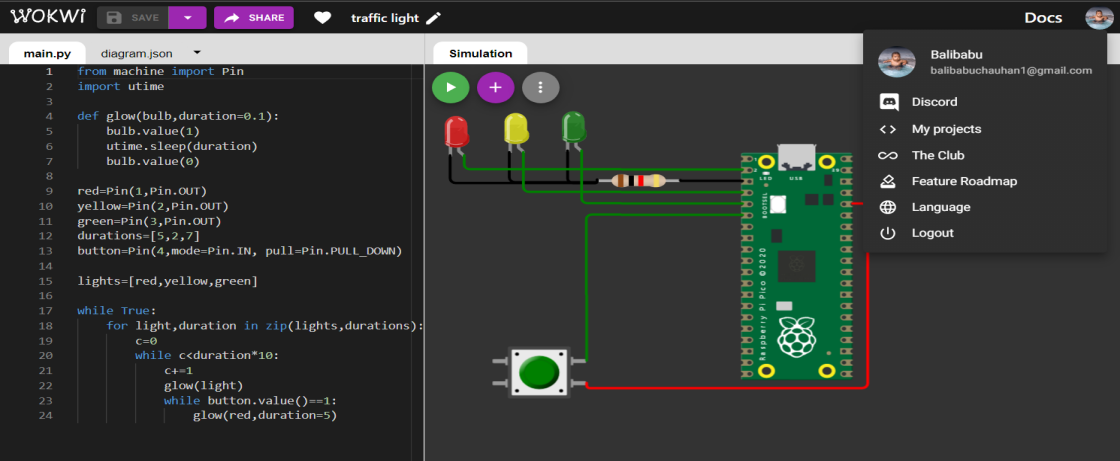
(Figure 6: Circuit diagram for implementation of a Digital Dice Game using LEDs and Push-buttons.)

# Objective 8

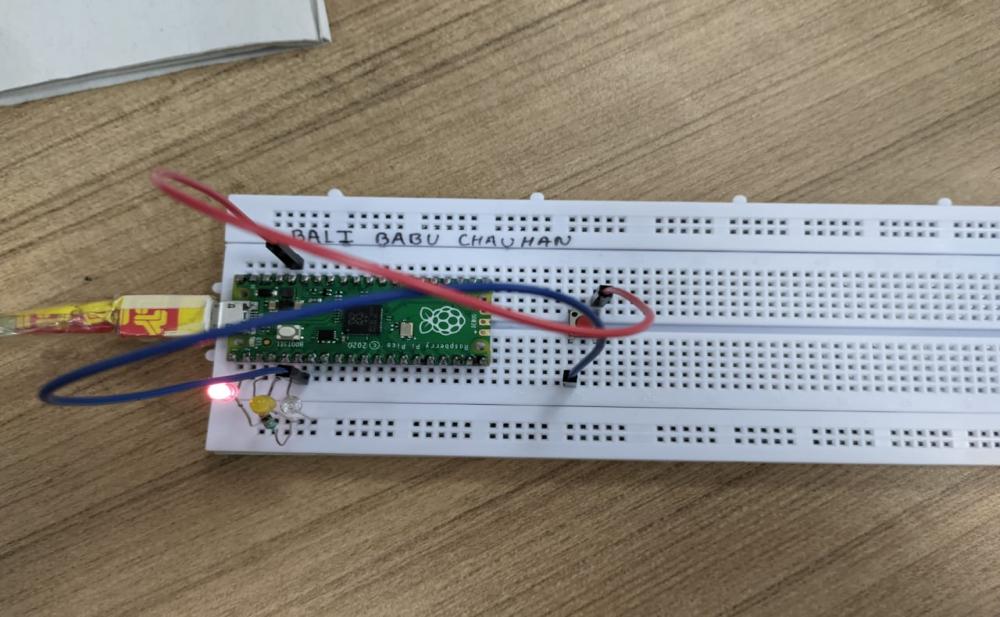
(Figure 7: Circuit diagram for implementation of Lucky Day of the Week using LEDs and Push-buttons.)

***Observation:***

# Objective 1

****

(Figure 8: Simulation based electronic circuit for implementation of mini pedestrian crossing system.)



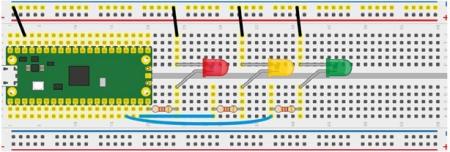
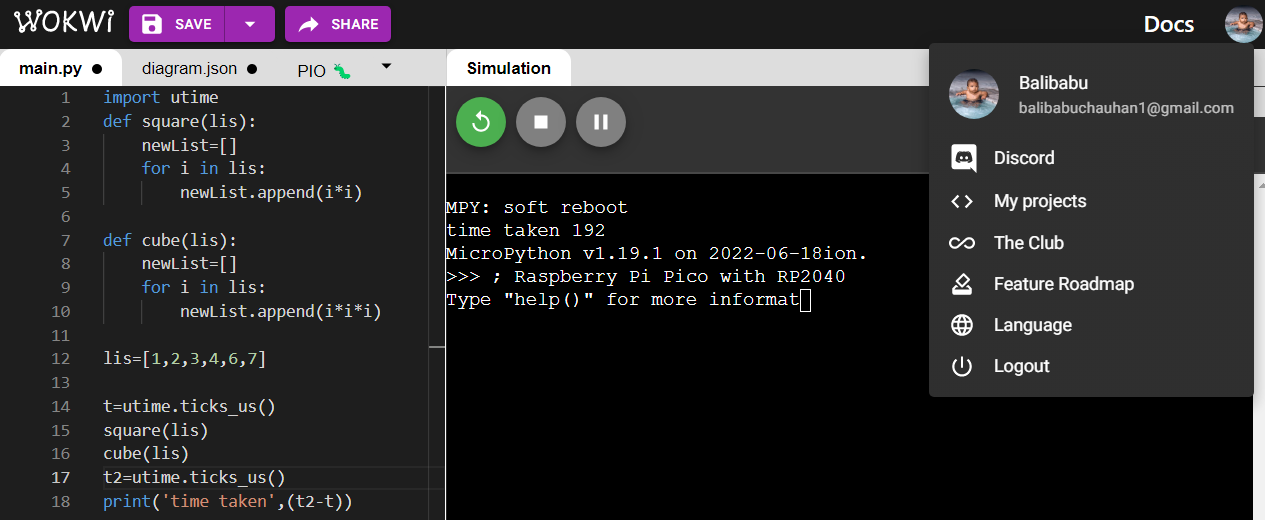


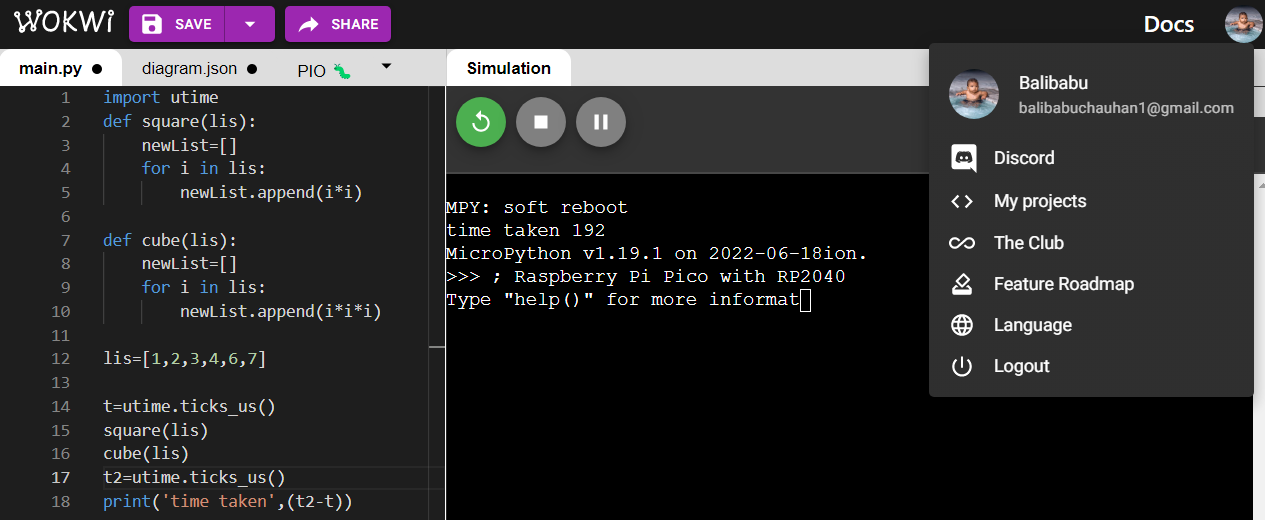
Figure 9: (Breadboard Schematic representation of an electronic circuit for implementation of mini pedestrian crossing system.)

Figure 10: (Hardware implementation based electronic circuit for implementation of mini pedestrian crossing system)

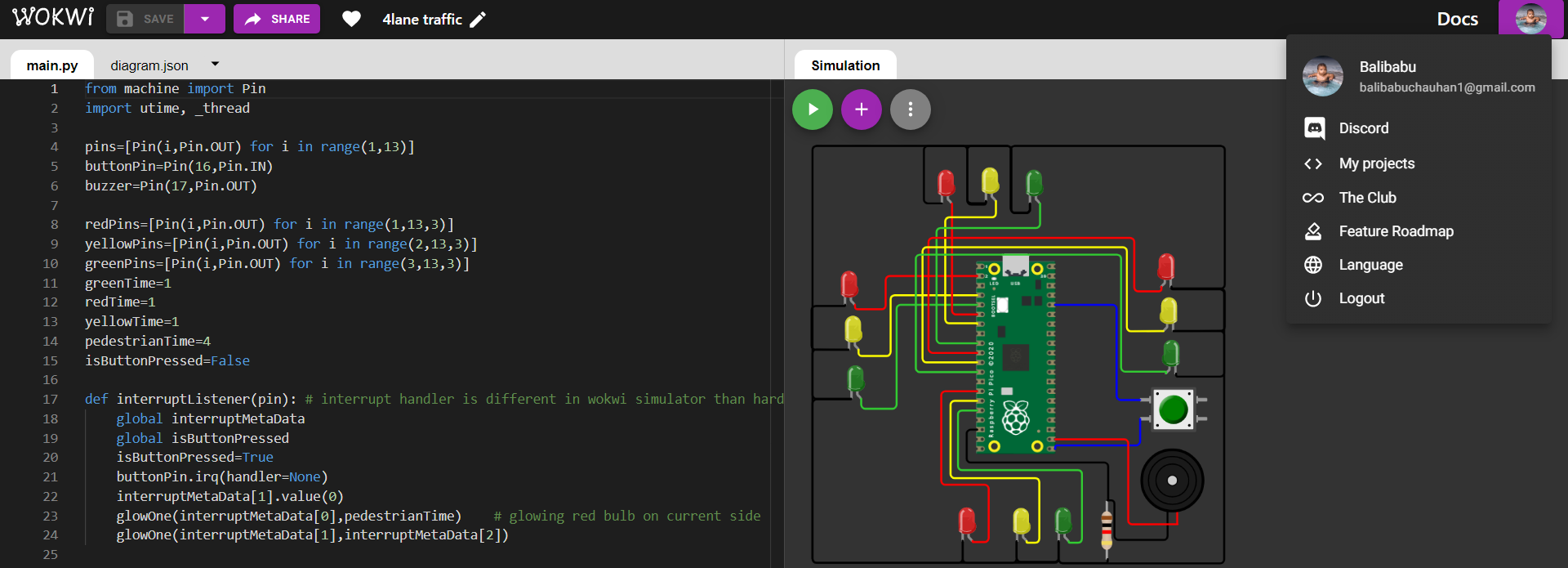
# Objective 2

****

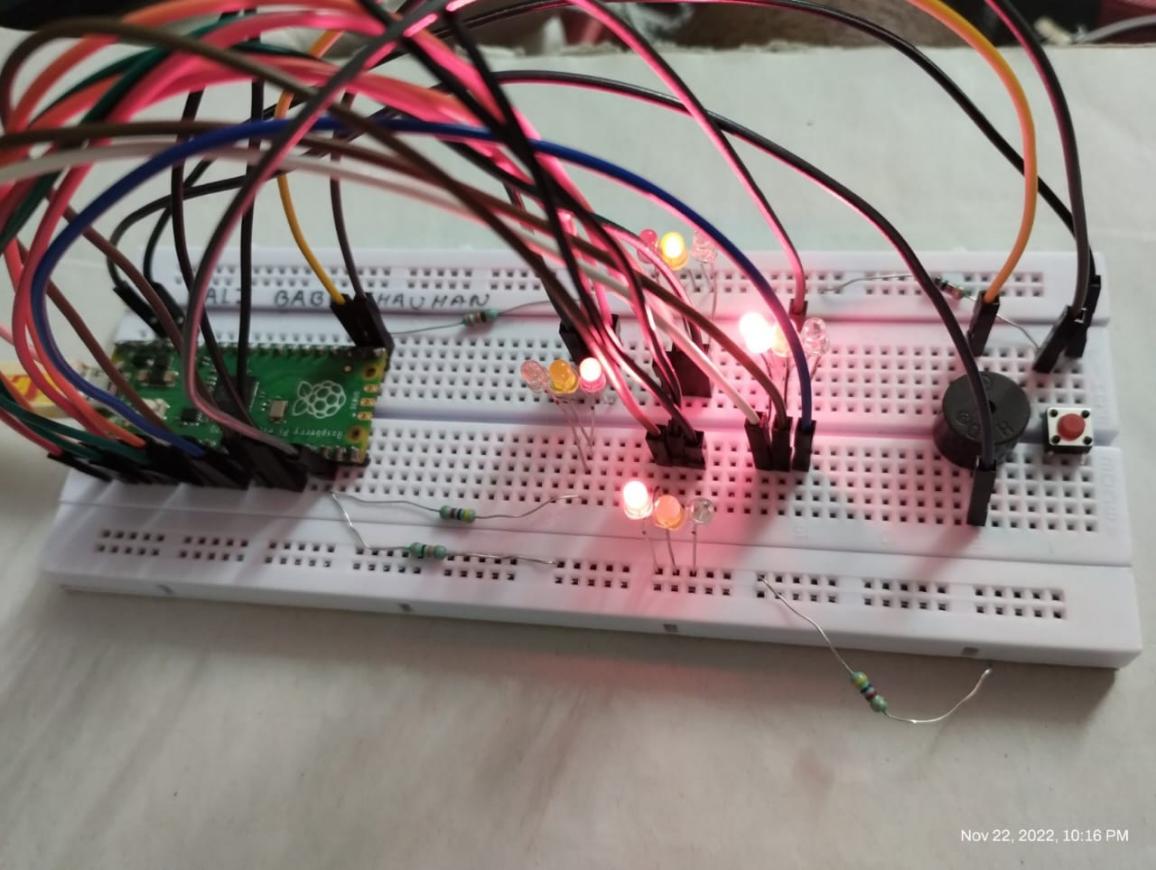
(Figure 11: Simulation based Comparing the processing time to calculate the square and cube of a series of numbers without multithreading concept.)

****

(Figure 12: Simulation based Comparing the processing time to calculate the square and cube of a series of numbers without multithreading concept.)



(Figure 13: Simulation based electronic circuit for implementation of Puffin crossing system.)



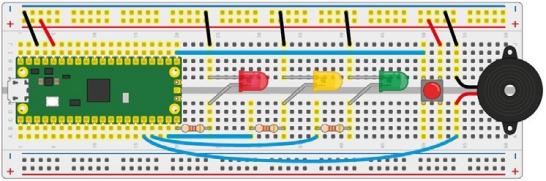
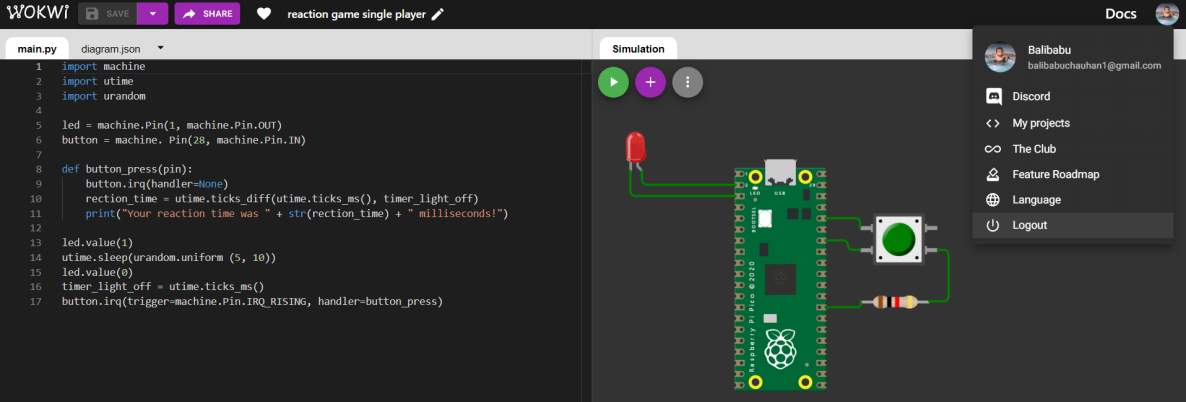


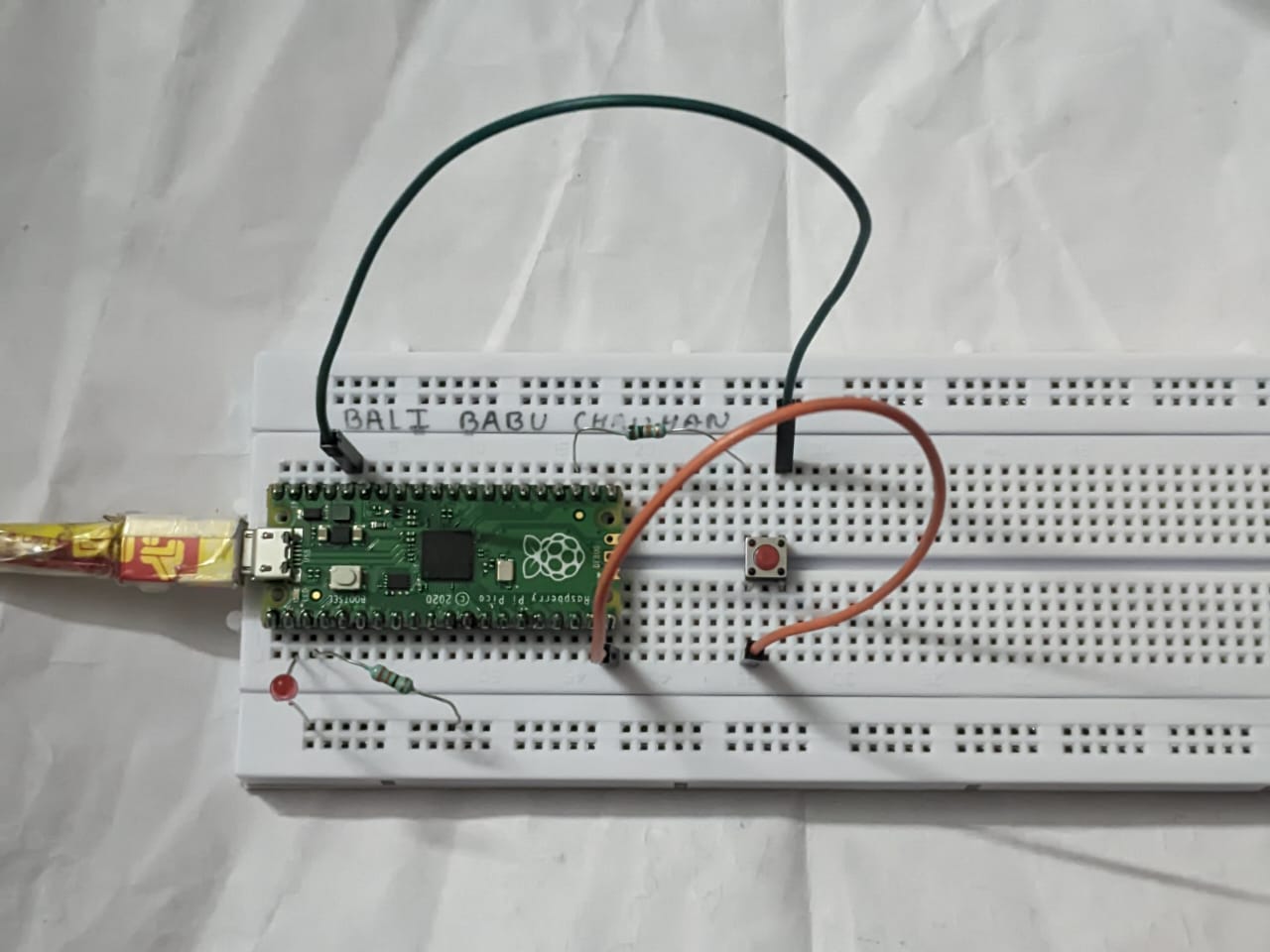
Figure 14: (Breadboard Schematic representation of an electronic circuit for implementation of Puffin crossing system.)

Figure 15: (Hardware implementation based electronic circuit for implementation of Puffin crossing system)

# Objective 4

****

(Figure 16: Simulation based electronic circuit for implementation of Single player based Simple Reaction Timing Game.)



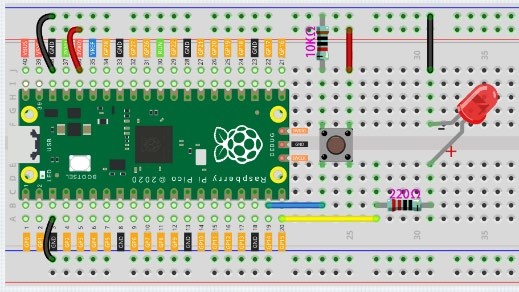
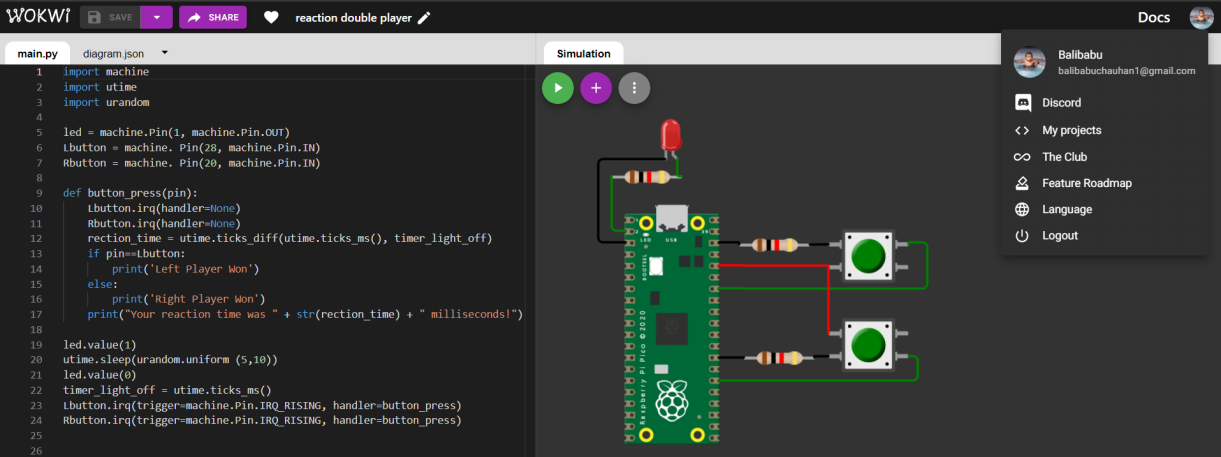


Figure 17: (Breadboard Schematic

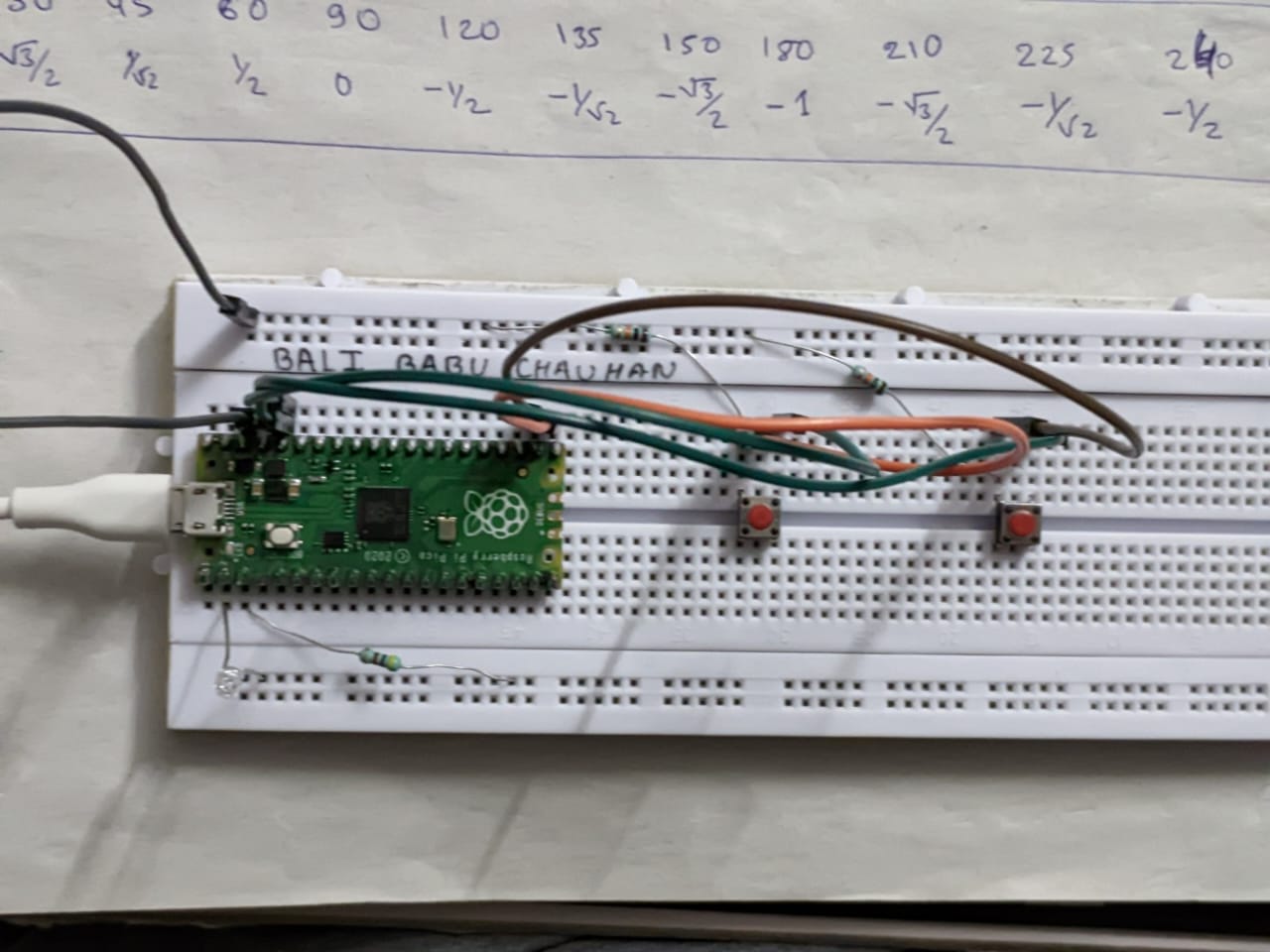
representation of an electronic circuit for implementation of Single player

based Simple Reaction Timing Game.) Figure 18: (Hardware implementation based electronic circuit for

implementation of Single player based Simple Reaction Timing Game)



(Figure 19: Simulation based electronic circuit for implementation of Two player based Simple Reaction Timing Game.)



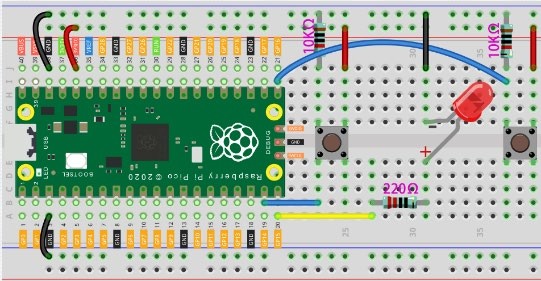
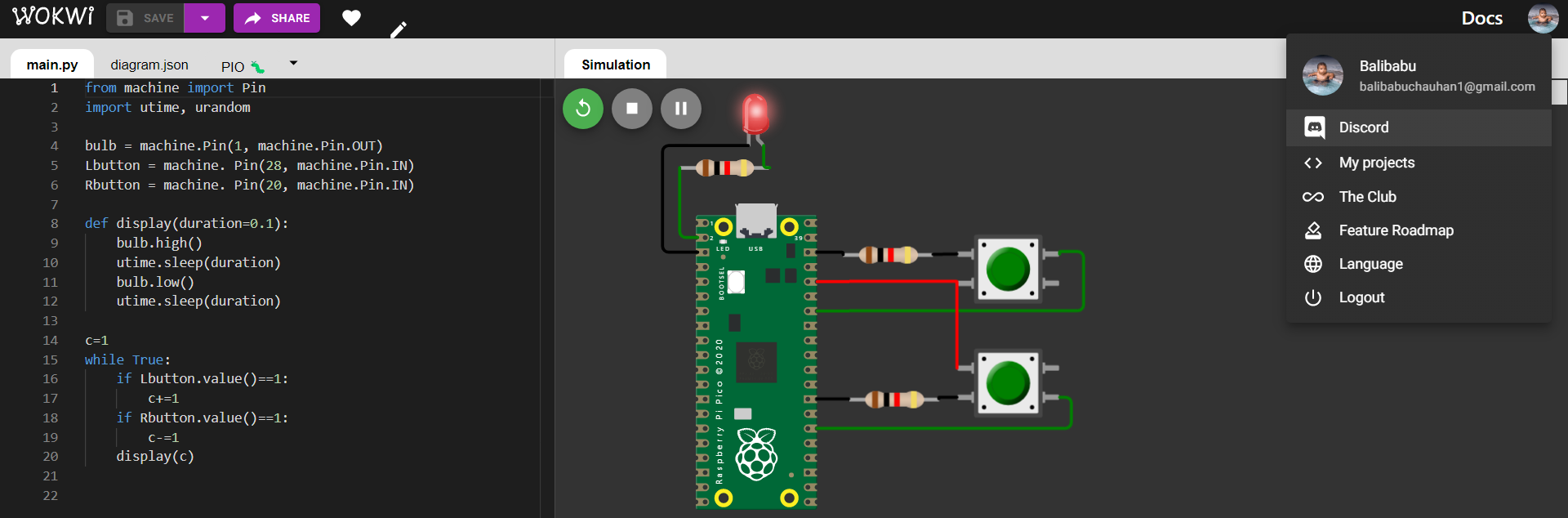


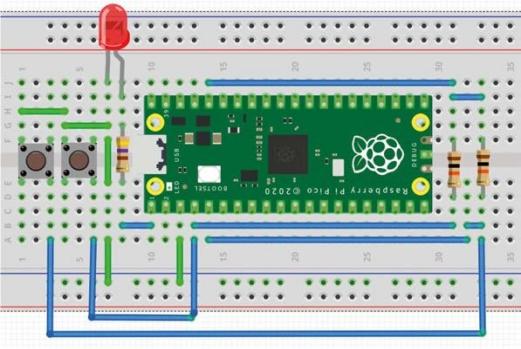
Figure 20: (Breadboard Schematic representation of an electronic circuit for implementation of Two player based Simple Reaction Timing Game.)

Figure 21: (Hardware implementation based electronic circuit for implementation of Two player based Simple Reaction Timing Game)

# Objective 6

****

(Figure 22: Simulation based electronic circuit for changing the LED flashing rate using pushbutton with Pull-up resistors.)



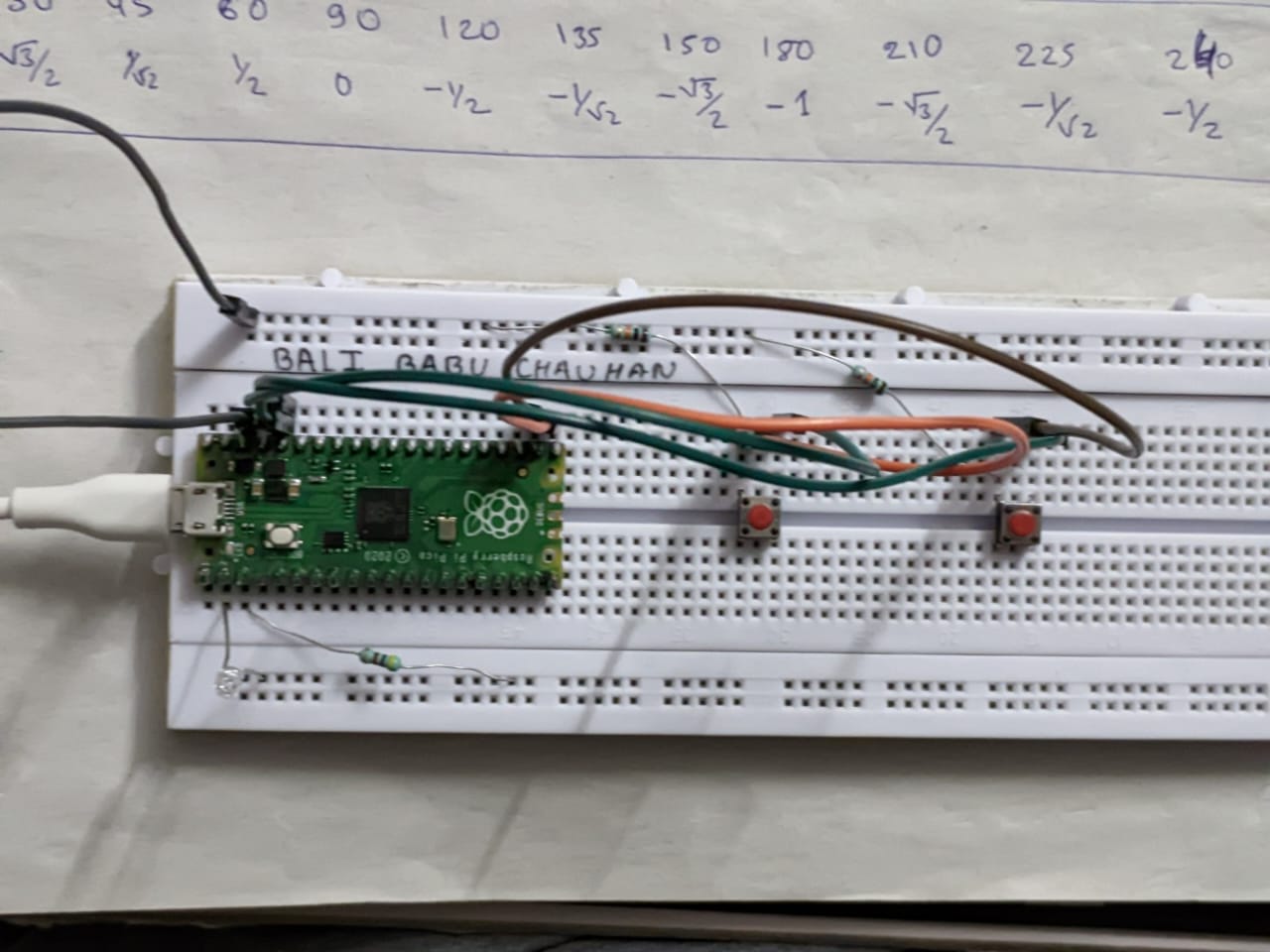
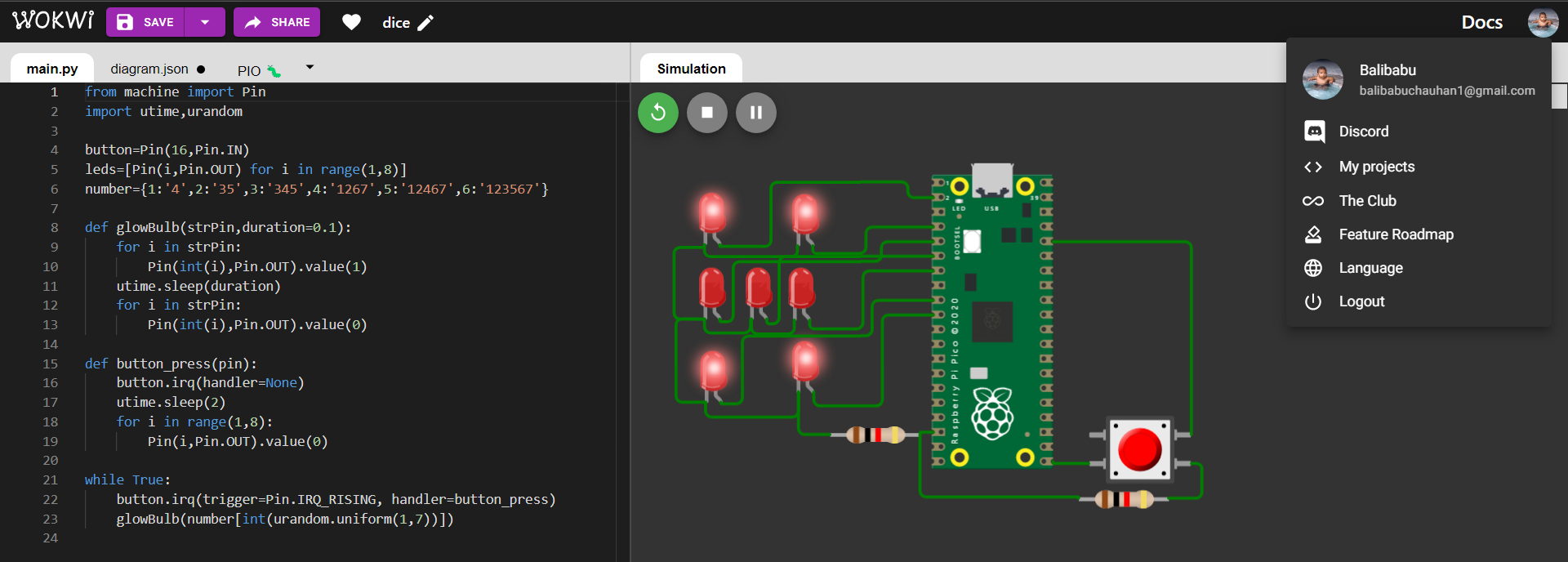


Figure 23: (Breadboard Schematic representation of an electronic circuit for changing the LED flashing rate using pushbutton with Pull-up resistors.)

Figure 24: (Hardware implementation based electronic circuit for changing the LED flashing rate using pushbutton with Pull-up resistors.)

# Objective 7

****

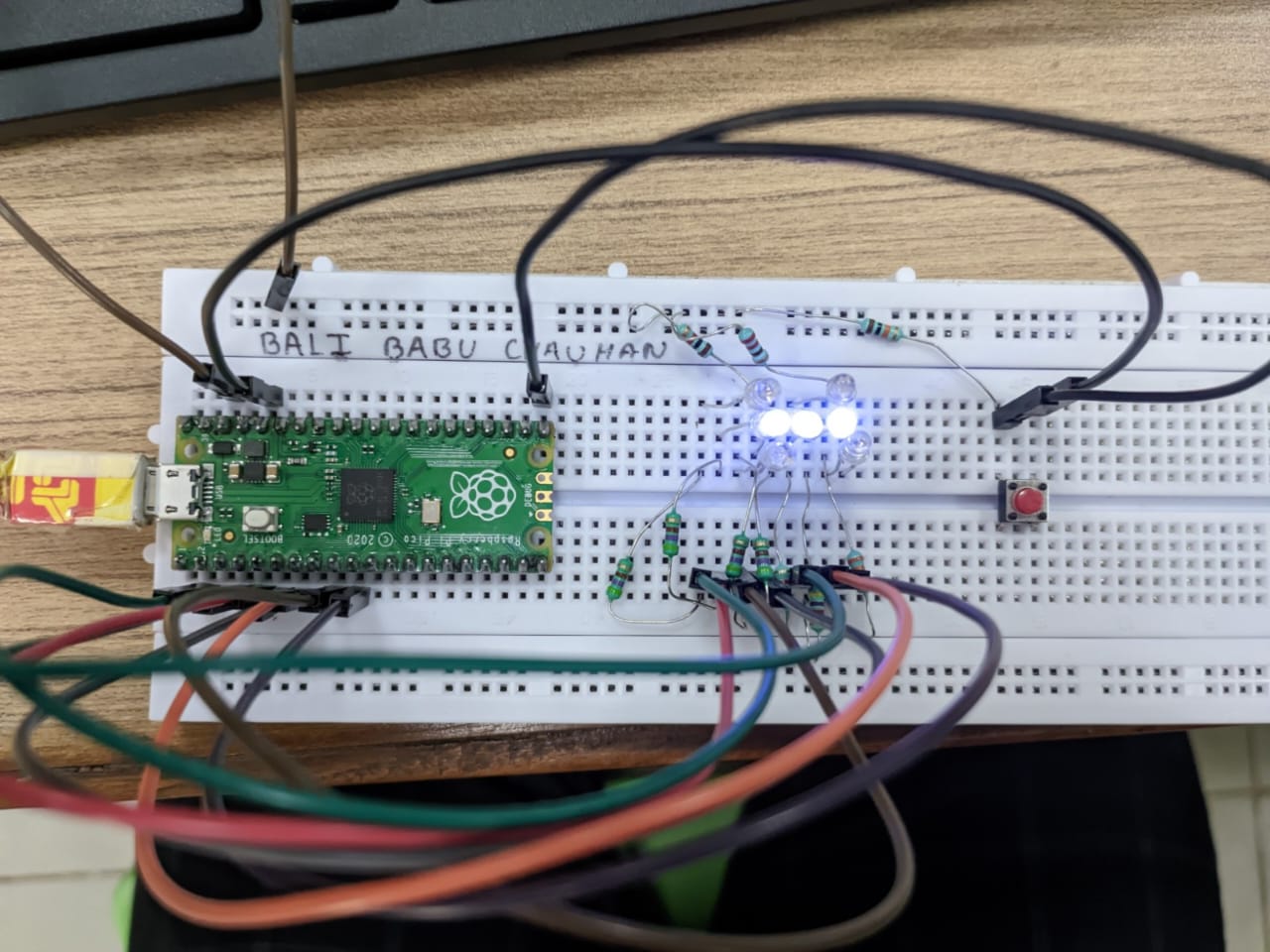
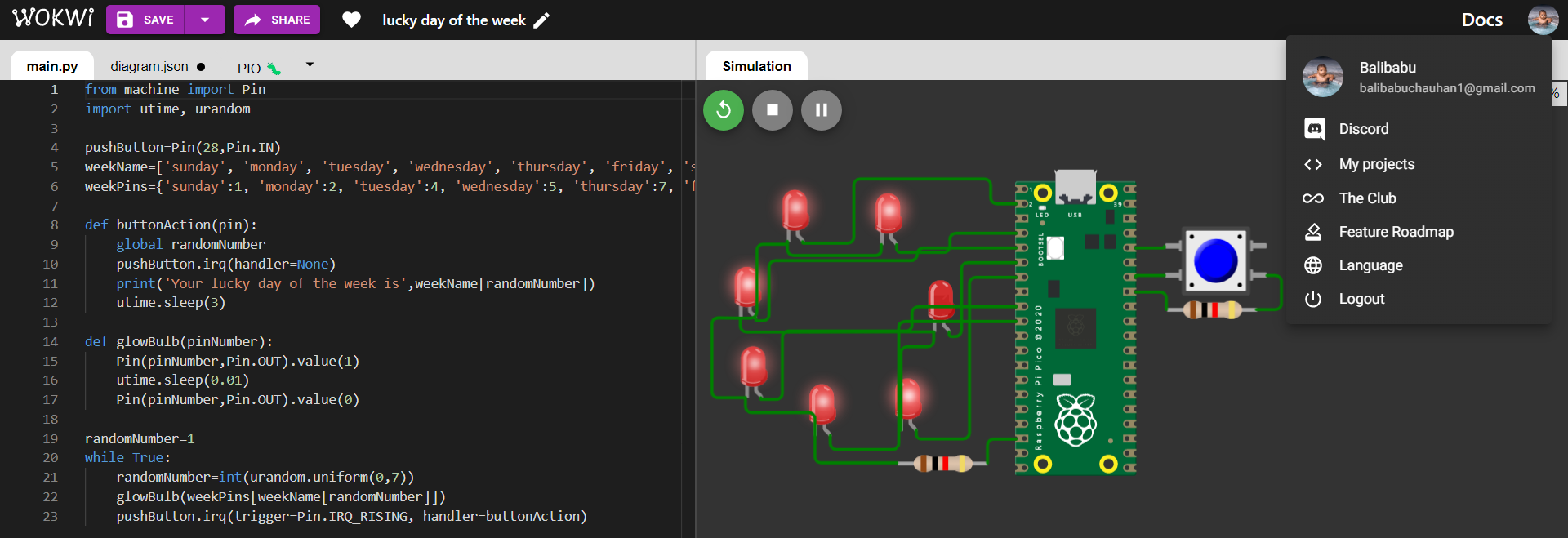


Figure 25: (Simulation based electronic circuit for implementation of a Digital Dice Game using LEDs and Push-buttons.)

Figure 26: (Hardware implementation based electronic circuit for implementation of a Digital Dice Game using LEDs and Push-buttons.)

# Objective 8

****

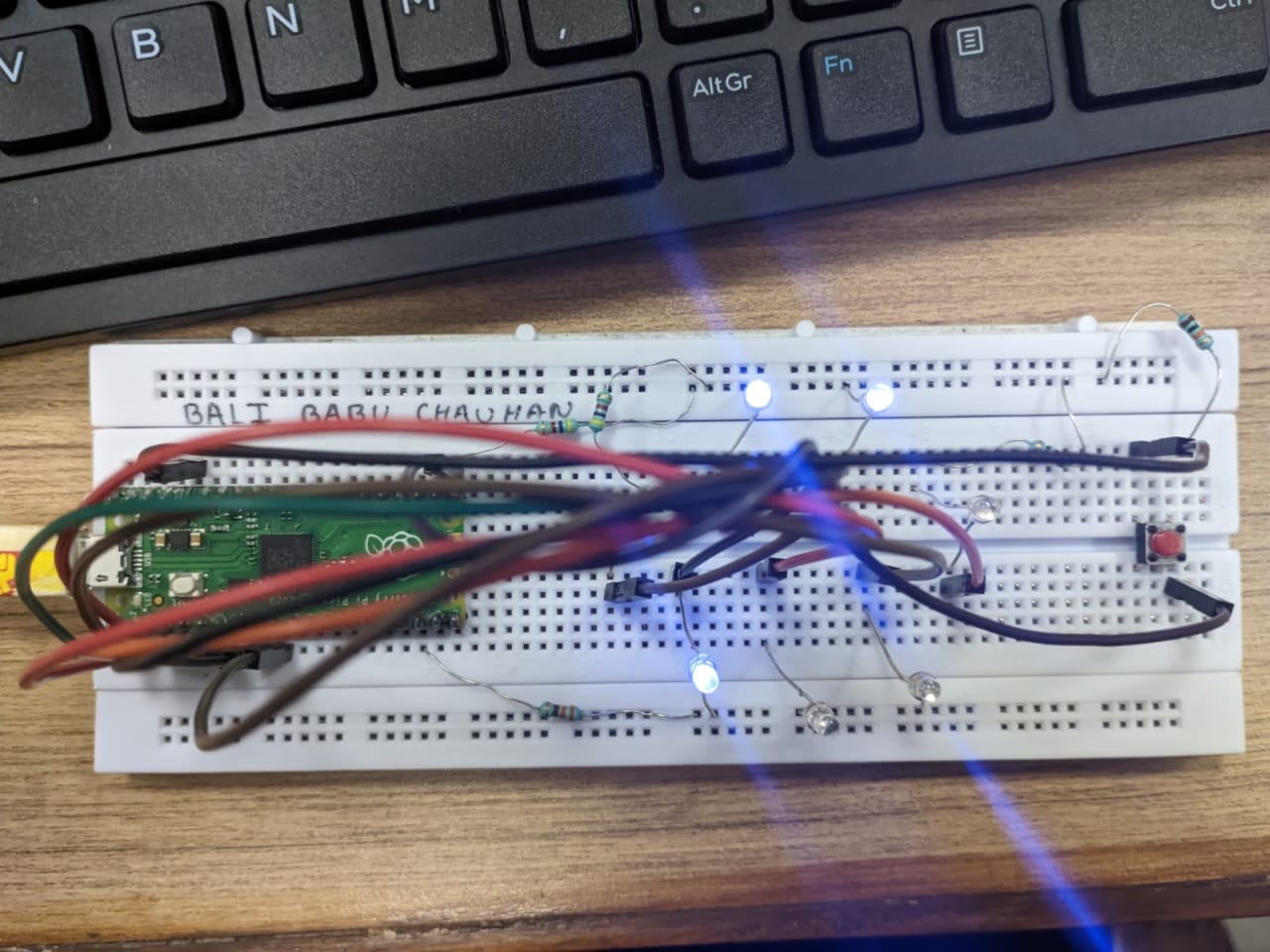


Figure 27: (Simulation based electronic circuit for Lucky Day of the Week using LEDs and Push-buttons.)

Figure 28: (Hardware implementation based electronic circuit for Lucky Day of the Week using LEDs and Push-buttons.)

***Codes:***

# Objective 1

print("Hello, Pi Pico!")

print("This is Experiment - 5 and Objective - 1")

print("Name:Bali Babu Chauhan ; Registration No.:19410121182 ") print("Objective : 1 Implementation of mini pedestrian crossing system using multiple LEDs without push-button ")

from machine import Pin

import utime

def glow(bulb,duration=0.1):

    bulb.value(1)

    utime.sleep(duration)

    bulb.value(0)

red=Pin(1,Pin.OUT)

yellow=Pin(2,Pin.OUT)

green=Pin(3,Pin.OUT)

durations=[5,2,7]

button=Pin(4,mode=Pin.IN, pull=Pin.PULL\_DOWN)

lights=[red,yellow,green]

while True:

    for light,duration in zip(lights,durations):

        c=0

        while c<duration\*10:

            c+=1

            glow(light)

            while button.value()==1:

                glow(red,duration=5)

# Objective 2

print("Objective : 2 Study the effect of Multithreading concept on processing time of an embedded system using Micro-python script ")

import utime,\_uthread

def square(lis):

    newList=[]

    for i in lis:

        newList.append(i\*i)

def cube(lis):

    newList=[]

    for i in lis:

        newList.append(i\*i\*i)

\_thread.start\_new\_thread(clickListener,(button,))

lis=[1,2,3,4,6,7]

t=utime.ticks\_us()

square(lis)

cube(lis)

t2=utime.ticks\_us()

print('time taken',(t2-t))

# Objective 3

print("Objective : 3 Implementation of Puffin Crossing System based on the concept of multi-threading using multiple LEDs with a push-button and a buzzer.")

from machine import Pin

import utime, \_thread

isPressed=0

def clickListener(button):

global isPressed

while True:

if button.value()==0: isPressed=0

else: isPressed=1

def glow(bulb,duration=0.1):

bulb.value(1)

utime.sleep(duration)

bulb.value(0)

red=Pin(1,Pin.OUT)

yellow=Pin(2,Pin.OUT)

green=Pin(3,Pin.OUT)

durations=[5,2,7]

button=Pin(4,mode=Pin.IN, pull=Pin.PULL\_DOWN)

lights=[red,yellow,green]

\_thread.start\_new\_thread(clickListener,(button,))

while True:

for light,duration in zip(lights,durations):

c=0

while c<duration\*10:

c+=1

glow(light)

while isPressed==1:

glow(red,duration=5)

**Objective 4**

print("Objective : 4 Implementation of Single player based Simple Reaction Timing Game based on the concept of Interrupt Requests, or IRQs using a push- button.”)

import machine, urandom, utime

led = machine.Pin(1, machine.Pin.OUT)

button = machine. Pin(28, machine.Pin.IN)

def button\_press(pin):

    button.irq(handler=None)

    rection\_time = utime.ticks\_diff(utime.ticks\_ms(), timer\_light\_off)

    print("Your reaction time was " + str(rection\_time) + " milliseconds!")

led.value(1)

utime.sleep(urandom.uniform (5, 10))

led.value(0)

timer\_light\_off = utime.ticks\_ms()

button.irq(trigger=machine.Pin.IRQ\_RISING, handler=button\_press)

# Objective 5

print("Objective : 5 Implementation of Two player based Simple Reaction Timing Game based on the concept of Interrupt Requests, or IRQs using a push-button.")

import machine, urandom, utime

led = machine.Pin(1, machine.Pin.OUT)

Lbutton = machine. Pin(28, machine.Pin.IN)

Rbutton = machine. Pin(20, machine.Pin.IN)

def button\_press(pin):

    Lbutton.irq(handler=None)

    Rbutton.irq(handler=None)

    rection\_time = utime.ticks\_diff(utime.ticks\_ms(), timer\_light\_off)

    if pin==Lbutton:

        print('Left Player Won')

    else:

        print('Right Player Won')

    print("Your reaction time was " + str(rection\_time) + " milliseconds!")

led.value(1)

utime.sleep(urandom.uniform (5,10))

led.value(0)

timer\_light\_off = utime.ticks\_ms()

Lbutton.irq(trigger=machine.Pin.IRQ\_RISING, handler=button\_press)

Rbutton.irq(trigger=machine.Pin.IRQ\_RISING, handler=button\_press)

# Objective 6

print("Objective : 6 Implementation of changing the LED flashing rate using pushbutton as external interrupts with external and internal Pull-up resistors.")

from machine import Pin

import utime, urandom

bulb = machine.Pin(1, machine.Pin.OUT)

Lbutton = machine. Pin(28, machine.Pin.IN)

Rbutton = machine. Pin(20, machine.Pin.IN)

def display(duration=0.1):

    bulb.high()

    utime.sleep(duration)

    bulb.low()

    utime.sleep(duration)

c=1

while True:

    if Lbutton.value()==1:

        c+=1

    if Rbutton.value()==1:

        c-=1

    display(c)

# Objective 7

print("Objective : 7 Implementation of a Digital Dice Game using LEDs and Push-buttons.")

from machine import Pin

import utime,urandom

button=Pin(16,Pin.IN)

leds=[Pin(i,Pin.OUT) for i in range(1,8)]

number={1:'4',2:'35',3:'345',4:'1267',5:'12467',6:'123567'}

def glowBulb(strPin,duration=0.1):

    for i in strPin:

        Pin(int(i),Pin.OUT).value(1)

    utime.sleep(duration)

    for i in strPin:

        Pin(int(i),Pin.OUT).value(0)

def button\_press(pin):

    button.irq(handler=None)

    utime.sleep(2)

    for i in range(1,8):

        Pin(i,Pin.OUT).value(0)

while True:

    button.irq(trigger=Pin.IRQ\_RISING, handler=button\_press)

glowBulb(number[int(urandom.uniform(1,7))])

# Objective 8

print("Objective : 8 Implementation of Lucky Day of the Week using LEDs and Push-buttons.")

from machine import Pin

import utime, urandom

pushButton=Pin(28,Pin.IN)

weekName=['sunday', 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday']

weekPins={'sunday':1, 'monday':2, 'tuesday':4, 'wednesday':5, 'thursday':7, 'friday':6, 'saturday':3}

def buttonAction(pin):

    global randomNumber

    pushButton.irq(handler=None)

    print('Your lucky day of the week is',weekName[randomNumber])

    utime.sleep(3)

def glowBulb(pinNumber):

    Pin(pinNumber,Pin.OUT).value(1)

    utime.sleep(0.01)

Pin(pinNumber,Pin.OUT).value(0)

randomNumber=1

while True:

    randomNumber=int(urandom.uniform(0,7))

    glowBulb(weekPins[weekName[randomNumber]])

    pushButton.irq(trigger=Pin.IRQ\_RISING, handler=buttonAction)

**Conclusion:**