SCAD COLLEGE OF ENGINEERING & TECHNOLOGY

Recognized under Section 2(f) & 12(B) of the UGC Act, 1956

Approved by AICTE Affiliated to Anna University, Chennai

Cheranmahadevi, Tirunelveli - 627414

CS3691-Embedded System and IoT

LAB RECORD

NAME: ….....

REGISTER NUMBER: ….....….

DEGREE&BRANCH: .....….

YEAR/SEMESTER: ….....……

SCAD COLLEGE OF ENGINEERING & TECHNOLOGY

Recognized under Section 2(f) & 12(B) of the UGC Act, 1956

Approved by AICTE Affiliated to Anna University, Chennai

Cheranmahadevi, Tirunelveli - 627414

Certified that this is a Bonafide record work done

By Selvan/Selvi..... with Register No ..... studying in III-year VI semester in Computer Science and Engineering branch of this Institution during the academic year 2024-2025 [EVEN SEM]

Staff in-charge

Head of the Department

Submitted for the Anna University practical examination held at SCAD College of Engineering and Technology, Cherranmahadevi on ……

Internal Examiner

External Examiner

INDEX

Ex. No.	Date	Name of the Experiment	Mark	Sign
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				

EXP.NO: 1	8051 ASSEMBLY LANGUAGE EXPERIMENTS USING SIMULATOR
DATE:	

AIM :

APPARTUS REQUIRED :

Keil µ software

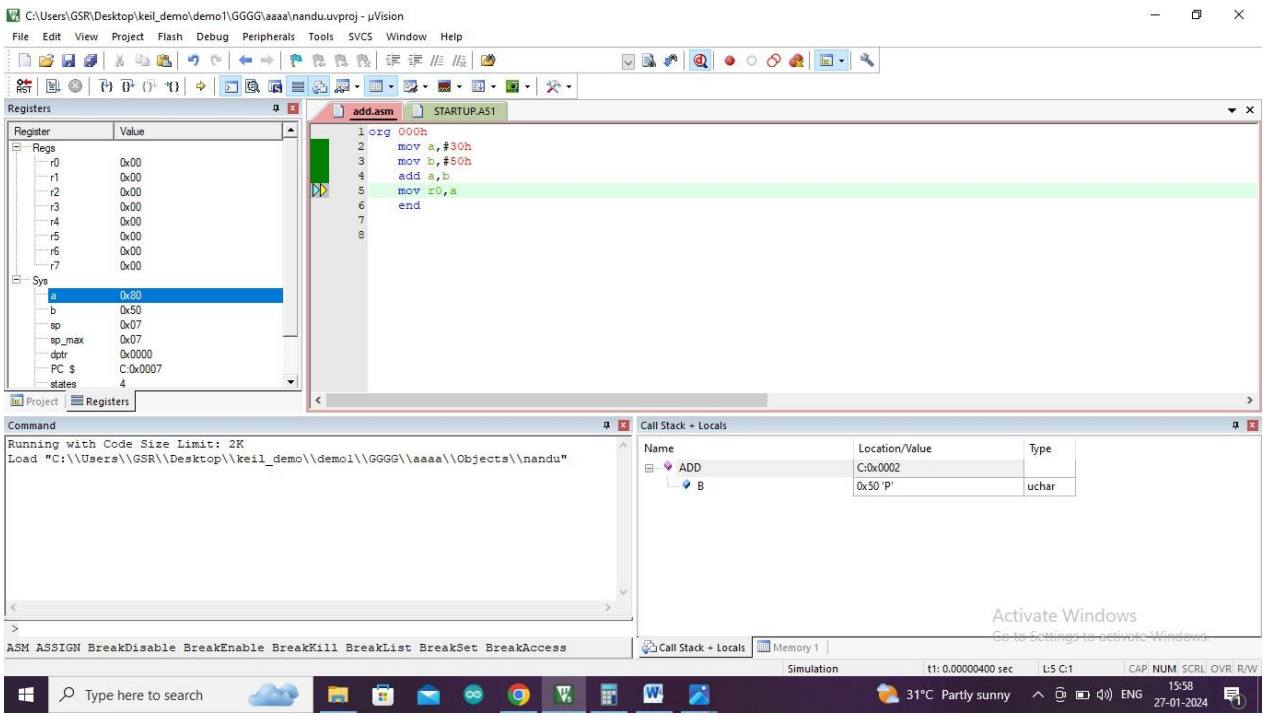
PRODURCE:

1. The 8051 microcontroller has a set of instructions for arithmetic, logic, and data manipulation operations. that demonstrates some basic ALU operations using assembly language for the 8051 microcontroller.
2. The various ALU operations, including addition, subtraction, multiplication, division, logical AND/OR/XOR, and rotation. Note that the specific instructions and registers used may vary depending on the exact 8051 microcontroller variant you are working with.
3. Always refer to the datasheet or reference manual for your specific microcontroller to ensure accurate programming.

PROGRAM :

```
org 000h
        mov a,#30h
        mov b,#50h
        add a,b
        mov r0,a
        end
```

OUTPUT:



RESULT:

EXP.NO: 2	TEST DATA TRANSFER BETWEEN REGISTER AND MEMORY
DATE:	

AIM:

APPARTUS REQUIRED :

Keil µ software

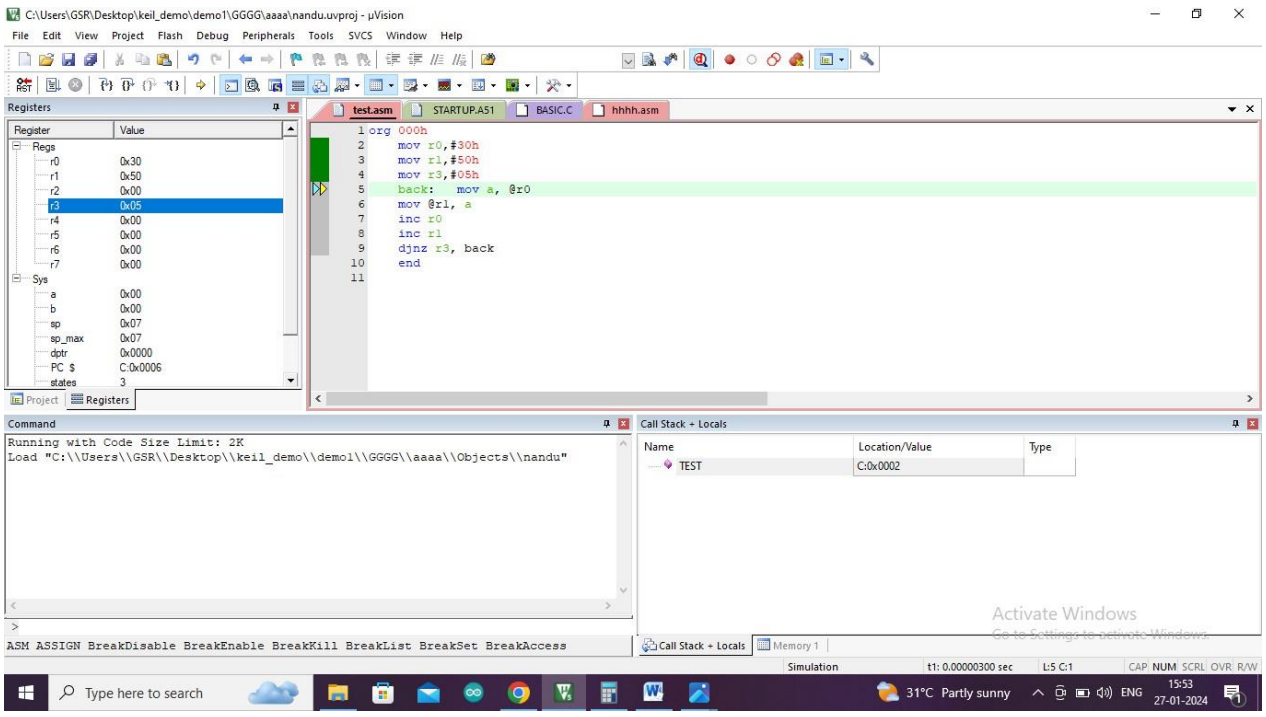
PRODURCE:

1. The 8051 microcontroller has a set of instructions for arithmetic, logic, and data manipulation operations. that demonstrates some basic ALU operations using assembly language for the 8051 microcontroller.
2. The various ALU operations, including addition, subtraction, multiplication, division, logical AND/OR/XOR, and rotation. Note that the specific instructions and registers used may vary depending on the exact 8051 microcontroller variant you are working with.
3. Always refer to the datasheet or reference manual for your specific microcontroller to ensure accurate programming.

PROGRAM :

```
org 000h
        mov r0,#30h
        mov r1,#50h
        mov r3,#05h
        back:    mov a, @r0
        mov @r1, a
        inc r0
        inc r1
        djnz r3, back
        end
```

OUTPUT:



RESULT:

EXP.NO: 3	PERFORM ALC OPERATIONS
DATE:	

AIM :

TO PERFORM ALC OPERATIONS

APPARTUS REQUIRED :

Keil µ software

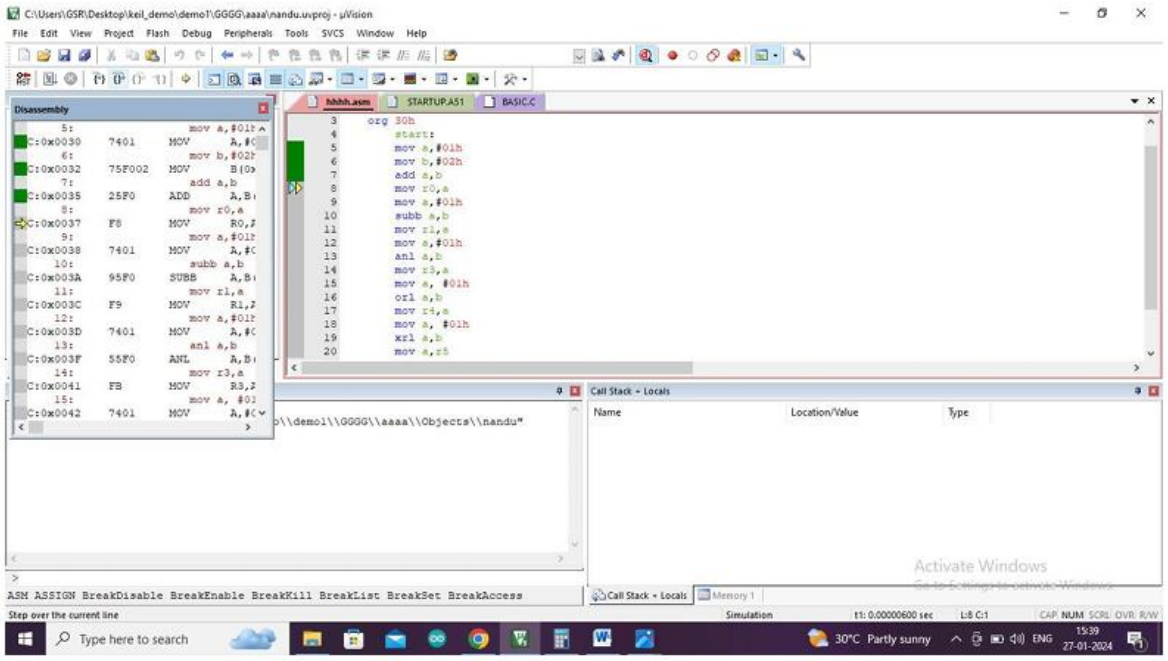
PRODURCE:

1. The 8051 microcontroller has a set of instructions for arithmetic, logic, and data manipulation operations. that demonstrates some basic ALU operations using assembly language for the 8051 microcontroller.
2. The various ALU operations, including addition, subtraction, multiplication, division, logical AND/OR/XOR, and rotation. Note that the specific instructions and registers used may vary depending on the exact 8051 microcontroller variant you are working with.
3. Always refer to the datasheet or reference manual for your specific microcontroller to ensure accurate programming.

PROGRAM :

```
org 00h
        sjmp start
        org 30h
                start:
                mov a,#01h
                mov b,#02h
                add a,b
                mov r0,a
                mov a,#01h
                subb a,b
                mov r1,a
                mov a,#01h
                anl a,b
                mov r3,a
                mov a, #01h
                orl a,b
                mov r4,a
                mov a, #01h
                xrl a,b
                mov a,r5

                end
```

OUTPUT:



RESULT

EXP.NO: 4	WRITE BASIC AND ARITHMETIC PROGRAM USING EMBEDDED C
DATE:	

AIM :

TO WRITE BASIC AND ARITHMETIC PROGRAM USING EMBEDDED C

APPARTUS REQUIRED :

Keil µ software

PRODURCE:

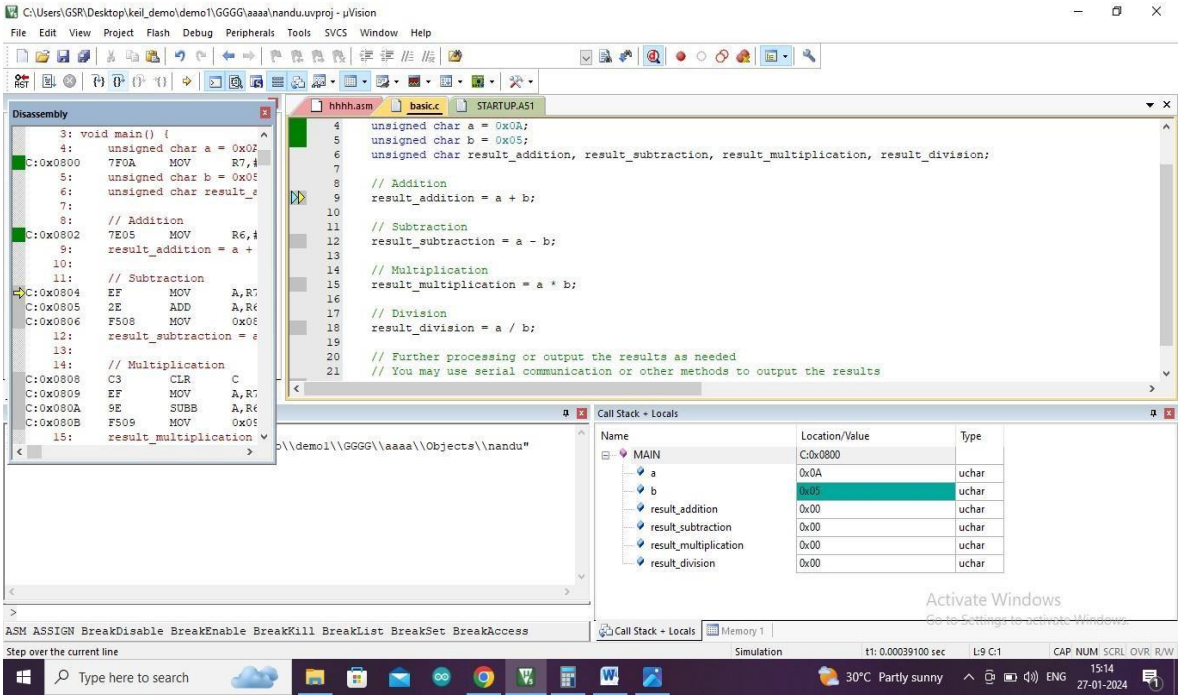
1. The 8051 microcontroller has a set of instructions for arithmetic, logic, and data manipulation operations. that demonstrates some basic ALU operations using assembly language for the 8051 microcontroller.
2. The various ALU operations, including addition, subtraction, multiplication, division, logical AND/OR/XOR, and rotation. Note that the specific instructions and registers used may vary depending on the exact 8051 microcontroller variant you are working with.
3. Always refer to the datasheet or reference manual for your specific microcontroller to ensure accurate programming.
4.

PROGRAM :

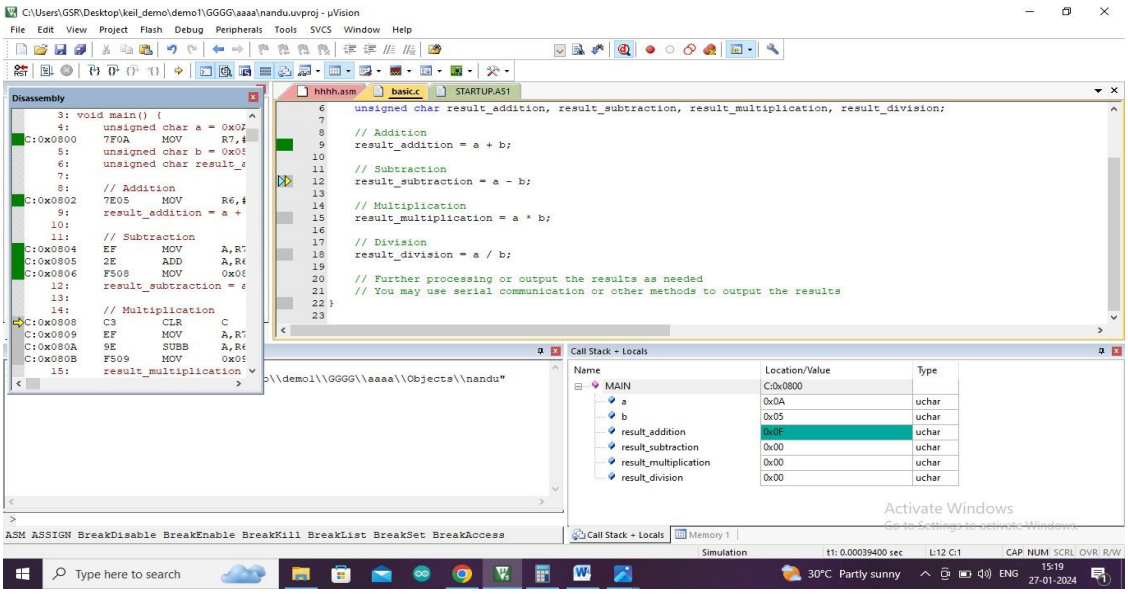
```
#include <8051.h>
void main() {
   unsigned char a = 0x0A;
   unsigned char b = 0x05;
   unsigned char result_addition, result_subtraction, result_multiplication, result_division;

   // Addition
   result_addition = a + b;

   // Subtraction
   result_subtraction = a - b;

   // Multiplication
   result_multiplication = a * b;

   // Division
   result_division = a / b;

   // Further processing or output the results as needed
   // You may use serial communication or other methods to output the results
}
```

OUTPUT:

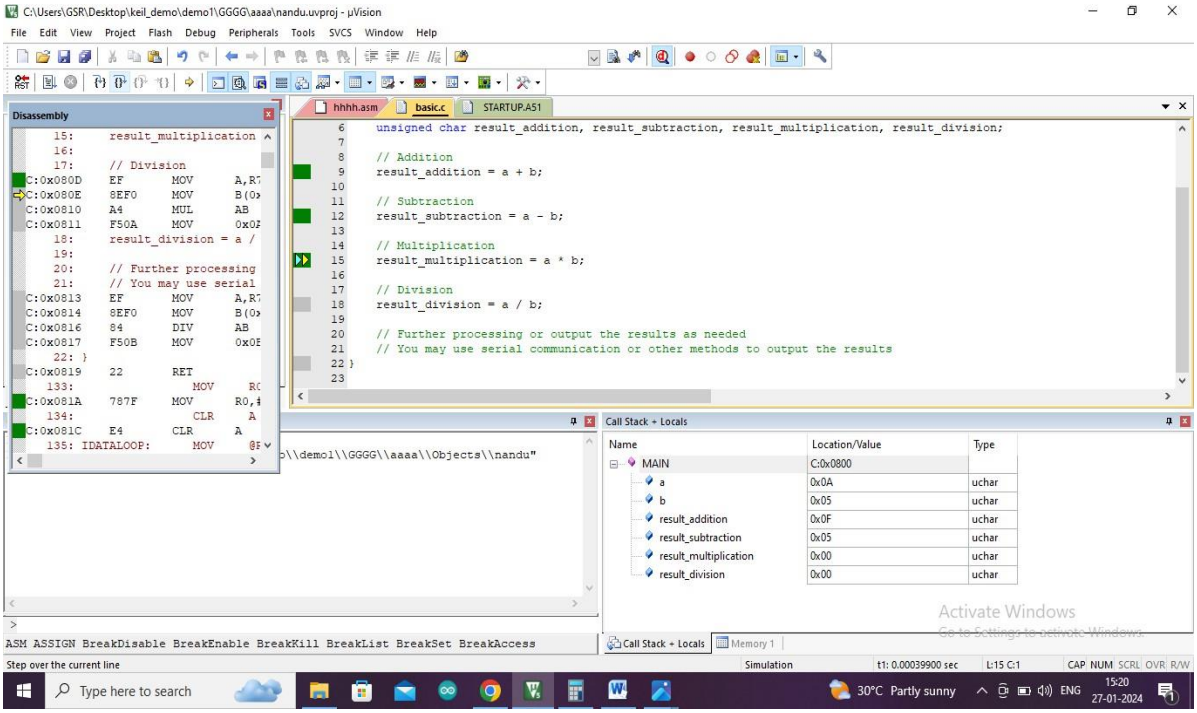
ADDITION :



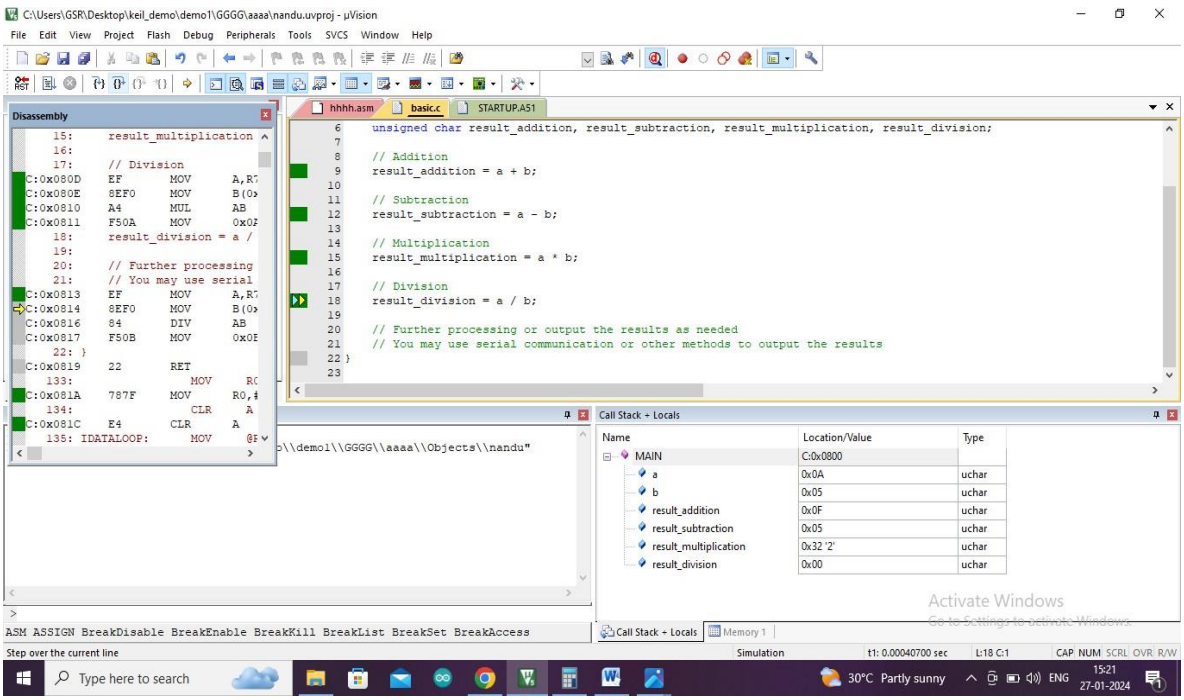
SUBTRACTION :



MULTIPLICATION :



DIVISION :



RESULT:

EXP.NO: 5	INTRODUCTION TO ARDUINO PLATFORM AND PROGRAMMING
DATE:	

AIM：

Objective:

- The objective of this lab experiment is to interface a buzzer with an Arduino and create a simple program to control the buzzer for different sounds and tones.

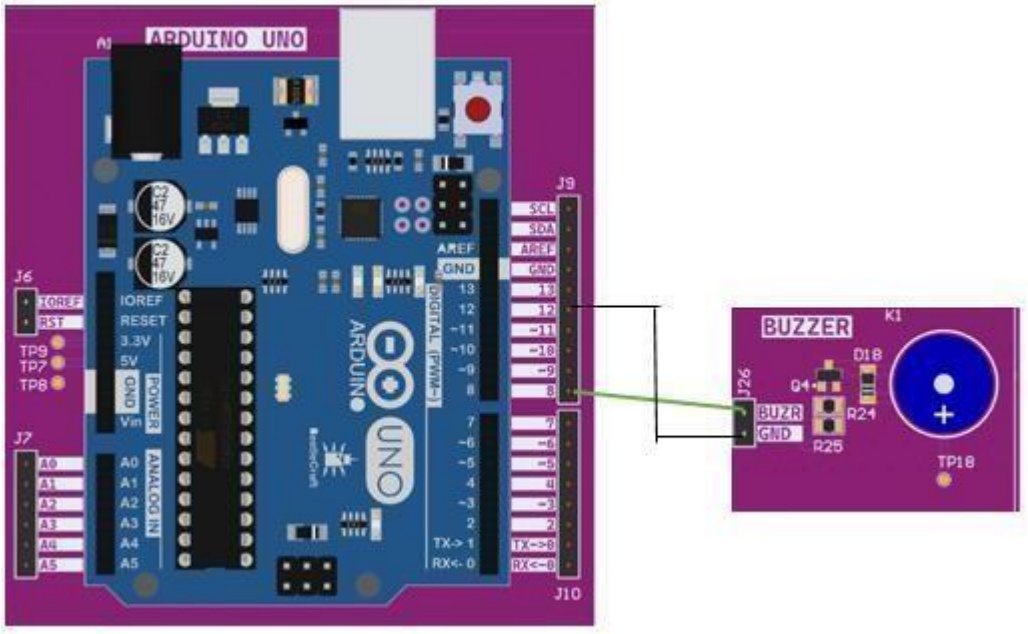
Equipment and Components:

- thingZkit IoT Board
- Jumper wires
- USB cable for Arduino Uno.
- Power Adapter Cable(12V/3A) for thingZkit IoT
- Computer with Arduino IDE installed.

Theory:

- A buzzer is an electronic component that produces sound when an electrical signal is applied to it. It is commonly used in various electronic devices and projects for auditory feedback, alarms, and signaling purposes.

Connection Diagram:



Pin Details:

thingZkit IoT-ARDUINO	Buzzer
8	BUZR
GND	GND

Procedure:

Setup Hardware:

- Connect your thingZkit IoT to your computer using the USB cable.
- Connect pin 8 of the Arduino to the onboard Buzzer.
- Ensure that the connections are secure and free from any short circuits.

Setup Software:

- Power on the thingZkit IoT board
- Open the Experiment 6 program in Arduino platform from the given source codefolder

Upload Code:

- Click the "Upload" button in the Arduino IDE to upload the code to your Arduino board.
  Note: Uploading or Flashing Procedure refer Annexure -I

Test the Experiment:

- Once the code is uploaded, observe the different sounds produced by the buzzer.

Result:

EXP.NO: 6	EXPLORE DIFFERENT COMMUNICATION METHODSWITH IOT DEVICES
DATE:	

AIM:

Objective:

- The objective of this hands-on lab experiment is to demonstrate the process of interfacing a Bluetooth module (specifically HC-05) with an Arduino board to wirelessly control the state of an LED. This experiment provides practical experience in combining hardware components and programming to create a Bluetooth-controlled LED system.

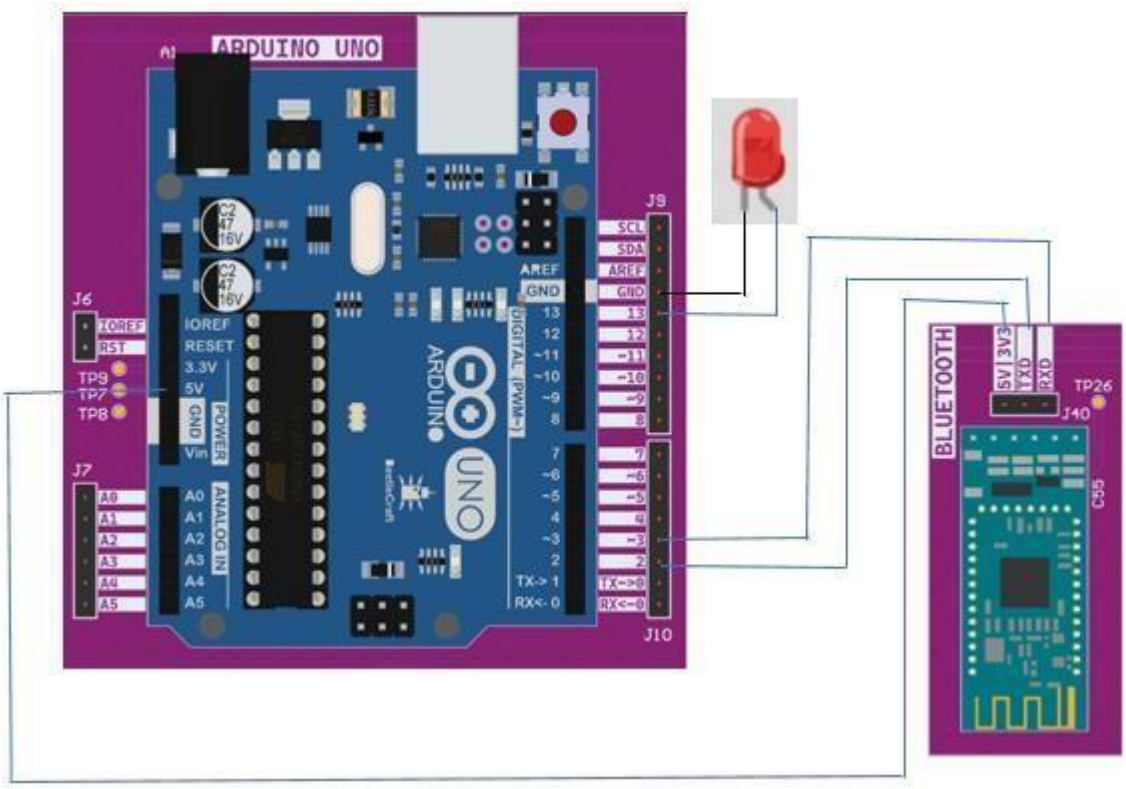
Equipment and Components:

- thingZkit IoT Board
- Jumper wires
- USB cable for Arduino Uno.
- Power Adapter Cable(12V/3A) for thingZkit IoT
- Computer with Arduino IDE installed.

Theory:

- Bluetooth is a wireless communication standard designed for short-range communication between devices. Bluetooth modules, such as the HC-05 and HC-06, are widely used in electronics projects to enable wireless communication between microcontrollers, like Arduino, and other devices like smartphones, tablets, or other Bluetooth-enabled gadgets.

Connection Diagram:



Pin Details:

thingZkit IoT-ARDUINO	Bluetooth Module & LED
Pin 2	TX
Pin 3	RX
5V	5V/3V
Pin 13	External LED (Long leg terminal)
GND	LED Short leg terminal

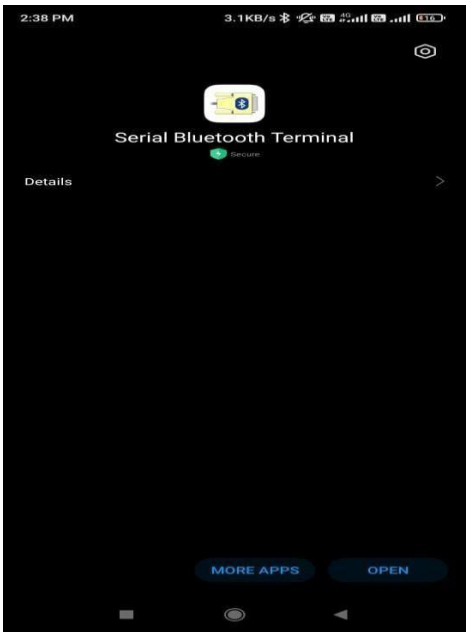
Procedure:

Setup Hardware:

- Connect your Arduino to your computer using the USB cable.
- Connect the Inbuilt Bluetooth Module with Arduino as described above
- Place the LED on the breadboard and connect it to the Arduino as described above.
- Ensure that all connections are secure and free from short circuits.

Setup Software:

- Power on the thingZkit IoT board
- Open the Experiment 15 program in Arduino platform from the given source codefolder
- Install the **mobile App** from the **Google Play Store,**

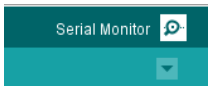


Upload Code:

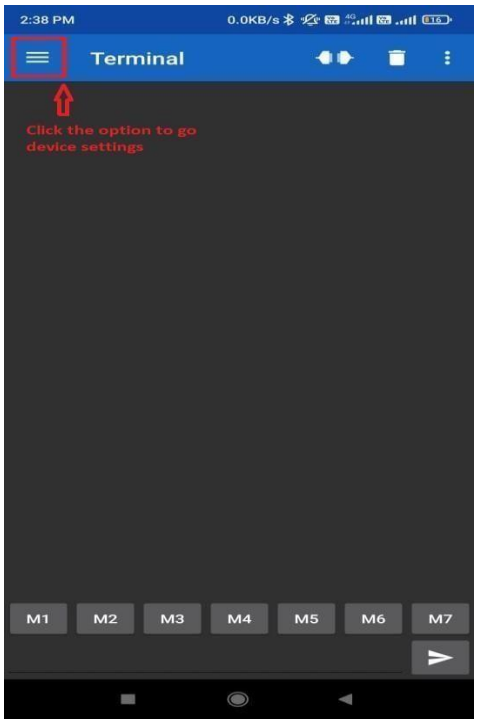
- Click the “Upload” button in the Arduino IDE to upload the code to your Arduino board.

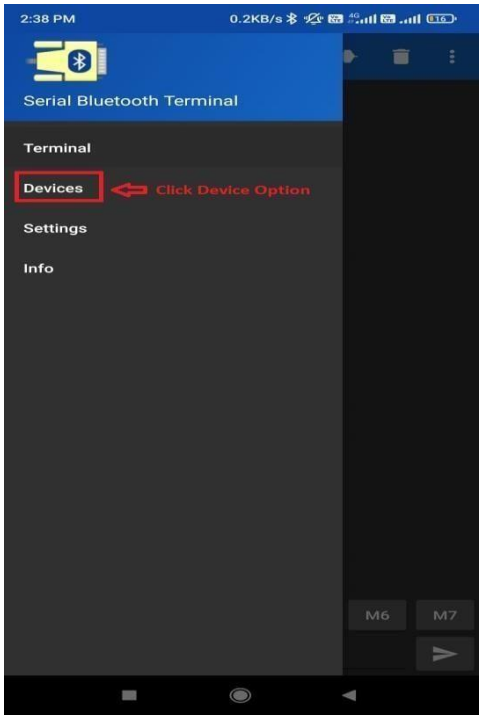
  Note: Uploading or Flashing Procedure refer Annexure -I

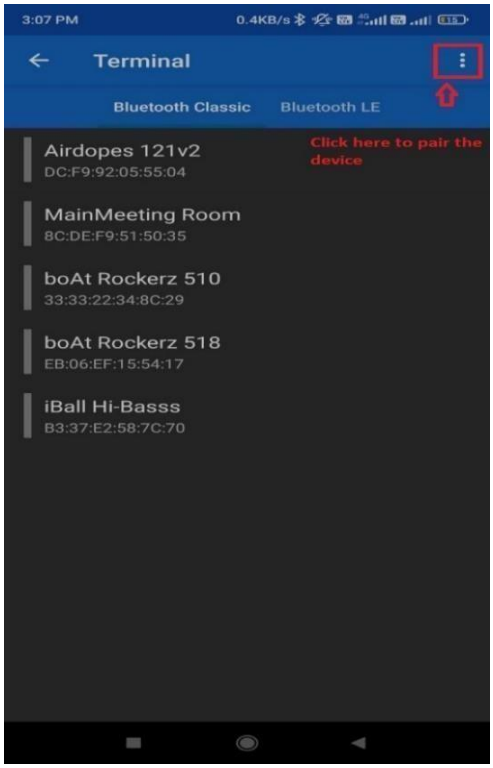
Open Serial Monitor:

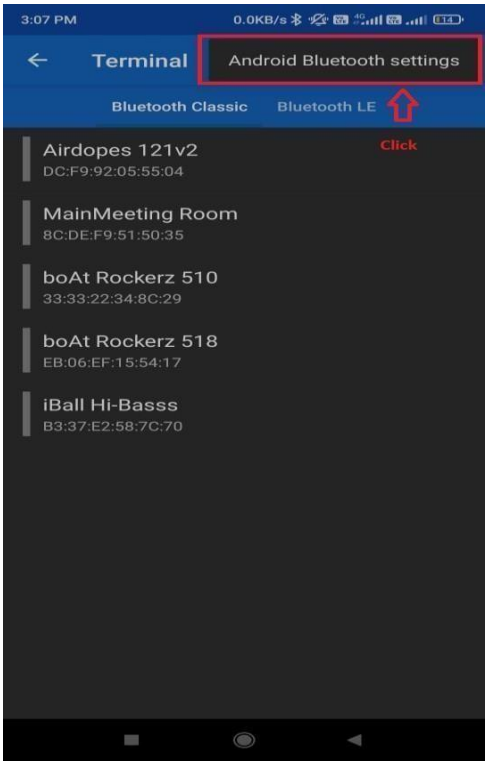


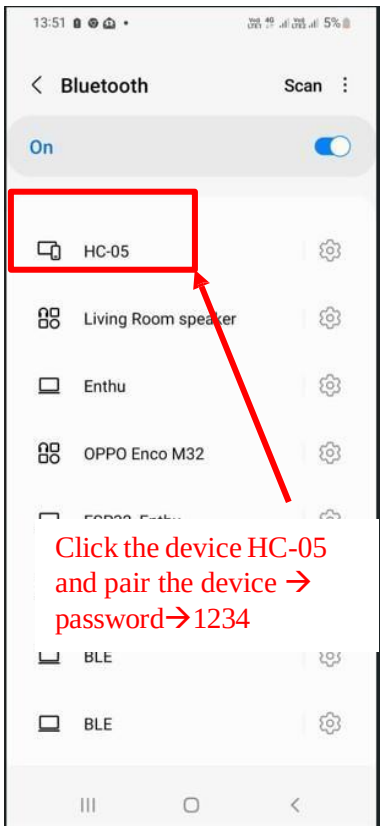
- Open the Arduino Serial Monitor from the Arduino IDE.
- Ensure that the baud rate in the Serial Monitor matches the baud rate in the Arduino code (in this case, 9600).
- Turn on Bluetooth in mobile. Open the App and follow the steps as mentioned in the below image,

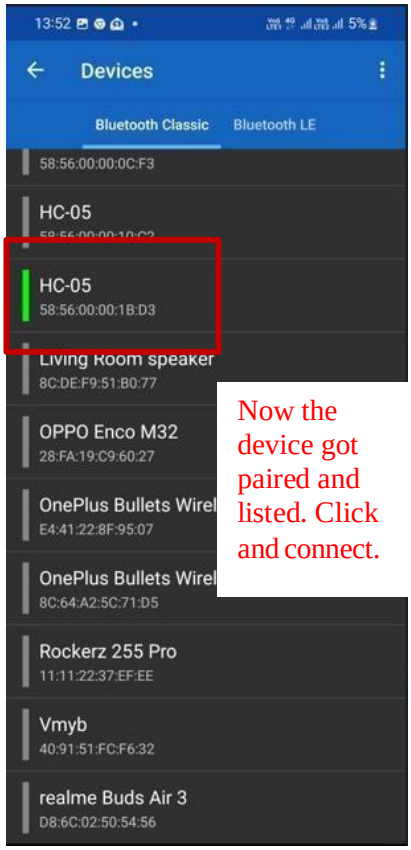


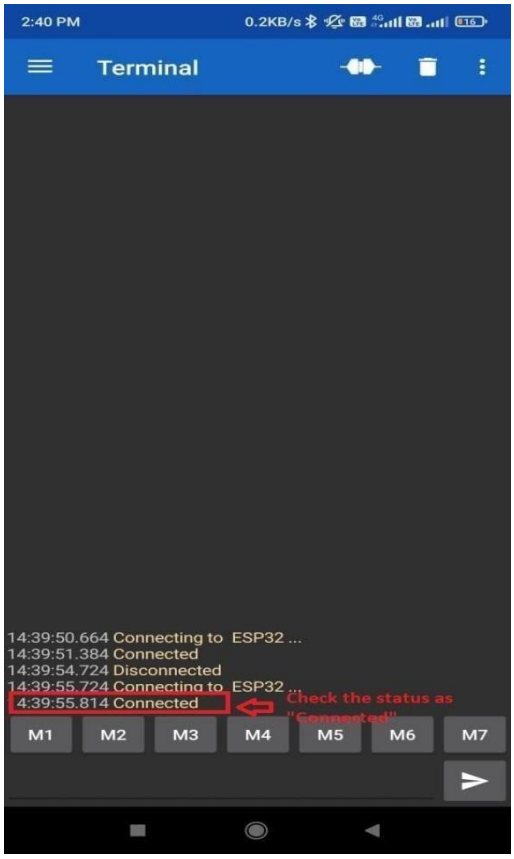



3:07 PM
0.4KB/s
Terminal
Bluetooth Classic
Bluetooth LE
Airdopes 121v2
DC:F9:92:05:55:04
Click here to pair the device
MainMeeting Room
8C:DE:F9:51:50:35
boAt Rockerz 510
33:33:22:34:8C:29
boAt Rockerz 518
EB:06:EF:15:54:17
iBall Hi-Basss
B3:37:E2:58:7C:70


3:07 PM
0.0KB/s
Terminal
Android Bluetooth settings
Bluetooth Classic
Bluetooth LE
Click
Airdopes 121v2
DC:F9:92:05:55:04
MainMeeting Room
8C:DE:F9:51:50:35
boAt Rockerz 510
33:33:22:34:8C:29
boAt Rockerz 518
EB:06:EF:15:54:17
iBall Hi-Basss
B3:37:E2:58:7C:70

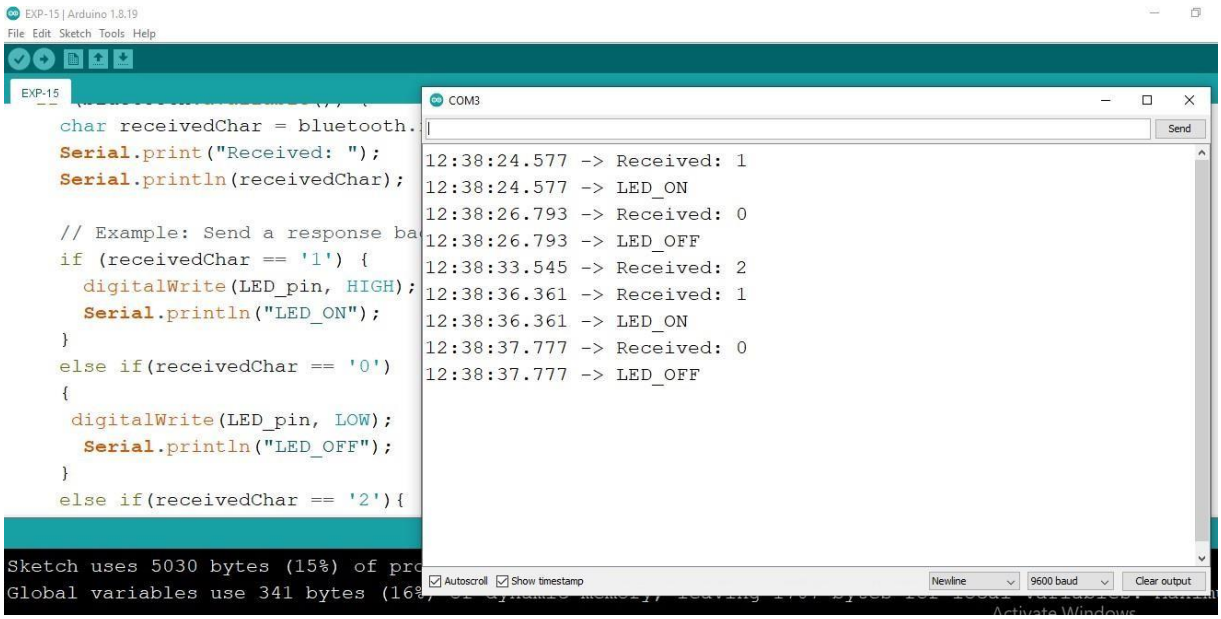

13:51
5%
Bluetooth
Scan
On
HC-05
Living Room speaker
Enthu
OPPO Enco M32
BLE
BLE
Click the device HC-05 and pair the device → password→1234


13:52
5%
Devices
Bluetooth Classic
Bluetooth LE
58:56:00:00:0C:F3
HC-05
HC-05
58:56:00:00:1B:D3
Living Room speaker
8C:DE:F9:51:B0:77
OPPO Enco M32
28:FA:19:C9:60:27
OnePlus Bullets Wirel
E4:41:22:8F:95:07
OnePlus Bullets Wirel
8C:64:A2:5C:71:D5
Rockerz 255 Pro
11:11:22:37:EF:EE
Vmyb
40:91:51:FC:F6:32
realme Buds Air 3
D8:6C:02:50:54:56
Now the device got paired and listed. Click and connect.


2:40 PM
0.2KB/s
Terminal
14:39:50.664 Connecting to ESP32 ...
14:39:51.384 Connected
14:39:54.724 Disconnected
14:39:55.724 Connecting to ESP32 ...
4:39:55.814 Connected
Check the status as "Connected"
M1
M2
M3
M4
M5
M6
M7


Terminal
12:37:19.963 Connecting to HC-05 ...
12:37:20.518 Connected
on
off
M3
M4
M5
M6
2


2:42 PM
0.0KB/s
Terminal
Click here to disconnect
14:41:46.396 Connecting to ESP32 ...
14:41:47.261 Connected
14:41:52.146 on
14:41:57.157 off
14:42:04.821 Disconnected
M1
M2
M3
M4
M5
M6
M7



- Observe the output in the serial monitor & the hardware

Test the Experiment:

- In the Serial Monitor, if the received character is ‘1’ it turn the LED on, and if received character is ‘0’ to turn the LED off.

Results:

EXP.NO: 7	INTRODUCTION TO RASPBERRY PI PLATFORM AND PYTHON PROGRAMMING
DATE:	

AIM:

Objective:

- The objective of this experiment is to interface a On Board pull-up push button with Pico control on the thingZkit IoT board. This experiment will demonstrate how to read the state of the push button and perform actions based on its status.

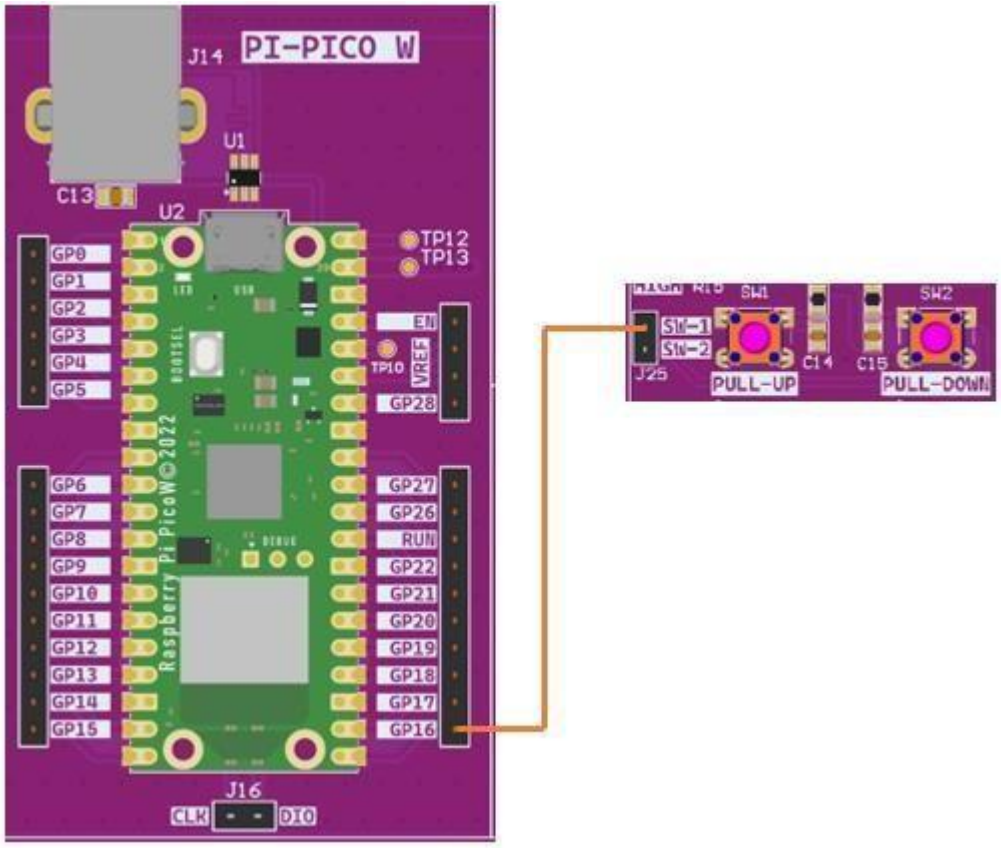
Equipment and Components:

- thingZkit IoT development board
- Jumper wires
- USB cable for PI-PICO.
- Power Adapter Cable(12V/3A) for ThingZkit IoT
- Computer with Thonny software installed.

Theory:

- In this experiment, we will interface a on Board pull-up push button with thingZkit IoT using Pico control. The pull-up configuration ensures that the input pin reads a high logic level when the button is not pressed and a low logic level when the button is pressed. Thonny software will be used for coding and uploading the program to thingZkit IoT.

Connection Diagram:



Pin Details:

PICO Controller	Push Button (Pull Up)
GP16	**SW1**

Experiment Setup:

Hardware Connection:

- Connect the On-Board pull-up push button to thingZkit IoT as per the Connection Diagram above.
- Connect the On-Board pull-up push button to thingZkit IoT (SW1 – GPIO 16 Pin)
- Ensure that the connections are secure, and the components are properly seated on the board.

thingZkit IoT Configuration:

- Connect thingZkit IoT to your computer using the USB cable.
- Open Thonny software and select the appropriate board and port.
- Open the Code Folder using Thonny Software.
- Upload the code and observe the output on the Thonny shell.
  Note: Uploading or Flashing Procedure refer Annexure -I

Results:

EXP.NO: 8	INTRODUCTION TO RASPBERRY PI PLATFORM AND PYTHON PROGRAMMING
DATE:	

AIM:

Objective:

- The objective of this experiment is to interface an On Board pull-Down push button with Pico control on the thingZkit IoT board. This experiment will demonstrate how to read the state of the push button and perform actions based on its status.

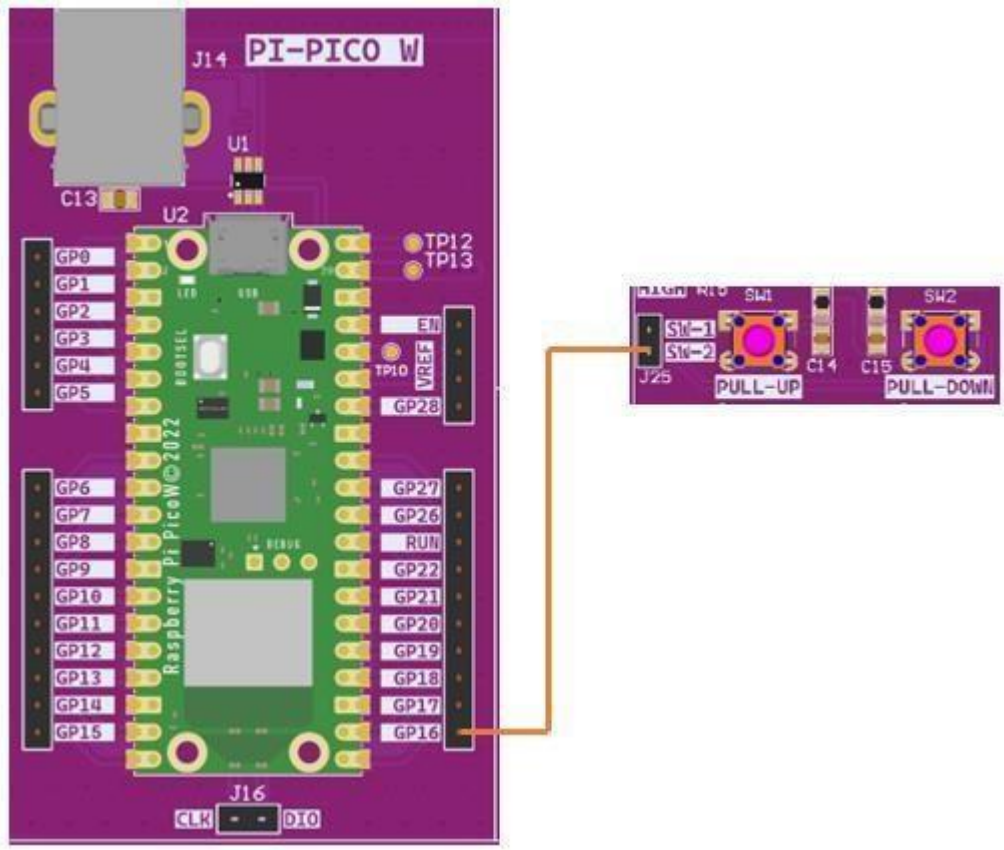
Equipment and Components:

- thingZkit IoT development board
- Jumper wires
- USB cable for PI-PICO.
- Power Adapter Cable(12V/3A) for ThingZkit IoT
- Computer with Thonny software installed.

Theory:

- In this experiment, we will interface a on Board Pull-Down push button with thingZkit IoT using Pico control. The pull-up configuration ensures that the input pin reads a low logic level when the button is not pressed and a high logic level when the button is pressed. Thonny software will be used for coding and uploading the program to thingZkit IoT.

Connection Diagram:



Pin Details:

PICO Controller	Push Button (Pull Down)
GP16	**SW2**

Experiment Setup:

Hardware Connection:

- Connect the On-Board pull-down push button to thingZkit IoT as per the Connection Diagram above.
- Connect the On-Board pull-up push button to thingZkit IoT (SW2 – GPIO 16 Pin)
- Ensure that the connections are secure, and the components are properly seated on the board.

thingZkit IoT Configuration:

- Connect thingZkit IoT to your computer using the USB cable.
- Open Thonny software and select the appropriate board and port.
- Open the Correct Code folder.
- Upload the code and observe the output on the Thonny shell.
  Note: Uploading or Flashing Procedure refer Annexure -I

Results:

EXP.NO: 9	INTERFACING SENSOR WITH RASPBERRY PI.
DATE:	

AIM:

Objective:

- The objective of this experiment is to interface the onboard IR sensor with Pico on the thingZkit IoT board. This experiment will demonstrate how to read the infrared signals from the onboard IR sensor.

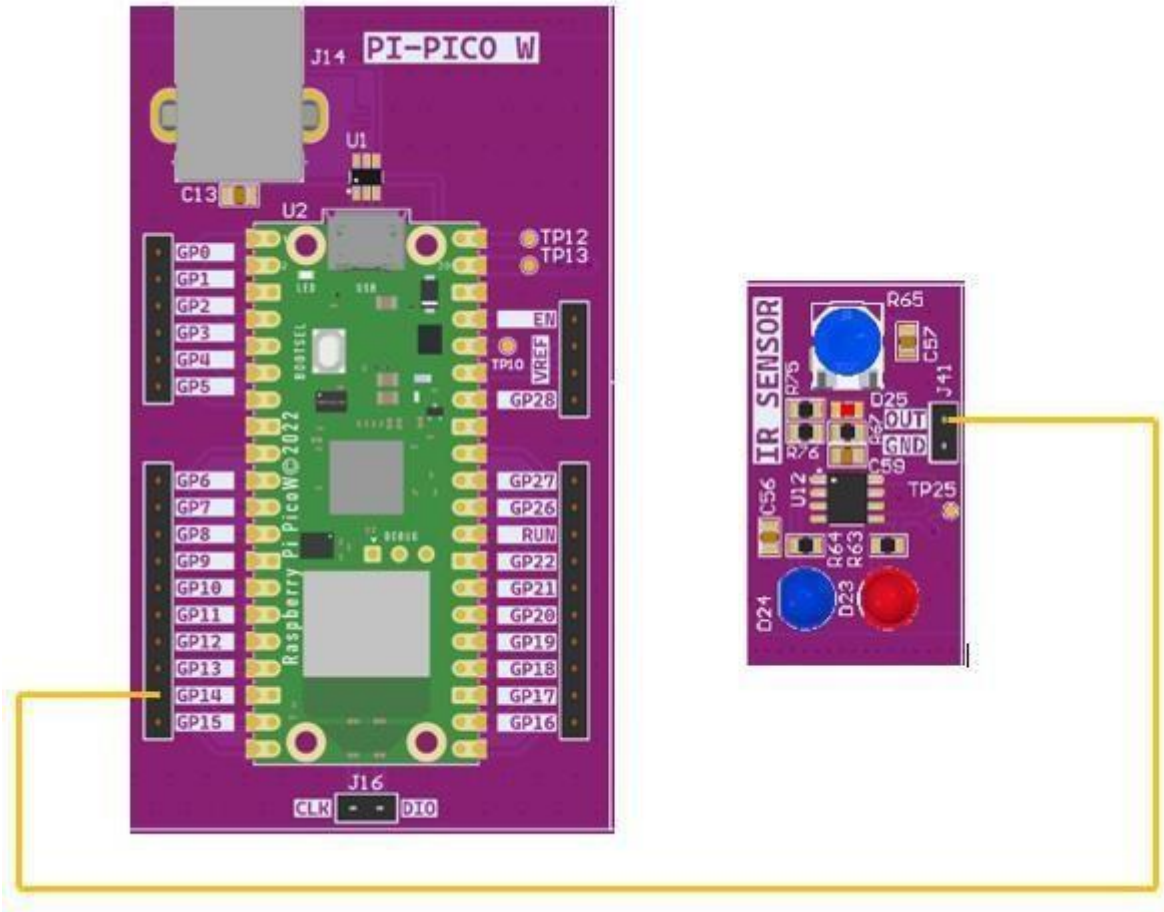
Equipment and Components:

- thingZkit IoT development board
- Jumper wires
- USB cable for PI-PICO.
- Power Adapter Cable(12V/3A) for ThingZkit IoT
- Computer with Thonny software installed.

Theory:

- In this experiment, we will interface the onboard IR sensor with thingZkit IoT using Pico control. The onboard IR sensor is a built-in component on the thingZkit IoT board. Thonny software will be used for coding and uploading the program to thingZkit IoT.

Connection Diagram:



Pin Details:

PICO Controller	IR Sensor
GP14	**OUT**

Experiment Setup:

Hardware Connection:

- Connect the On-Board IR Sensor to thingZkit IoT as per the Connection Diagram above.
- Connect the On Board IR Sensor to thingZkit IoT (OUT– GPIO 14 Pin)
- Ensure that the connections are secure, and the components are properly seated on the board.

thingZkit IoT Configuration:

- Connect thingZkit IoT to your computer using the USB cable.
- Open Thonny software and select the appropriate board and port.
- Open the Correct Code folder.
- Upload the code and observe the output on the hardware and Thonny shell.
  Note: Uploading or Flashing Procedure refer Annexure -I

Results:

EXP.NO: 10	LOD DATA USING RASPBERY PI AND UPLOAD TO THE CLOUD PLATFORM
DATE:	

AIM:

Objective:

- The objective of this experiment is to control the onboard LED and push random data to the cloud using Wi-Fi and the HTTP protocol with Pico on the ThingZkit IoT board. This experiment will demonstrate how to establish a Wi-Fi connection, control the onboard LED, and send data to the thingZmate cloud.

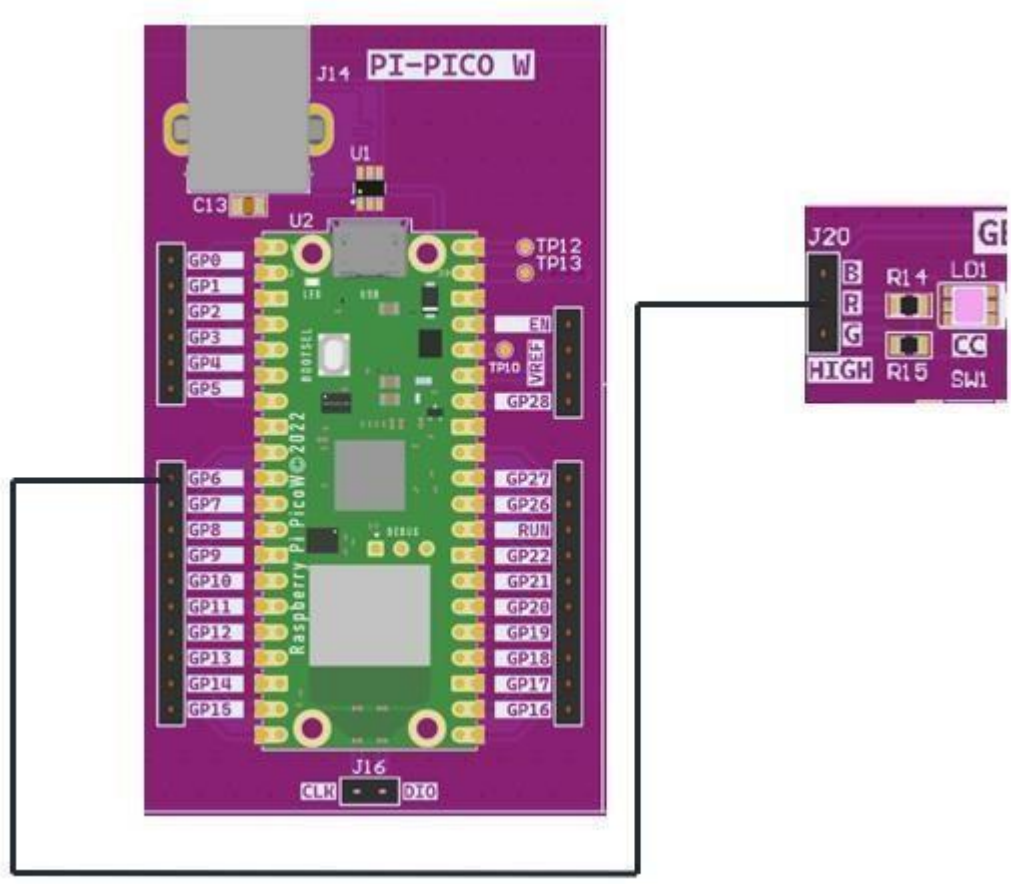
Equipment and Components:

- thingZkit IoT development board
- Jumper wires
- USB cable for PI-PICO.
- Power Adapter Cable(12V/3A) for ThingZkit IoT
- Computer with Thonny software installed.
- Cloud platform account (e.g., ThingZmate)

Theory:

- In this experiment, we will use the Wi-Fi capability of thingZkit IoT to connect to a Wi-Fi router. The onboard LED connected to Pin 6 will be controlled based on commands received over the Wi-Fi HTTP connection.
- Additionally, random data will be generated and sent to a thingZmate cloud platform using HTTP requests.

Connection Diagram:



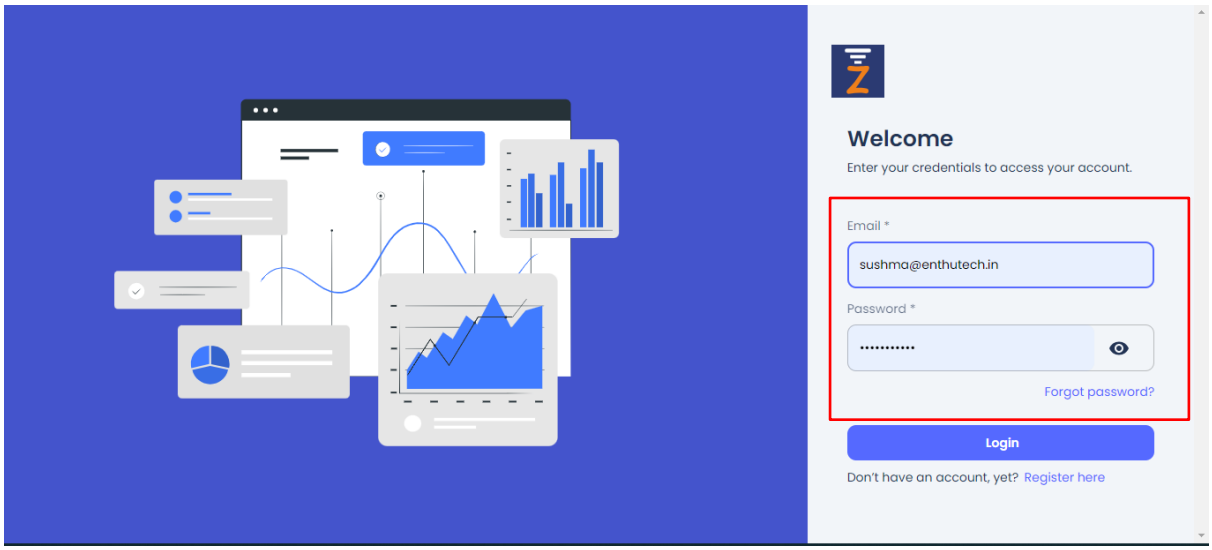
Pin Details:

PICO	RGB (Common Cathode)
GP6	**R**

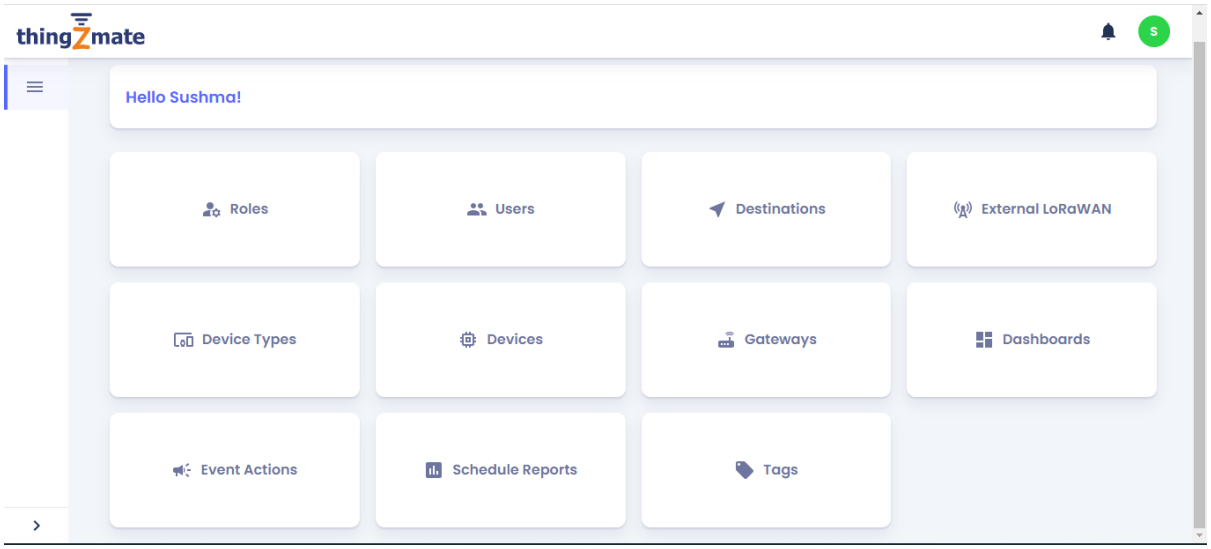
Configuration Steps:

- Open the HTTP code using Thonny Software

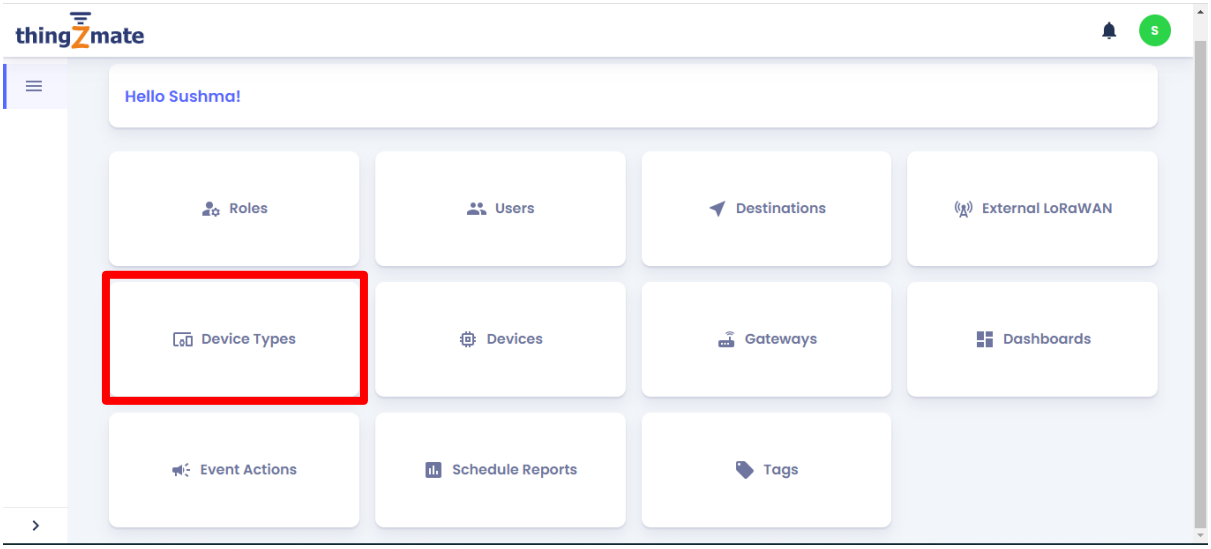
- Open thingZmate Cloud plat form and create an account



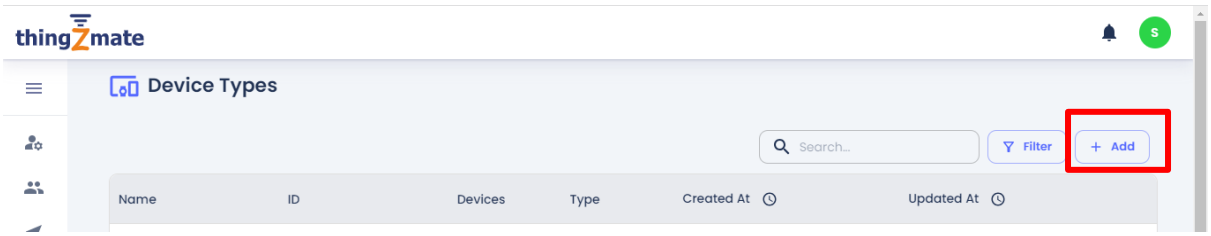
❖ After Login the thingZmate Cloud platform user can see the thingZmate Home page



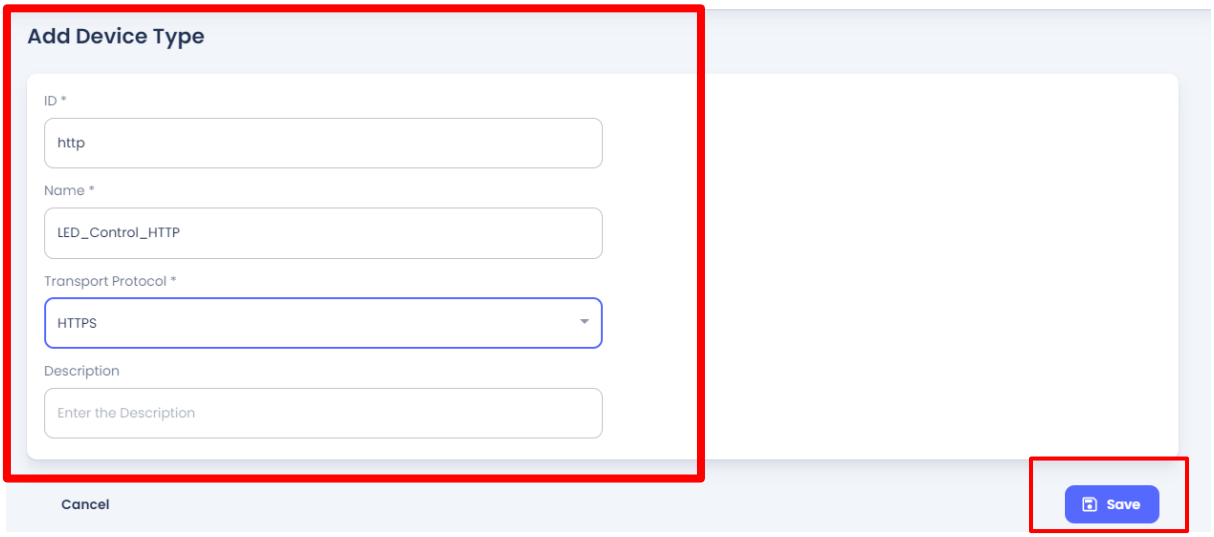
❖ Click the Device Types



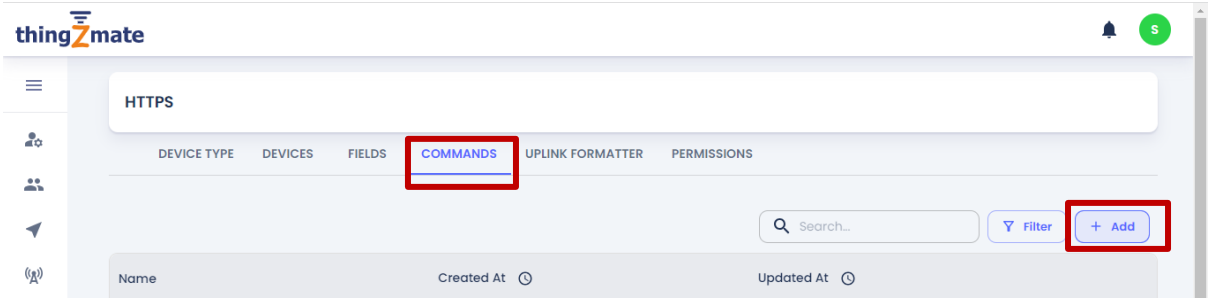
❖ Click ADD button in Device Types page.



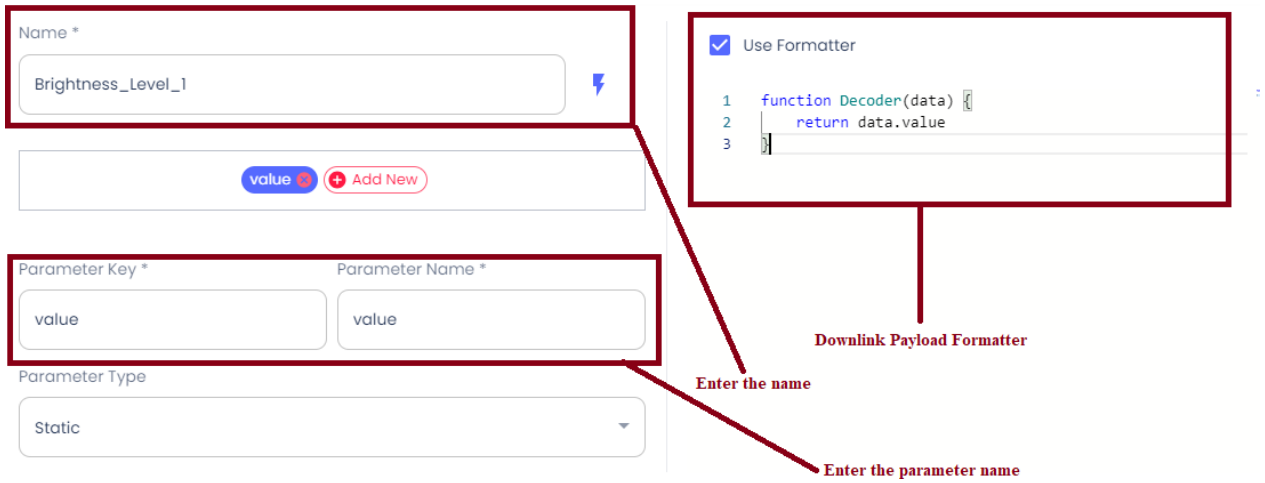
❖ User Can set ID and Name and select Transport Protocol has HTTPS and click Save Button



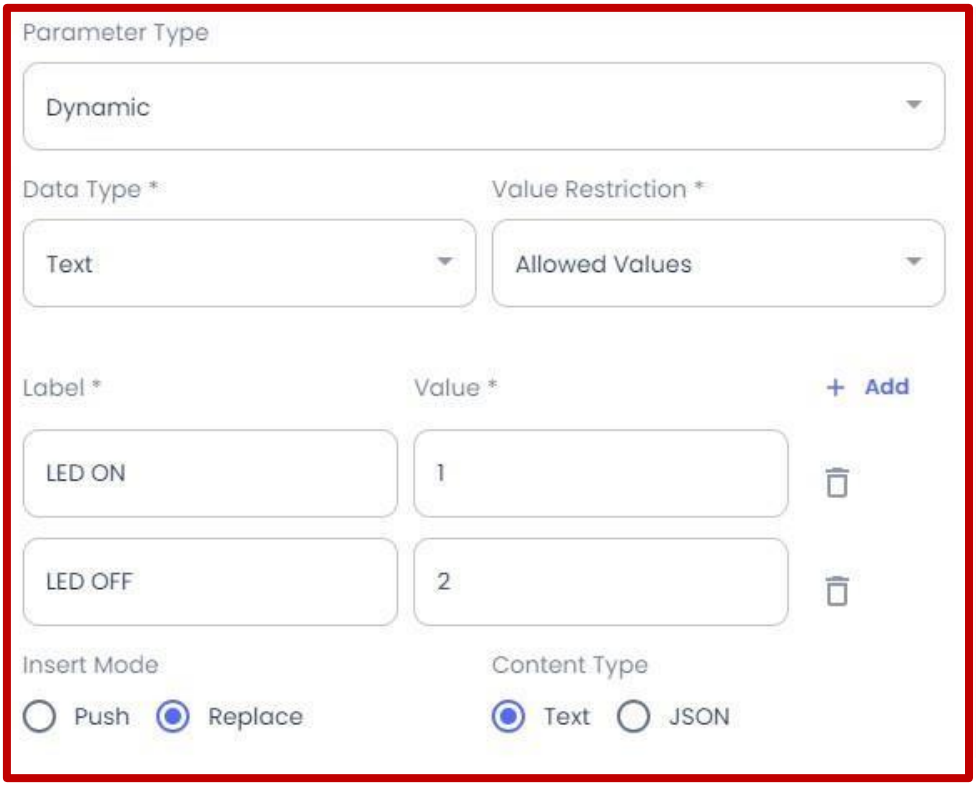
❖ After Saving the Device Types→ Select **COMMANDS** Field→Click Add Button



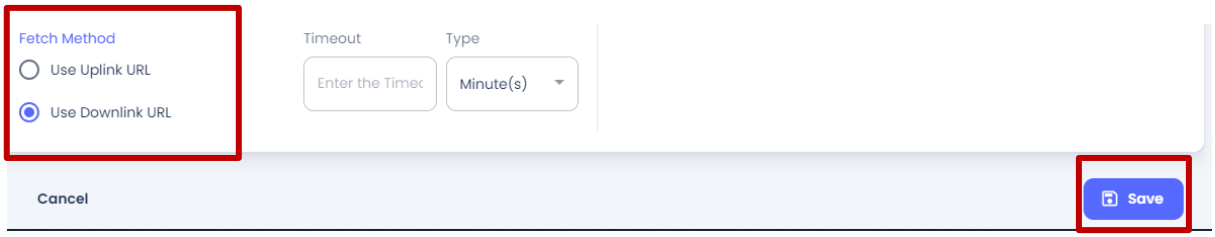
- User can Enter the Command name, parameter name , parameter Key and Type Downlink payload formatter



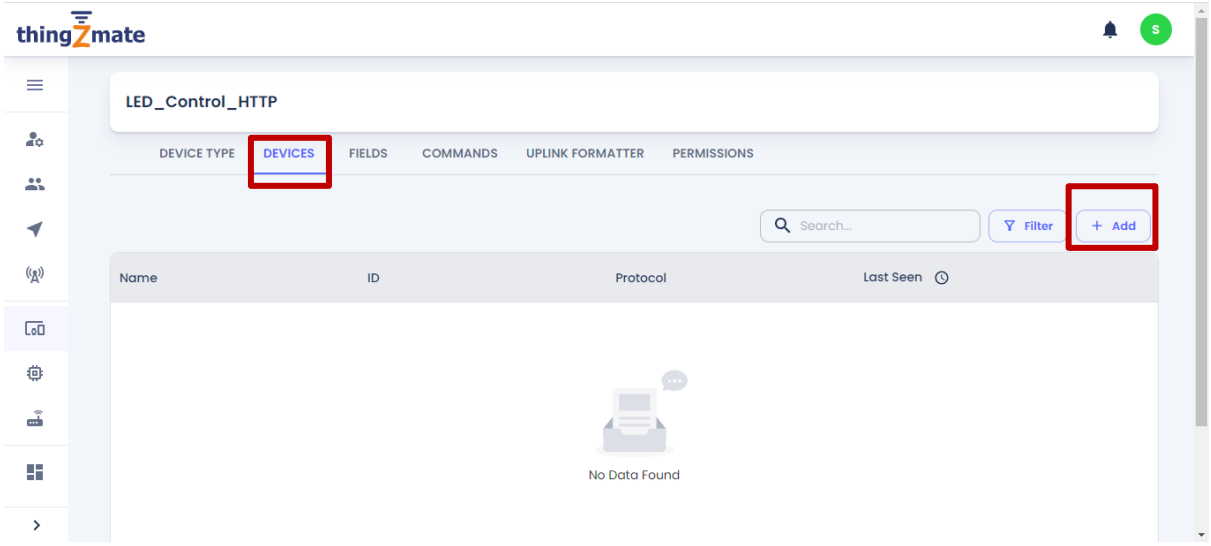
- Enter the parameter Type → Dynamic.
- Select Data Type → Text.
- Select Value Restriction → Allowed Values
- Enter the Command Label Name
- Enter the Value → 1 → LED ON → 2 → LED OFF
- Select Insert Mode→ Replace.
- Select Content Type → Text



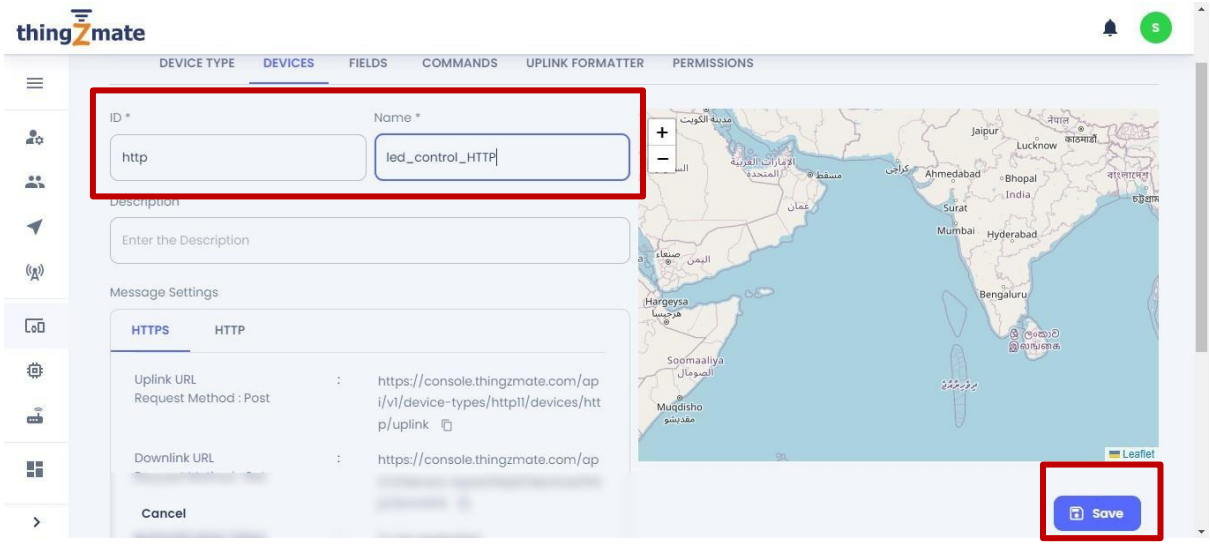
- Select the Fetch Method →Use Downlink URL
- Click Save Button



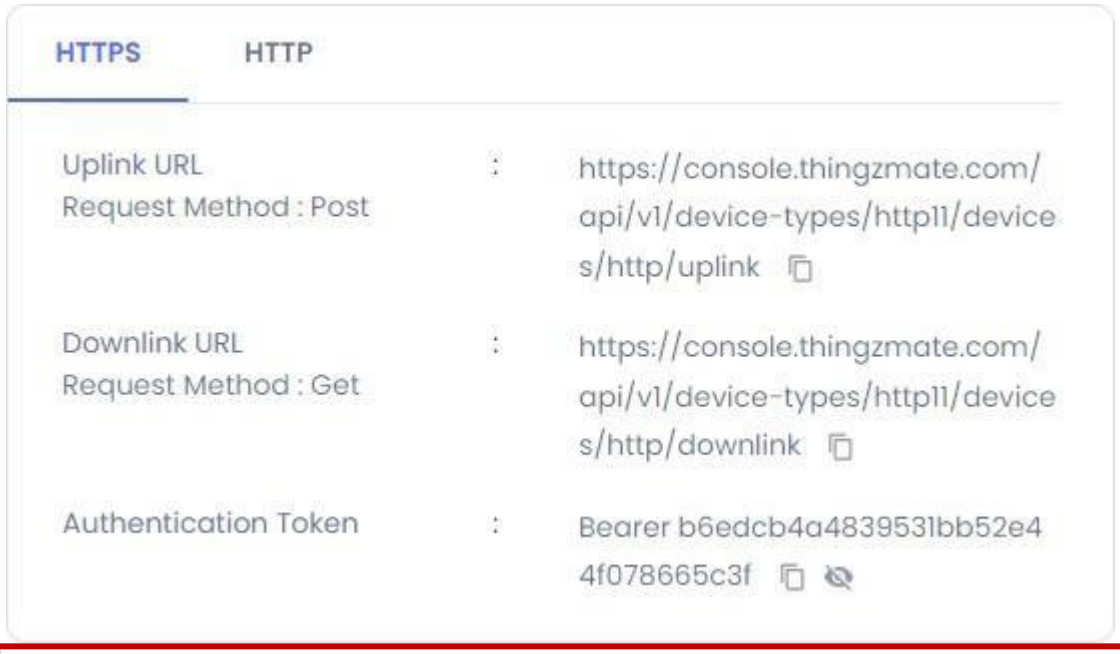
- After that→ Select **DEVICES** Field → Click ADD Button



- User can set Device ID and Name → Click Save Button



- ❖ After Saving the device → Select HTTPS → Copy the URL and Paste the Source Code
- ❖ Copy the Uplink URL (Cloud plat form) →paste the Uplink_Server_url(Code)
- ❖ Copy the Downlink URL (Cloud plat form) →paste the downlink_Server_url(Code)
- ❖ Copy the Authentication Token without Bearer word (Cloud plat form) →paste the auth_token (Code)



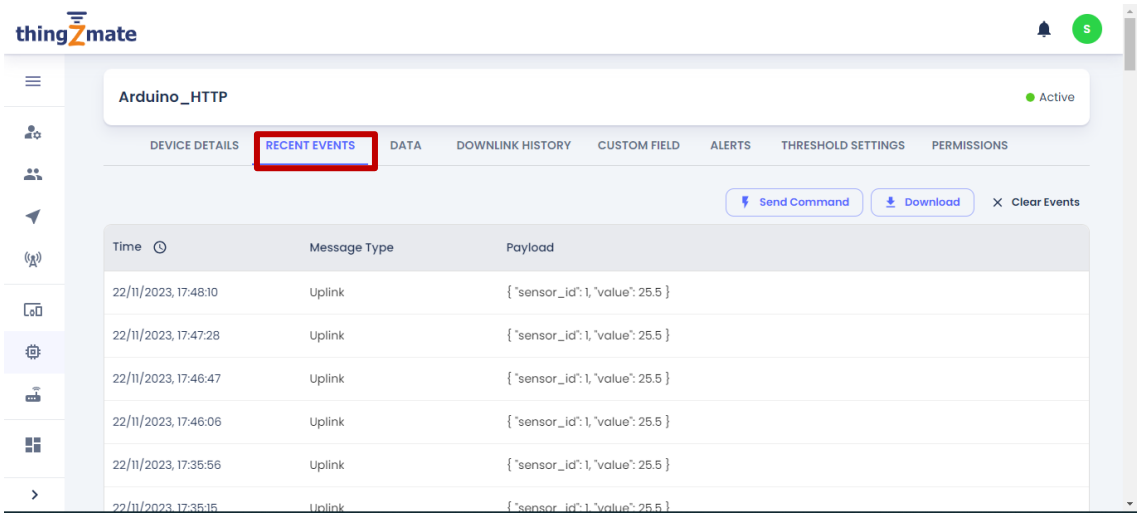

```
# Define the server URLs & auth_token
uplink_server_url = "https://console.thingzmate.com/api/v1/device-types/http1/devices/http/uplink"
downlink_server_url = "https://console.thingzmate.com/api/v1/device-types/http1/devices/http/downlink"
auth_token = "e07f5cffacfe7d782adbaa1f67de7e6a"
```

- ❖ After that→ Give your Wi-Fi Credentials in Code

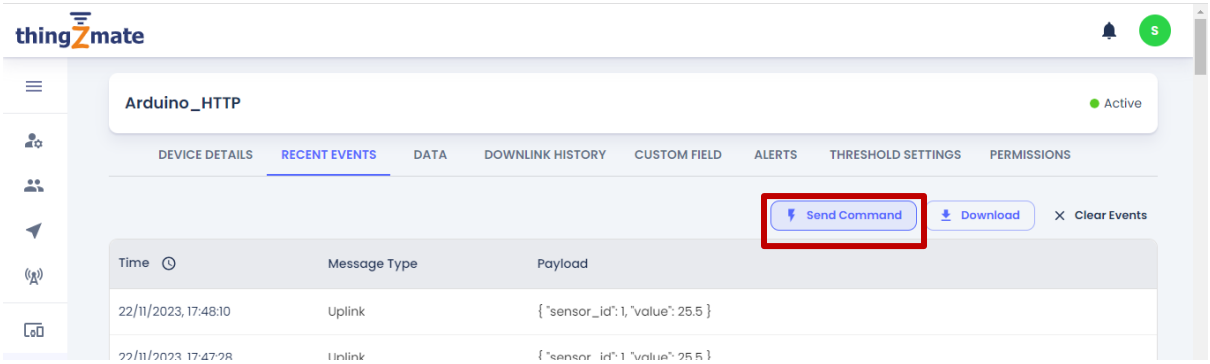
```
# Your Wi-Fi credentials
ssid = 'XXXX'
password = 'XXXXX'
```

❖ Upload the code in Thonny Software
Note: Uploading or Flashing Procedure refer Annexure -I

❖ After Uploading code → Select Recent Events Field →User can see the Result



❖ Click the Send Command



- Select your LED ON and OFF Command Name → Click send Button.

- Observe the Output on the Hardware and thingZmate cloud and ThonnyShell

Results: