POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Date:

Experiment – 6

Aim of the Experiment:

Familiarization with Pulse Width Modulation (PWM) using various sensors and actuators interfacing with Raspberry Pi and LEDs.

Objective:

1) Introduction to **Pulse Width Modulation (PWM)** Technique, its behaviour and applications.
2) Implementation of PWM on Micro-python with Raspberry Pi: **LED Breathing** using **Duty Cycle and Brightness Modification** of **inbuilt LED** of Raspberry Pi.
3) Implementation of PWM on Micro-python with Raspberry Pi: **LED Breathing** using **Duty Cycle and Brightness Modification** of **external LED**.
4) Implementation of **LED Breathing: Complete the challenge** with Minimum, Intermediate and Maximum Brightness Level.
5) Implementation of **Colorful Light: Alternately flashing red, green, and blue** colour without PWM technique and **Colorful Light: Mystery of additive color mixing** with PWM technique using Common Cathode type RGB LED and Raspberry Pi.
6) Implementation of **Custom Tone: Create a Melody** using active buzzer, transistor and Raspberry Pi.
7) Implementation of a **Custom Tone: Classic Happy Birthday Melody** using active buzzer, transistor and Raspberry Pi.

Components/Equipment/items Required:

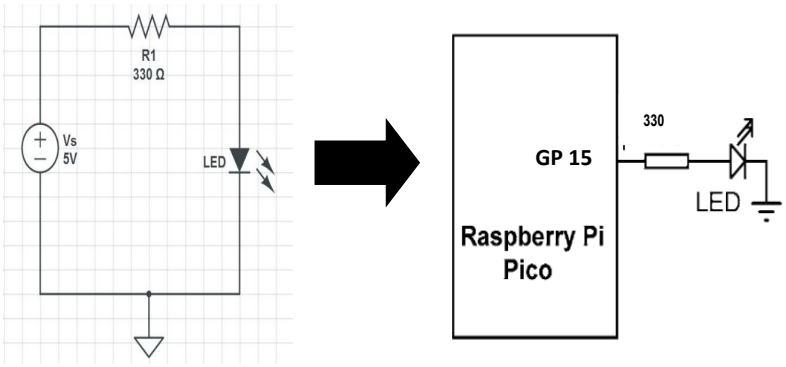
Sl No.	Name of the Component/Equipment	Specification	Quantity
1	Raspberry Pi Pico	RP2040 microcontroller chip, 125MHz	1
2	Raspberry Pi Pico cable	USB Type A to Micro-B	1
3	Resistors (carbon type)	¼ watt (330 Ω)	16
4	LED	3mm, Red	16
5	RGB LED	Common Cathode (CC)	1
6	Buzzer	Active	1
7	Transistor	NPN (2N2222)	1
8	Breadboard	840 Tie points	1
9	Jumper Wire	----------	As per requirement

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

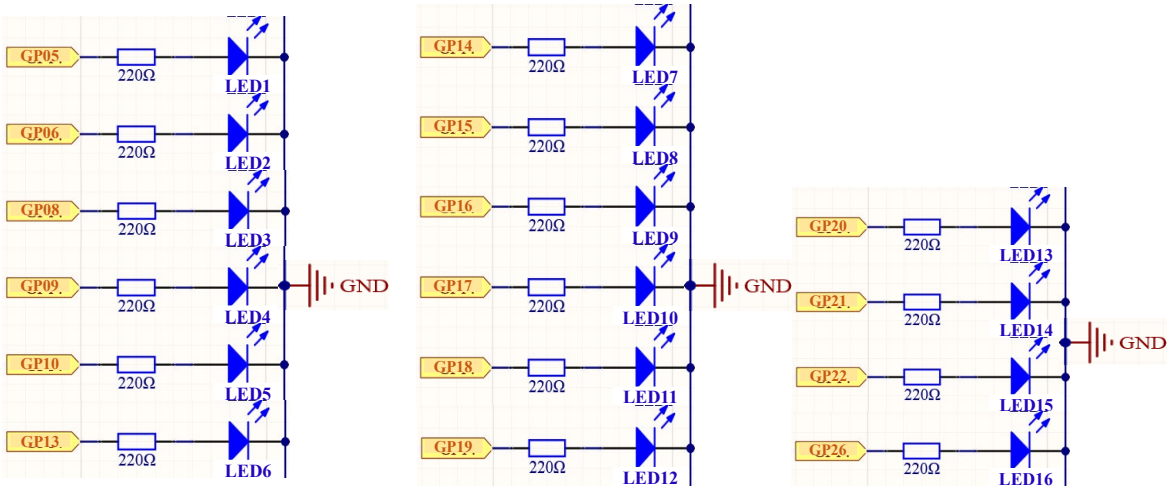
Circuit/Schematic Diagram:

Objective 3



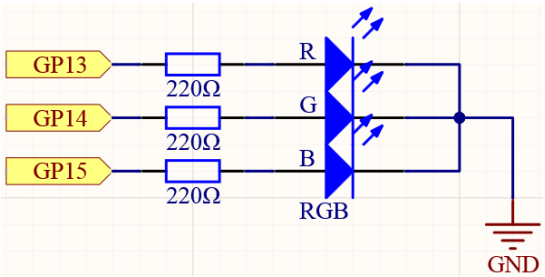
(Figure 1: Circuit diagram for implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of external LED.)

Objective 4



(Figure 2: Circuit diagram for implementation of LED Breathing: Complete the challenge with Minimum, Intermediate and Maximum Brightness Level.)

Objective 5

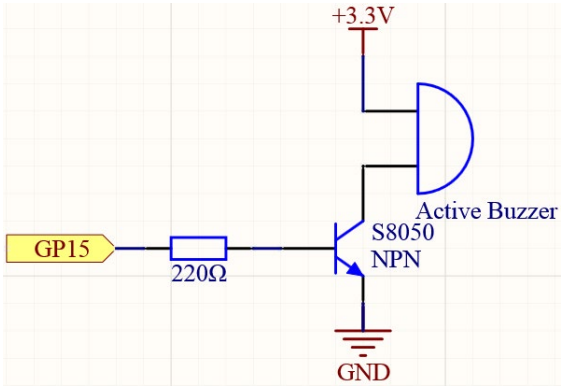


(Figure 3: Circuit diagram for implementation of Colorful Light: Alternately flashing red, green, and blue colour without PWM technique and Colorful Light: Mystery of additive color mixing with PWM technique using Common Cathode type RGB LED and Raspberry Pi.)

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

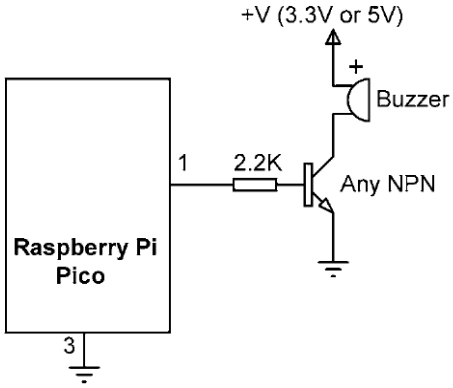
Experiment – 6

Objective 6



(Figure 4: Circuit diagram for implementation of Custom Tone: Create a Melody using active buzzer, transistor and Raspberry Pi.)

Objective 7



(Figure 5: Circuit diagram for implementation of a Custom Tone: Classic Happy Birthday Melody using active buzzer, transistor and Raspberry Pi.)

Observation:

Objective 2

(Figure 6: Simulation based electronic circuit for implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of inbuilt LED of Raspberry Pi.)

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Figure 7: (Hardware implementation based electronic circuit for implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of inbuilt LED of Raspberry Pi)

Objective 3

(Figure 8: Simulation based electronic circuit for implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of external LED)

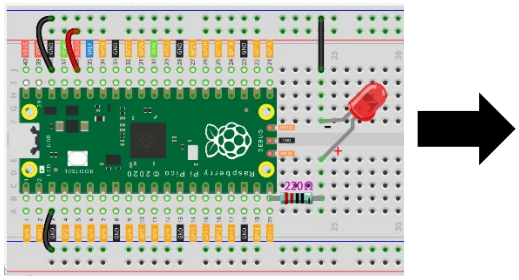


Figure 9: (Breadboard Schematic representation of an electronic circuit for implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of external LED.)

Figure 10: (Hardware implementation based electronic circuit for implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of external LED)

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Objective 4

Figure 11: (Simulation based LED Breathing: Complete the challenge with Minimum, Intermediate and Maximum Brightness Level.)

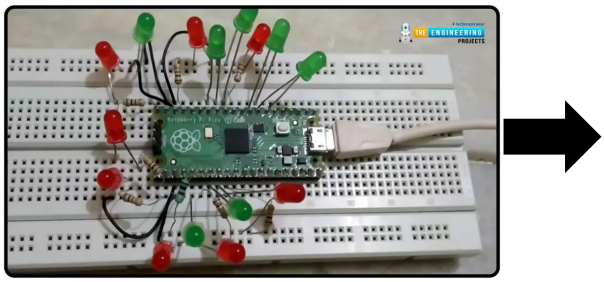


Figure 12: (Breadboard Schematic representation of an electronic circuit for implementation of LED Breathing: Complete the challenge with Minimum, Intermediate and Maximum Brightness Level.)

Figure 13: (Hardware implementation based electronic circuit for implementation of LED Breathing: Complete the challenge with Minimum Brightness Level.)

Figure 14: (Hardware implementation based electronic circuit for implementation of LED Breathing: Complete the challenge with Intermediate Brightness Level.)

Figure 15: (Hardware implementation based electronic circuit for implementation of LED Breathing: Complete the challenge with Maximum Brightness Level.)

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Objective 5

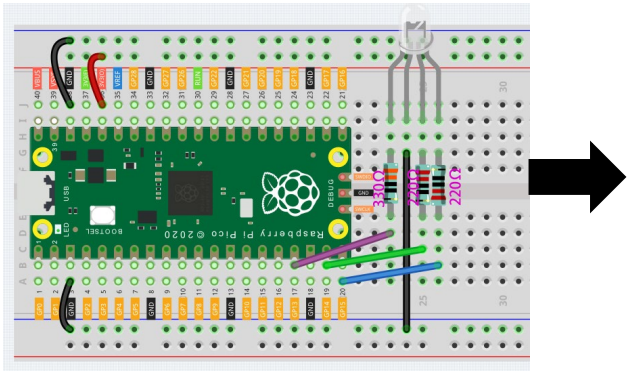


Figure 16: (Breadboard Schematic representation of an electronic circuit for implementation of LED Breathing: Complete the challenge with Minimum, Intermediate and Maximum Brightness Level.)

(Figure 17: Hardware implementation based electronic circuit for Colorful Light: Alternately flashing red, green, and blue colour without PWM technique using Common Cathode type RGB LED and Raspberry Pi)

(Figure 18: Hardware implementation based electronic circuit for Colorful Light: Mystery of additive color mixing with PWM technique using Common Cathode type RGB LED and Raspberry Pi.)

Objective 6

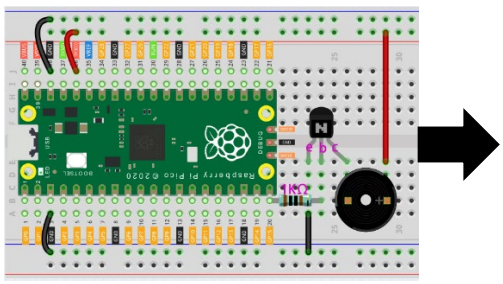


Figure 19: (Breadboard Schematic representation of an electronic circuit for implementation of Custom Tone: Create a Melody using active buzzer, transistor and Raspberry Pi.)

Figure 20: (Hardware implementation based electronic circuit for implementation of Custom Tone: Create a Melody using active buzzer, transistor and Raspberry Pi)

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Objective 7

Figure 21: (Hardware implementation based electronic circuit for implementation of a Custom Tone: Classic Happy Birthday Melody using active buzzer, transistor and Raspberry Pi)

Codes:

Objective 2

```
print("Hello, Pi Pico!")
print("This is Experiment - 6 and Objective - 2")
print("Name:                          ; Registration No.:                ")
print("Objective : 2  Implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of inbuilt LED of Raspberry Pi.")
```

Objective 3

```
print("Hello, Pi Pico!")
print("This is Experiment - 6 and Objective - 3")
print("Name:                          ; Registration No.:                ")
print("Objective : 3   Implementation of PWM on Micro-python with Raspberry Pi: LED Breathing using Duty Cycle and Brightness Modification of external LED.")
```

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Objective 4

```
print("Hello, Pi Pico!")
print("This is Experiment - 6 and Objective - 4")
print("Name:                          ; Registration No.:                ")
print("Objective : 4   Implementation of LED Breathing: Complete the challenge
with Minimum, Intermediate and Maximum Brightness Level.")
```

Objective 5

```
print("Hello, Pi Pico!")
print("This is Experiment - 6 and Objective - 5")
print("Name:                          ; Registration No.:                ")
print("Objective : 5   Implementation of Colorful Light: Alternately flashing
red, green, and blue colour without PWM technique and Colorful Light: Mystery
of additive color mixing with PWM technique using Common Cathode type RGB LED
and Raspberry Pi.”)
```

Objective 6

```
print("Hello, Pi Pico!")
print("This is Experiment - 6 and Objective - 6")
print("Name:                          ; Registration No.:                ")
print("Objective : 6    Implementation of Custom Tone: Create a Melody using
active buzzer, transistor and Raspberry Pi.")
```

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Objective 7

```
print("Hello, Pi Pico!")
print("This is Experiment - 6 and Objective - 7")
print("Name:                          ; Registration No.:                ")
print("Objective : 7   Implementation of a Custom Tone: Classic Happy Birthday
Melody using active buzzer, transistor and Raspberry Pi.")
```

Conclusion:

Precautions:

POST-LAB IOT Projects using Python (CSE 4110) ASSIGNMENT

Experiment – 6

Post Experiment Questionnaire:

Answer all the Questions in brief with some appropriate examples.

1) Write a Micro-Python code which turns on the buzzer, connected to GPIO16 of the Raspberry Pi Pico by PWM signal having 200 Hz frequency and 20% duty cycle by pressing the push button connected to GPIO21.
2) A PWM signal with a frequency of 4kHz is to be generated on the Raspberry Pi Pico using MicroPython. What value should be passed to the freq() function to set this frequency?
3) A PWM signal with a duty cycle of 75% is to be generated on the Raspberry Pi Pico using MicroPython. What value should be passed to the duty() function to set this duty cycle?
4) A PWM signal with a frequency of 2kHz is currently being generated on a specific channel of the Raspberry Pi Pico using MicroPython. The frequency needs to be changed to 1kHz. What value should be passed to the freq() function to set this new frequency?
5) A PWM signal with a duty cycle of 50% is currently being generated on a specific channel of the Raspberry Pi Pico using MicroPython. The duty cycle needs to be changed to 25%. What value should be passed to the duty() function to set this new duty cycle?
6) A software PWM signal with a frequency of 1kHz and a duty cycle of 50% is to be generated on a specific digital output pin of the Raspberry Pi Pico using MicroPython. How many times per second should the digital output pin be toggled to achieve this?
7) An input PWM signal with a frequency of 3kHz and a duty cycle of 33% is being received on a specific digital input pin of the Raspberry Pi Pico using MicroPython. What is the width of the input pulse in microseconds?
8) A PWM signal with a frequency of 2MHz is to be generated on the Raspberry Pi Pico using MicroPython. What value should be passed to the freq() function to set this frequency?
9) A PWM signal with a duty cycle of 90% is to be generated on the Raspberry Pi Pico using MicroPython. What value should be passed to the duty() function to set this duty cycle?
10) A PWM signal with a frequency of 1kHz is currently being generated on a specific channel of the Raspberry Pi Pico using MicroPython. The frequency needs to be

changed to 500Hz. What value should be passed to the freq() function to set this new frequency?

11) A PWM signal with a duty cycle of 20% is currently being generated on a specific channel of the Raspberry Pi Pico using MicroPython. The duty cycle needs to be changed to 10%. What value should be passed to the duty() function to set this new duty cycle?

12) A software PWM signal with a frequency of 5kHz and a duty cycle of 25% is to be generated on a specific digital output pin of the Raspberry Pi Pico using MicroPython. How many times per second should the digital output pin be toggled to achieve this?

13) An input PWM signal with a frequency of 1kHz and a duty cycle of 50% is being received on a specific digital input pin of the Raspberry Pi Pico using MicroPython. What is the width of the input pulse in microseconds?

Name of the Student
Registration No
Semester
Branch, Section