ANNA UNIVERSITY REGIONAL CAMPUS

COIMBATORE - 641 046.

CS3691 EMBEDDED SYSTEMS AND IOT LABORATORY

Name: ………………………………………………………………….

Register No.: …………………………………………………………..

Degree & Branch: …………………………………………………..

Semester: ………………………………………………………………

Subject Code & Title: ………………………………………………..

ANNA UNIVERSITY REGIONAL CAMPUS

COIMBATORE – 641 046.

Register No.: ________________________

BONAFIDE CERTIFICATE

Certified to be the BONAFIDE RECORD work done by

Mr./ Ms. ____ of VI Semester, B.Tech. Artificial Intelligence and Data science Discipline in the <u>CS3691 Embedded Systems and IoT Laboratory</u> Practical Course during the Academic year 2024- 2025.

Date ……………………….

Staff in charge **Head of the Department**

Submitted for the University Practical Examination held on ……………

Internal Examiner **External Examiner**

TABLE OF CONTENTS

S.No.	Date	Title of the Experiment	Page No	Signature

S.No.	Date	Title of the Experiment	Page No	Signature

EXP NO. : 01	8051 ASSEMBLY LANGUAGE EXPERIMENT USING SIMULATOR
DATE:	

AIM:

To write an 8051 Assembly language experiment using simulator.

APPARATUS REQUIRED:

S.NO.	NAME OF COMPONENT	SPECIFICATION	QUANTITY
1.	Keil µ vision 5 software	Version 5	1
2.	PC with Windows	Windows 7	1

PROCEDURE:

Step 1 : Create a new project in Keil µ Vision & Select the device AT89C51.

Step 2 : Add the code to a new C file in the project & Save it as .asm file

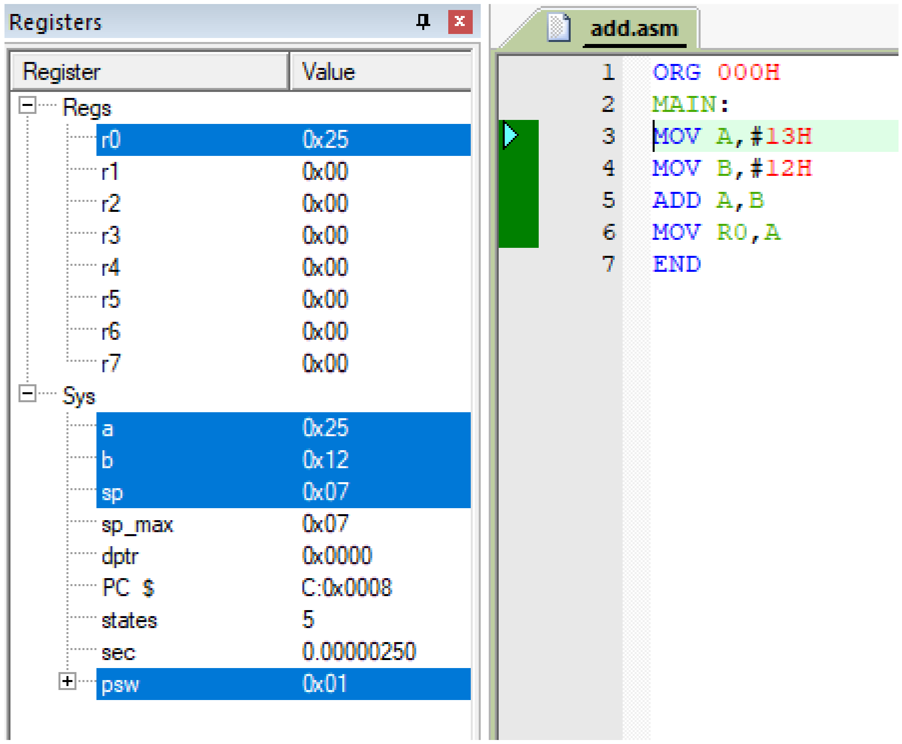
Step 3 : Debug the Code using Debug → Start / Stop Debug Session.

Step 4 : Stop the Program.

PROGRAM:

```
ORG

0000H

CLR C

MOV A, #13H

ADD A, #12H

MOV R0, A

END
```

OUTPUT:



RESULT:

Thus the 8051-assembly language program is written and executed successfully.

EXP NO. : 02	TEST DATA TRANSFER BETWEEN REGISTERS AND MEMORY
DATE:	

AIM:

To execute an Assembly language program to transfer data between registers and memory.

APPARATUS REQUIRED

S.NO.	NAME OF COMPONENT	SPECIFICATION	QUANTITY
1.	Keil µ vision 5 software	Version 5	1
2.	PC with Windows	Windows 7	1

PROCEDURE:

Step 1 : Create a new project in Keil µ Vision & Select the device AT89C51.

Step 2 : Add the code to a new C file in the project & Save it as .asm file.

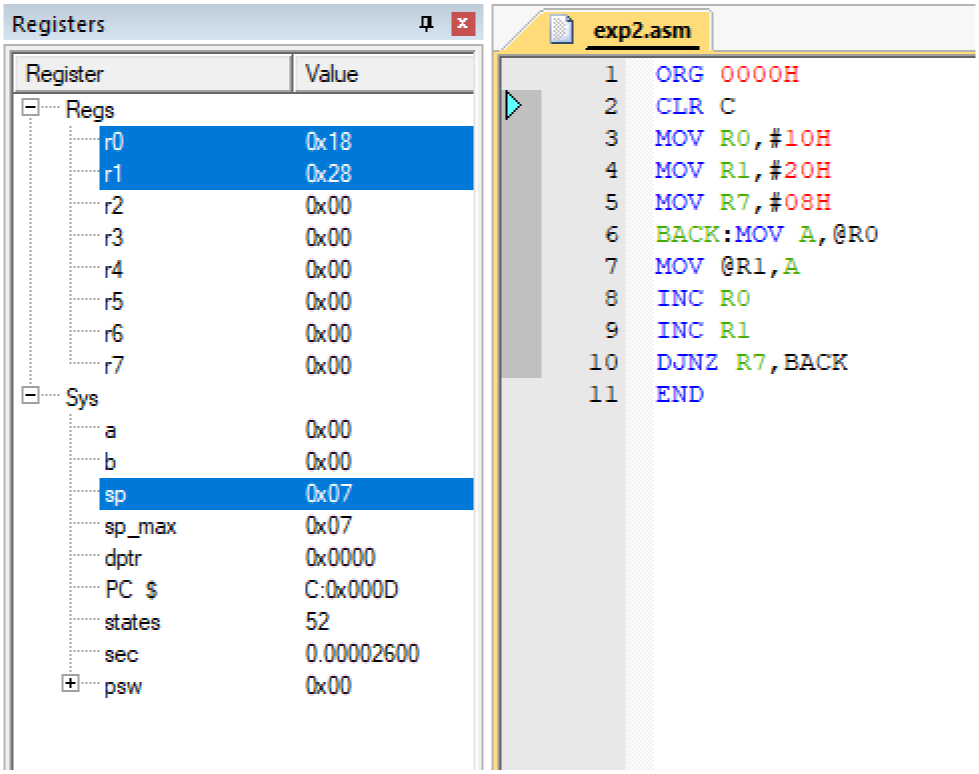
Step 3 : Debug the Code using Debug → Start / Stop Debug Session.

Step 4 : Stop the Program.

PROGRAM:

```
ORG 0000H
CLR C
MOV R0, #10H
MOV R1, #20H
MOV R7, #08H
BACK: MOV A, @R0
MOV @R1, A
INC R0
INC R1
DJNZ R7, BACK
END
```

OUTPUT:



RESULT:

Thus, Assembly language program to transfer data between registers and memory is written and executed successfully.

EXP NO. : 03	ALU PROGRAM
DATE:	

AIM:

To write and execute the ALU program using the Keil simulator.

APPARATUS REQUIRED:

S.NO.	NAME OF COMPONENT	SPECIFICATION	QUANTITY
1.	Keil µ vision 5 software	Version 5	1
2.	PC with Windows	Windows 7	1

PROCEDURE:

Step 1 : Create a new project in Keil µ Vision & Select the device AT89C51.

Step 2 : Add the code to a new C file in the project & Save it as .asm file

Step 3 : Debug the Code using Debug → Start / Stop Debug Session.

Step 4 : Stop the Program.

THEORY

An arithmetic-logic unit is the part of a central processing unit that carries out arithmetic and logic operations on the operands in computer instruction words. In some processors, the ALU is divided into two units: an arithmetic unit (AU) and a logic unit (LU). Some processors contain more than one AU -- for example, one for fixed-point operations and another for floating-point operations.

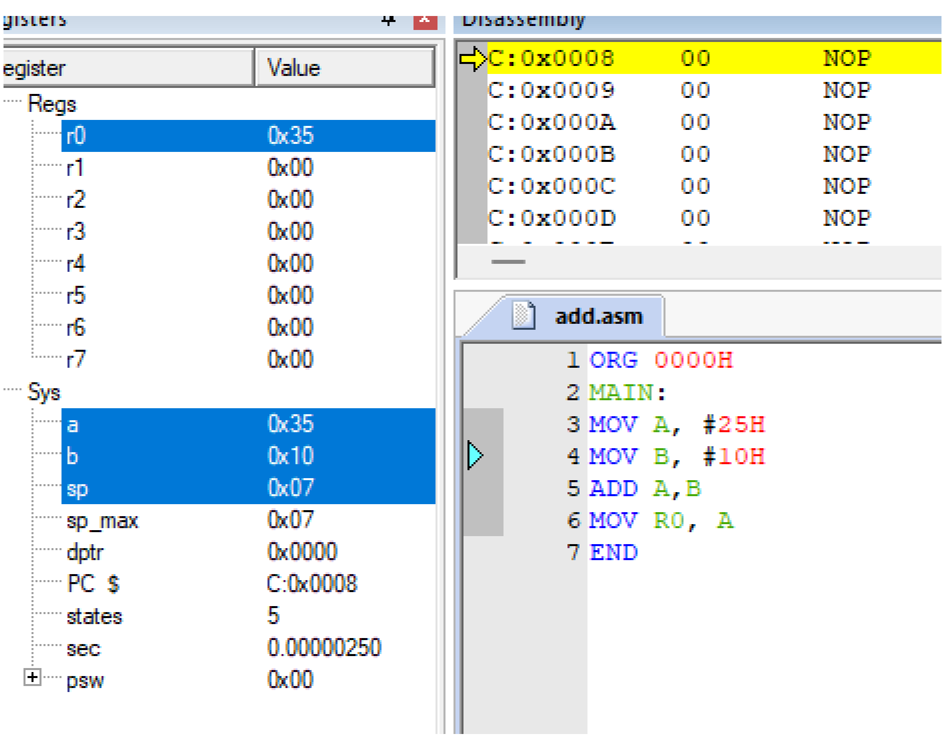
Typically, the ALU has direct input and output access to the processor controller, main memory (random access memory or RAM in a personal computer) and input/output devices. Inputs and outputs flow along an electronic path that is called a bus. The input consists of an instruction word, sometimes called a machine instruction word, that contains an operation code or "opcode," one or more operands and sometimes a format code. The operation code tells the ALU what operation to perform and the operands are used in the operation.

PROGRAM:

ADDITION :

```
ORG 000H
 MAIN:
 MOV A, #25H
 MOV  B,  #10H
 ADD A, B
 MOV    R0,    A
 END
```

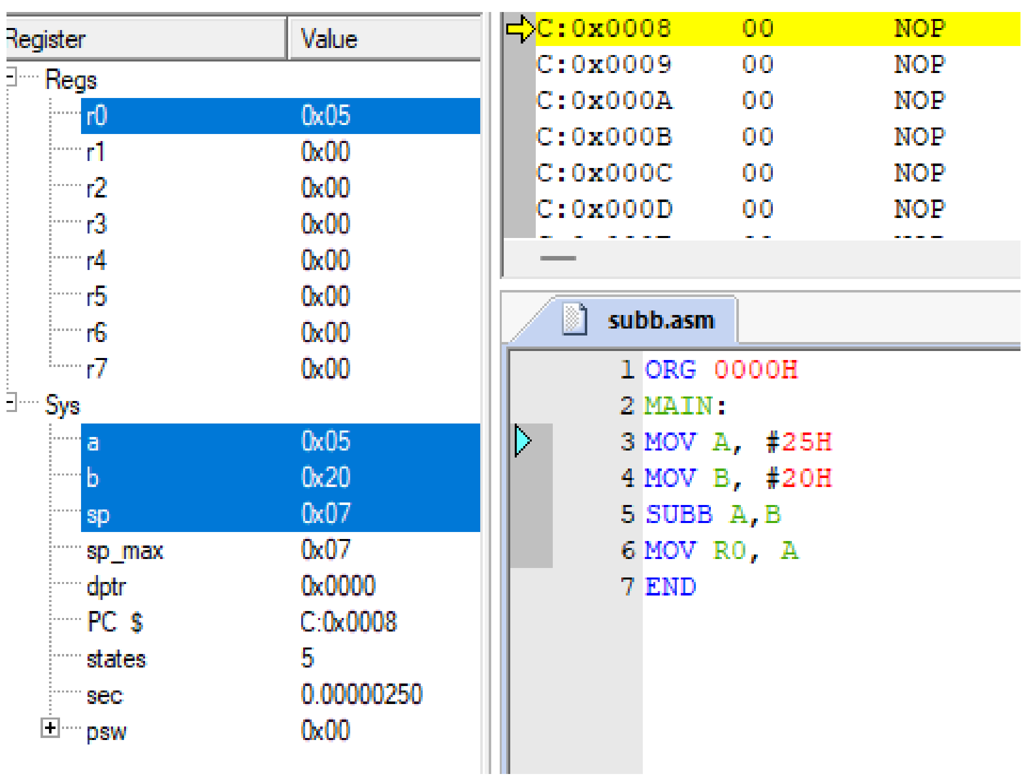
OUTPUT:



SUBTRACTION :

```
ORG 000H
MAIN:
MOV A, #25H
MOV B, #20H
SUBB A, B
MOV R0, A
END
```

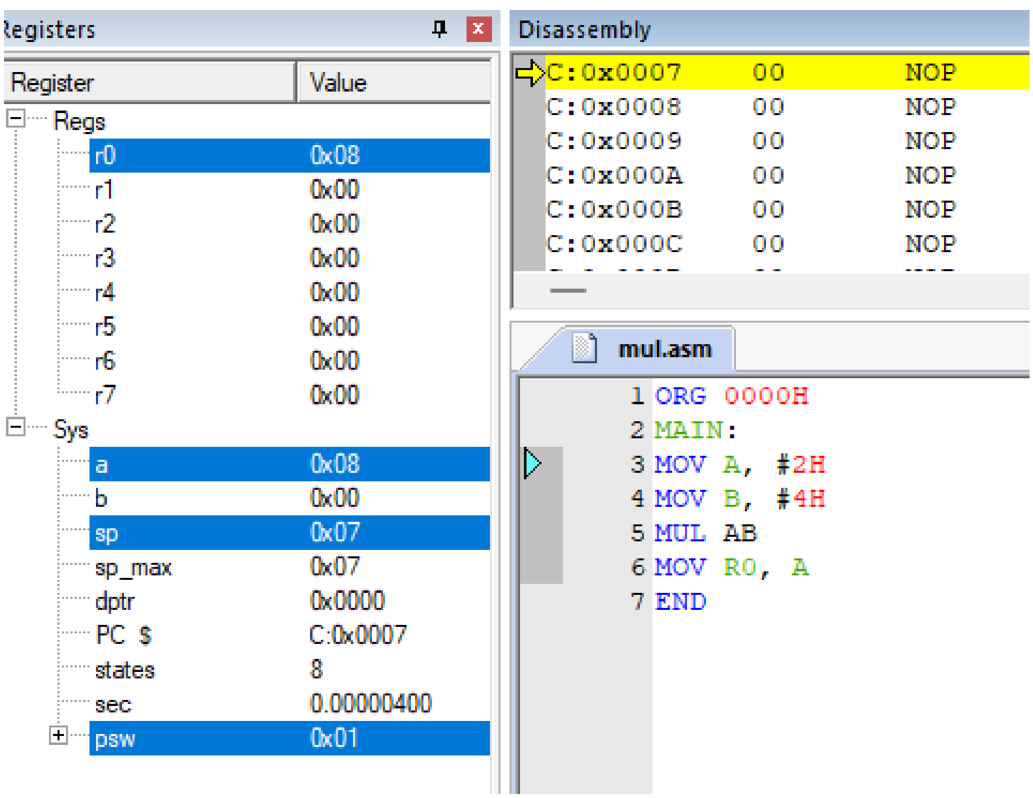
OUTPUT:



MULTIPLICATION:

```
ORG 000H
MAIN:
MOV A, #2H
MOV B, #4H
MUL AB
MOV R0, A
END
```

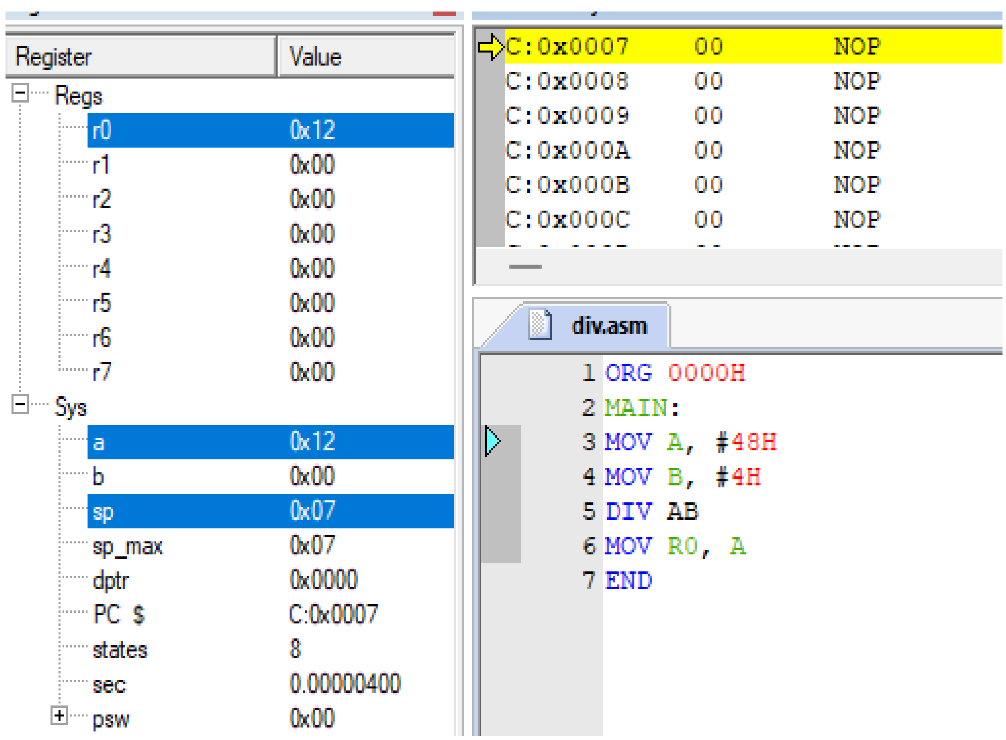
OUTPUT:



DIVISION:

```
ORG 000H

MAIN:

MOV A, #48H

MOV B, #4H

DIV AB

MOV R0, A

END
```

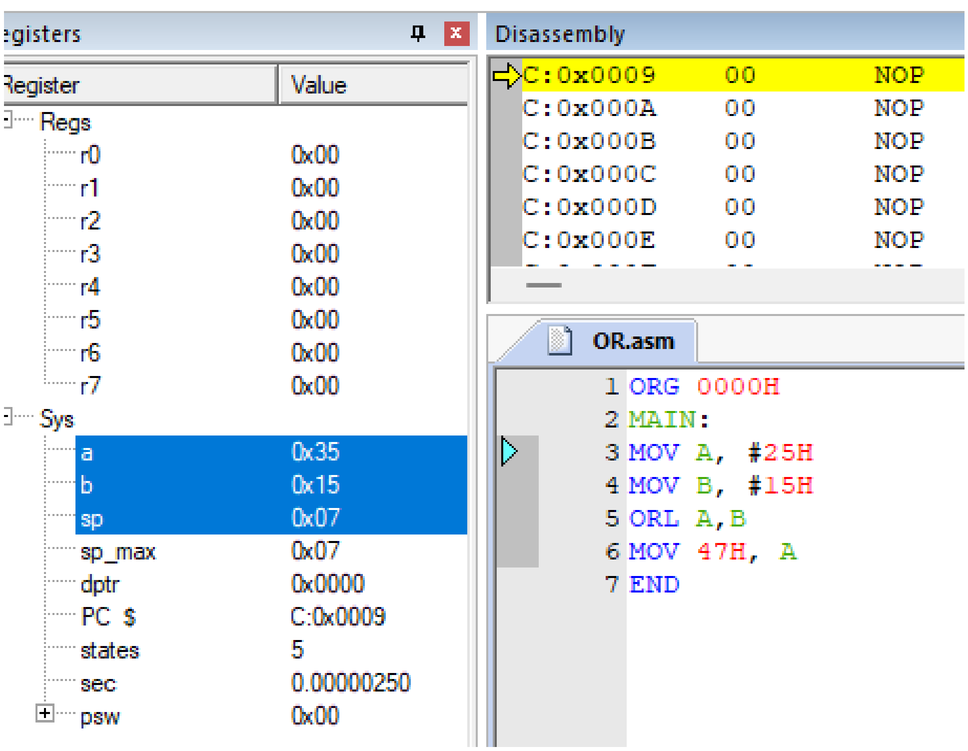
OUTPUT:



OR :

```
ORG 000H
MAIN:
MOV A, #25H
MOV B, #15H
ORL  A, B
MOV 47H,  A
END
```

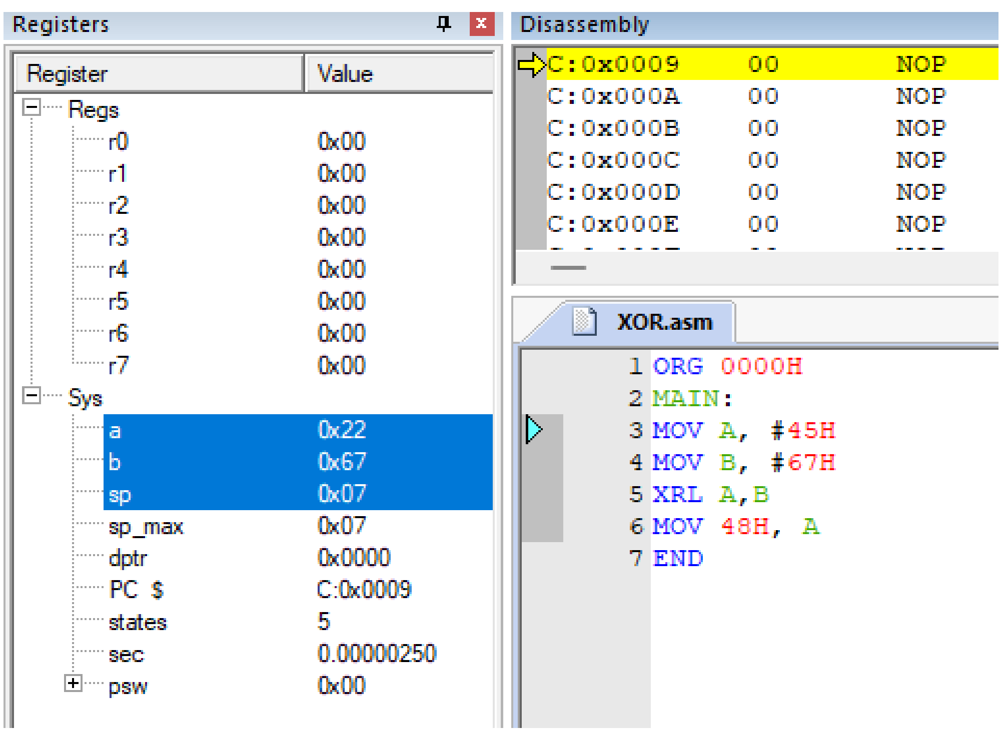
<u>OUTPUT</u>



XOR :

```
ORG 000H
MAIN:
MOV A, #45H
MOV B, #67H
XRL  A, B
MOV 48H,  A
END
```

OUTPUT:



RESULT:

Thus, the ALU program using the Keil simulator is written and executed successfully.

EXP NO. : 04	**BASIC ARITHMETIC PROGRAM USING EMBEDDED C**
DATE:	

AIM:

To write a basic arithmetic Program using Embedded C

APPARATUS REQUIRED:

S.NO.	NAME OF COMPONENT	SPECIFICATION	QUANTITY
1.	Keil µ vision 5 software	Version 5	1
2.	PC with Windows	Windows 7	1

PROCEDURE:

Step 1 : Create a new project in Keil µ Vision & Select the device AT89C51.

Step 2 : Add the code to a new C file in the project & Save it as .asm file

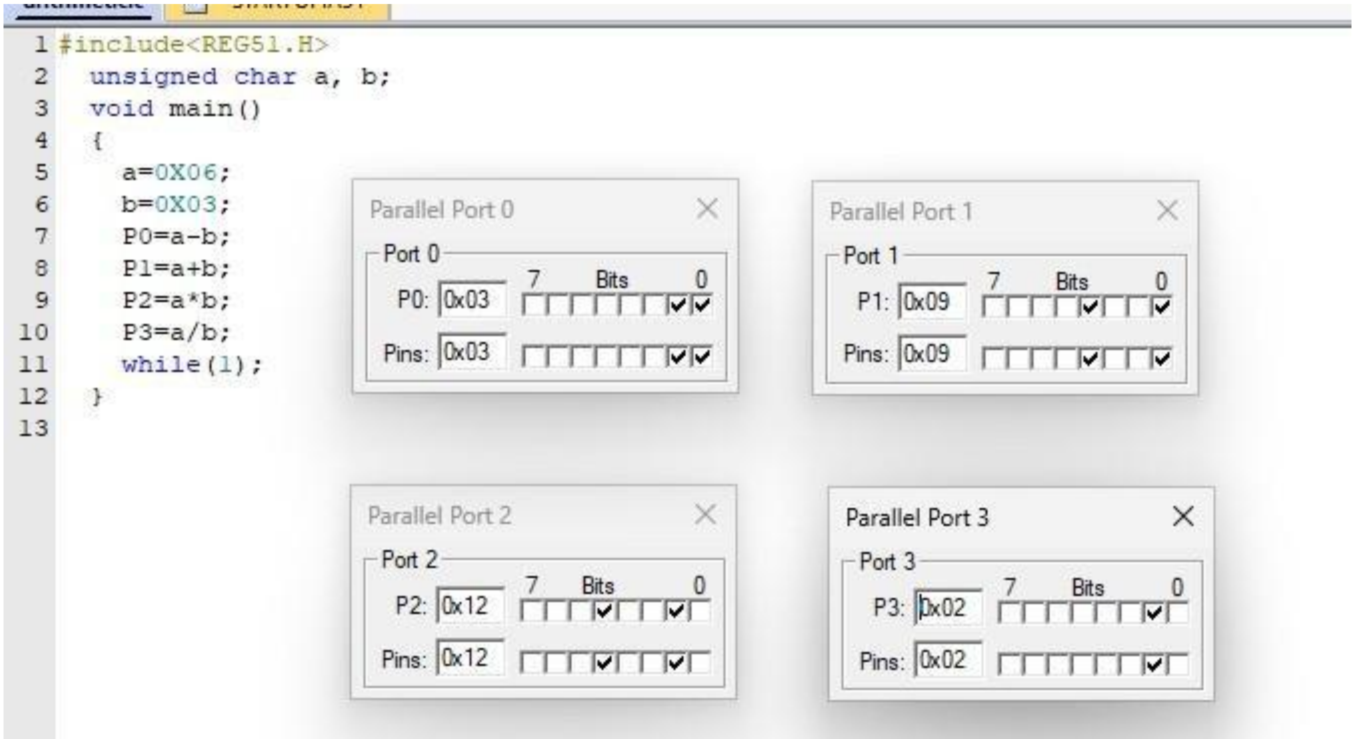
Step 3 : Debug the Code using Debug → Start / Stop Debug Session.

Step 4 : Stop the Program.

PROGRAM:

```
#include<REG51.H>
unsigned char a, b;
void main()
{
a=0x06;
b=0x03;
P0=a-b;
P1=a+b;
P2=a*b;
P3=a/b;
while(1);
```

OUTPUT:



RESULT:

Thus, a basic arithmetic Program using Embedded C was written and executed successfully.

EXP NO.: 05	INTRODUCTION TO ARDUINO PLATFORM AND PROGRAMMING
DATE:	

AIM:

To study the basics of Arduino Uno board and Arduino IDE 2.0 software.

HARDWARE & SOFTWARE TOOLS REQUIRED:

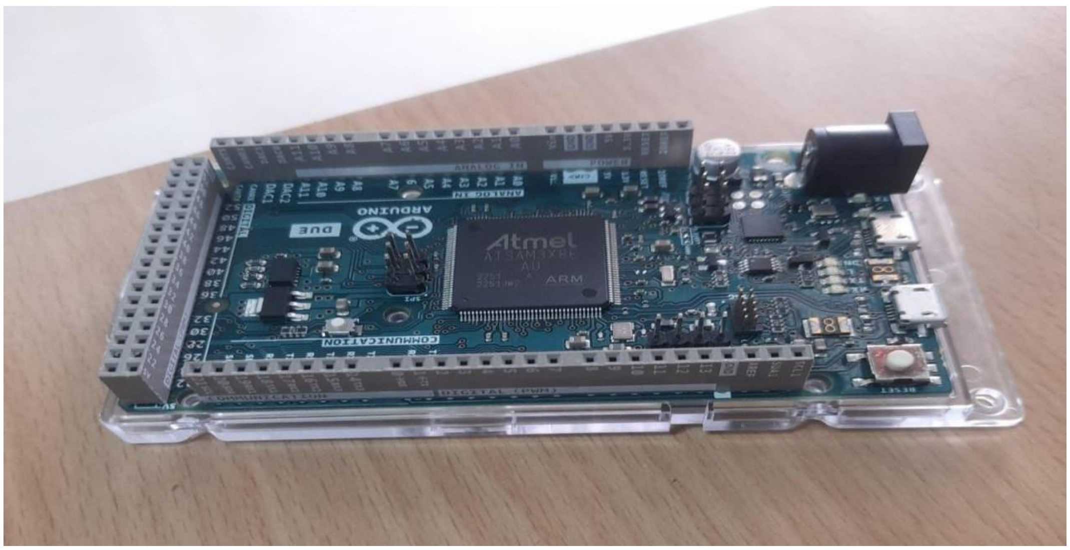
S.NO	Hardware & Software Requirement	Quantity
1	Arduino IDE 2.0	1
2	Arduino UNO Board	1
3	Jump wires	Few
4	Arduino USB Cable	1

INTRODUCTION TO ARDUINO:

Arduino is a project, open-source hardware, and software platform used to design and build electronic devices. It designs and manufactures microcontroller kits and single-board interfaces for building electronics projects. The Arduino boards were initially created to help students with the non-technical background. The designs of Arduino boards use a variety of controllers and microprocessors. Arduino is an easy-to-use open platform for creating electronic projects. Arduino boards play a vital role in creating different projects. It makes electronics accessible to non-engineers, hobbyists, etc. The various components present on the Arduino boards are a Microcontroller, Digital Input/output pins, USB Interface and Connector, Analogue Pins, reset buttons, Power buttons, LEDs, Crystal oscillators, and Voltage regulators. The most standard and popular board used over time is Arduino UNO. The ATmega328 Microcontroller present on the UNO board makes it rather powerful than other boards. There are various types of Arduino boards usedfor different purposes and projects. The Arduino Boards are organized using the Arduino (IDE), which can run on various platforms. Here, IDE stands for Integrated Development Environment.

ARDUINO DUE:

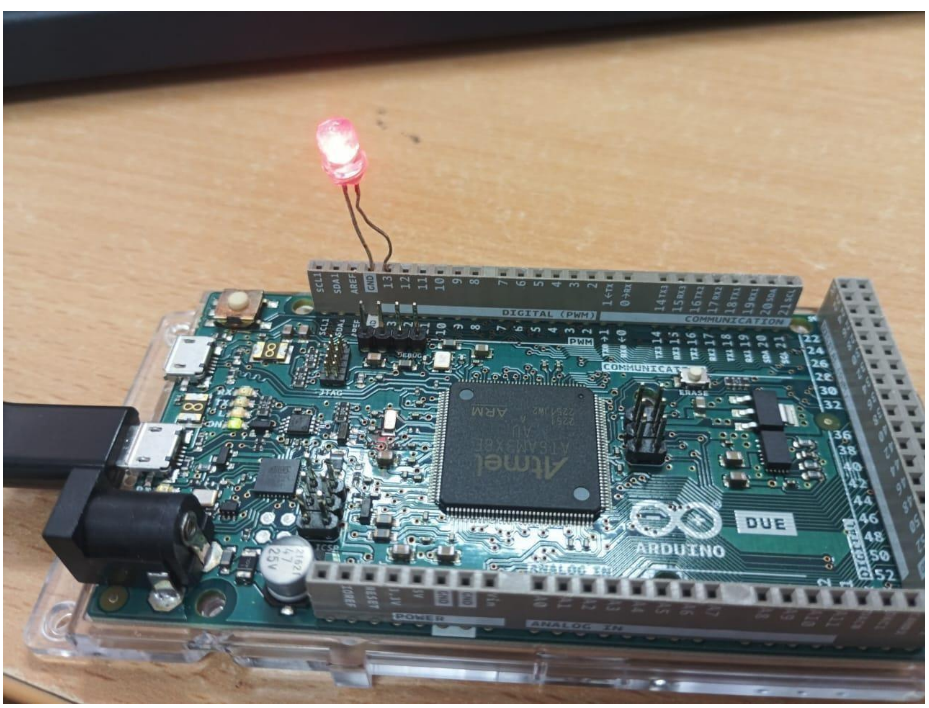
The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 processor, making it the first Arduino to use a 32-bit ARM core. Operating at 3.3V with an 84 MHz clock speed, it offers higher processing power compared to 8-bit boards. It features 54 digital I/O pins, 12 analog inputs, 2 analog outputs (DAC), and USB host capabilities. With 512 KB of flash memory and 96 KB of SRAM, it is ideal for advanced projects involving data processing, robotics, and signal generation. However, care must be taken as its pins are not 5V tolerant, unlike most Arduino boards.



PROGRAM:

```
void setup()

{

pinMode(4,OUTPUT);

}

void loop() {

  digitalWrite(4,HIGH);

  delay(1000);

  digitalWrite(4, LOW);

  delay(1000);

}
```

OUTPUT:



RESULT:

Thus the study of Arduino DEU board and Arduino IDE platform is done successfully.

EXP NO.: 06	EXPLORE DIFFERENT COMMUNICATION METHODS WITH IOT DEVICES (ZIGBEE, GSM, BLUETOOTH)
DATE:	

AIM:

To Explore different communication methods with IoT devices (Zigbee, GSM, Bluetooth).

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Hardware & Software Requirement	Quantity
1	Bluetooth Module	1
2	Arduino UNO Board	1
3	Jump wires	Few
4	Arduino USB Cable	1
5	Serial Bluetooth Terminal	1

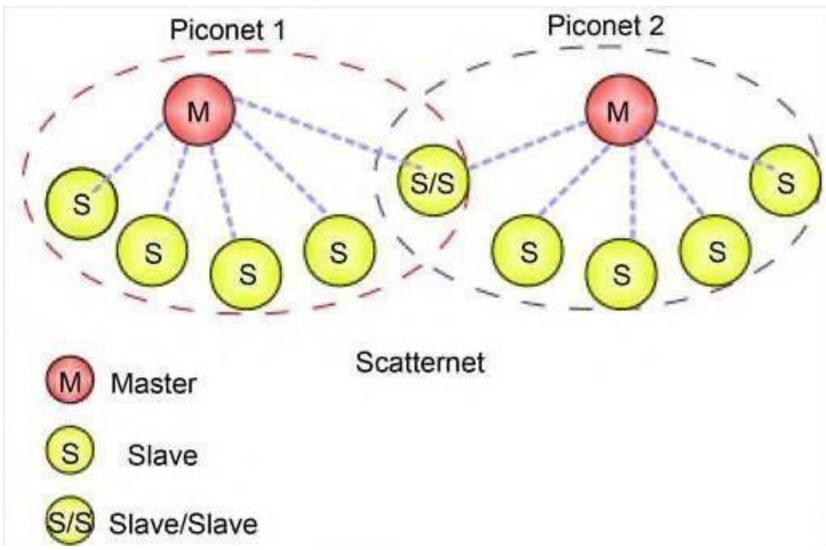
THEORY

Bluetooth:

Bluetooth is a widely used short-range wireless communication technology that enables devices to exchange data over the 2.4 GHz ISM frequency band. Originally developed to eliminate the need for physical cables between devices such as phones, headsets, and computers, Bluetooth has become an essential part of many IoT applications. Its ability to provide reliable communication within a range of about 10 meters, combined with relatively low power consumption, makes it suitable for various consumer and industrial use cases.

A notable advancement in this technology is Bluetooth Low Energy (BLE), introduced in Bluetooth version 4.0. BLE is optimized for applications that require minimal power consumption and only periodic communication. This makes it especially well-suited for battery-powered IoT devices like fitness trackers, medical sensors, smartwatches, and other wearables. Unlike traditional Bluetooth, which maintains continuous connections and higher energy usage, BLE operates in short bursts, conserving battery life while maintaining sufficient data transfer rates for typical IoT needs.

Bluetooth continues to play a critical role in the development of IoT solutions, particularly in areas such as healthcare monitoring, wearable devices, and smart home systems. It allows seamless interaction between mobile apps and IoT devices, enabling remote control and real-time data tracking. However, factors like its limited range, potential interference in crowded radio environments, and lower data throughput compared to alternatives like Wi-Fi must be considered when selecting it for specific applications. Still, its convenience, energy efficiency, and widespread device compatibility make Bluetooth a popular choice in the IoT ecosystem.



CONNECTIONS:

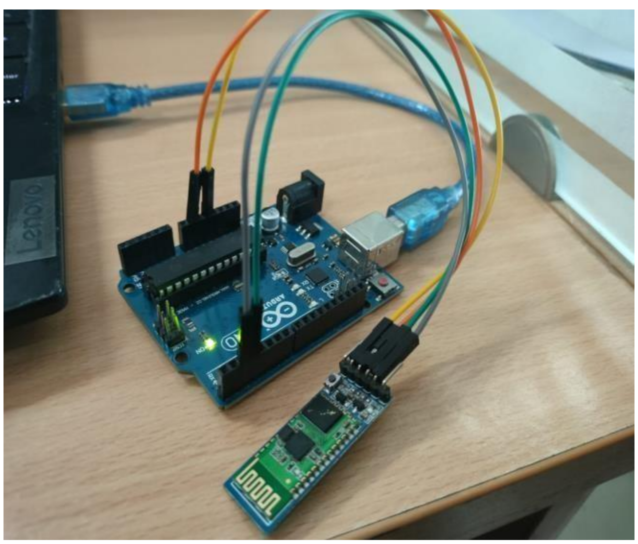
Arduino UNO Pin	Bluetooth Module	Arduino Development Board
VCC	5V	-
GND	GND	-
2	Tx	-
3	Rx	-
4	-	LED

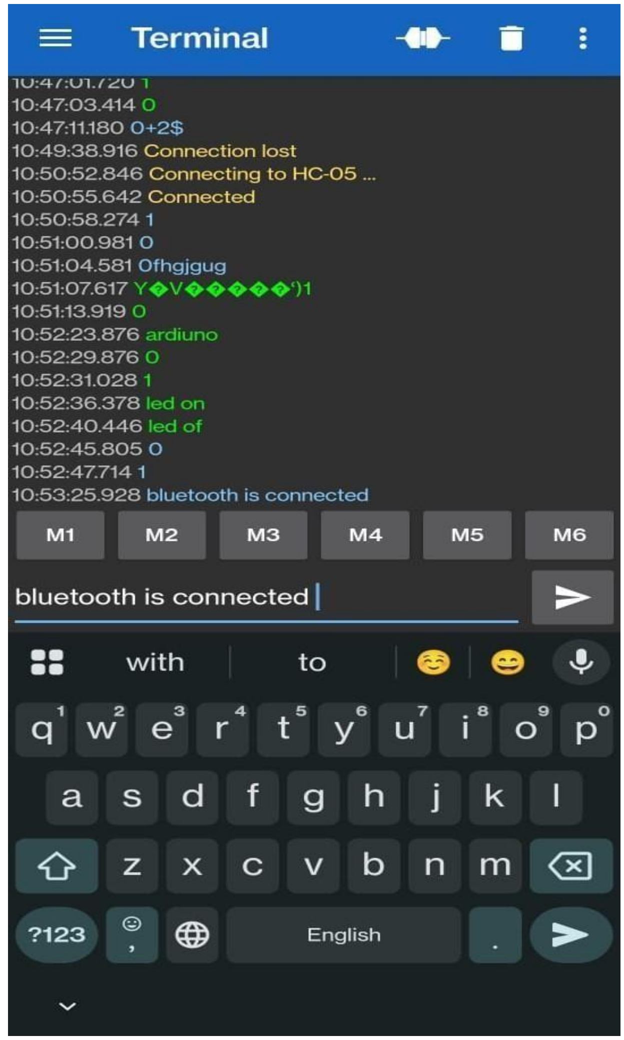
PROGRAM:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(3, 2); //HC-05 Tx & Rx is connected to Arduino 3 & 2 dpio
void setup(){
  Serial.begin(9600);
  mySerial.begin(9600);
  pinMode(13,OUTPUT);
  Serial.println("Initializing...");
  Serial.println("The device started, now you can pair it with bluetooth !");
}
void loop(){
  if(Serial.available())  {
  char data=Serial.read();
  Serial.print("Received: ");
  Serial.println(data);
  mySerial.write(data);
    if (data == '1') {
          digitalWrite(13, HIGH);
          Serial.println("LED ON");
    } else if (data == '0') {
          digitalWrite(13, LOW);
          Serial.println("LED OFF");
     }
  }
  if(mySerial.available()) {
        char data = mySerial.read();
```

```
        Serial.print("Received: ");
        Serial.println(data);
        if (data == '1') {
            digitalWrite(13, HIGH);
            Serial.println("LED ON");
        } else if (data == '0') {
            digitalWrite(13, LOW);
            Serial.println("LED OFF");
        }
}}
```

OUTPUT:





RESULT:

Thus, the different communication methods with IoT devices (Zigbee, GSM, Bluetooth) are studied successfully.

EXP NO.: 07	INTRODUCTION TO RASPBERRY PI PLATFORM AND PYTHON PROGRAMMING
DATE:	

AIM:

To study the Raspberry Pi platform and python programming.

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Hardware & Software Requirement	Quantity
1	Raspberry Pi 3 Model	1
2	HDMI Cable for connect raspberry pi to monitor	1
3	Jump wires	Few
4	LED	1

THEORY

Introduction to Raspberry Pi 5.3:

Raspberry Pi 5.3 is the latest version of the small, affordable computer developed by the Raspberry Pi Foundation. It is a single-board computer, meaning all the main components like the CPU, RAM, USB ports, HDMI, Ethernet, etc., are built into a single board.

Raspberry Pi 5.3 offers a performance boost compared to earlier versions. It includes:

- Faster processor (Broadcom BCM2712 quad-core Cortex-A76)
- Improved GPU for video output and light gaming
- More RAM options (up to 8GB)
- Dual 4K display support
- PCIe 2.0 support for advanced hardware connections
- Better power management and cooling system

PROCEDURE:

Step 1: Take your Raspberry Pi, 1 LED, and 2 jumper wires.

Step 2: Connect the long leg of the LED to GPIO 17 (Pin 11) using a jumper wire.

Step 3: Connect the short leg of the LED directly to GND (Pin 6) with another jumper wire.

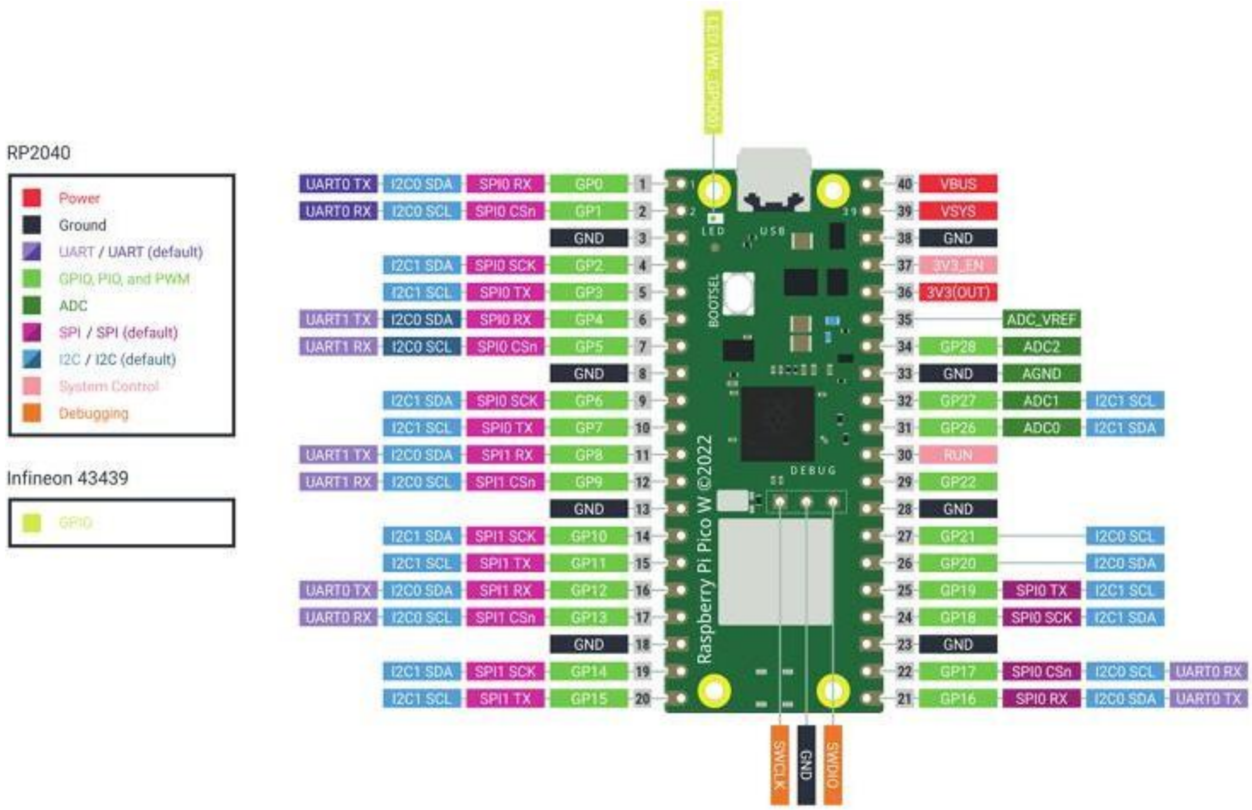
Step 4: Plug in HDMI, keyboard, and mouse, then power on your Pi.

Step 5: Open terminal and install GPIO library sudo apt install python3-rpi.gpio.

Step 6: Create a Python file and write a simple code to turn GPIO 17 ON and OFF.

Step 7: Run the program using python3 blink.py and watch the LED blink.

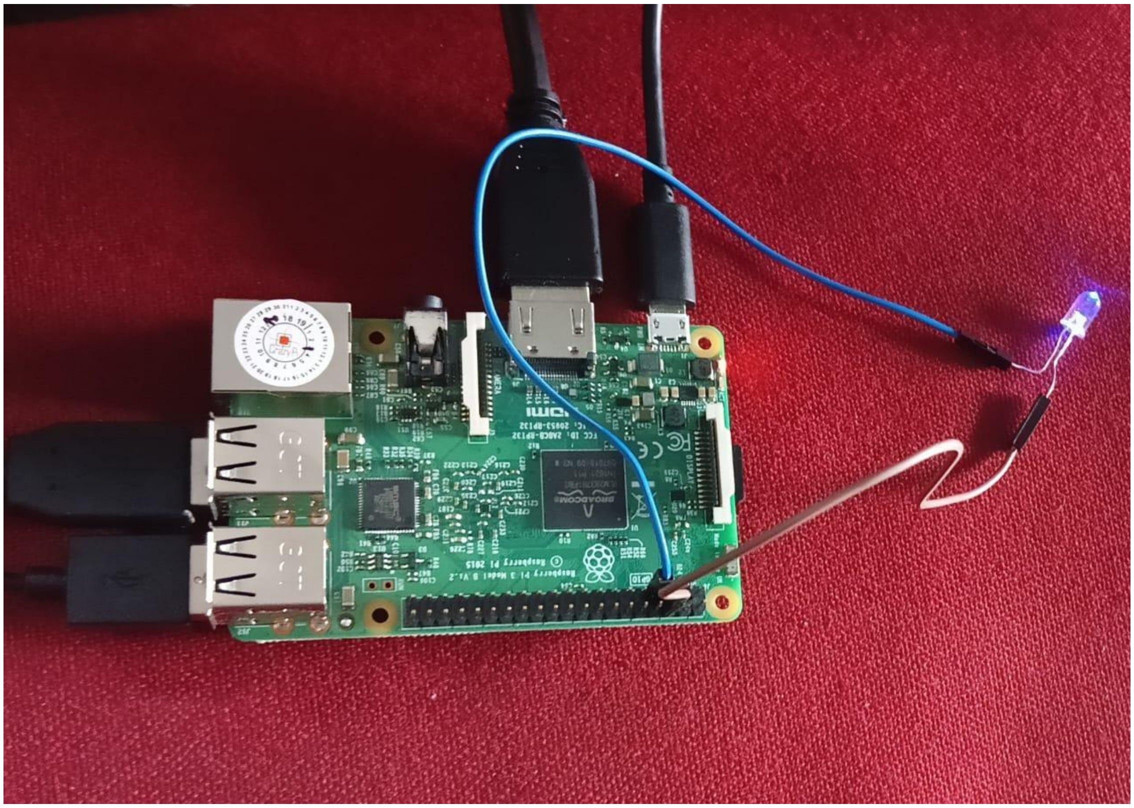
PIN DIAGRAM:



PROGRAM:

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT, initial=GPIO.LOW)
while True:
    GPIO.output(7, GPIO.HIGH)
    print("LED ON")
    sleep(1)
    GPIO.output(7, GPIO.LOW)
    print("LED OFF")
    sleep(1)
```

OUTPUT:



RESULT:

Thus, the Raspberry Pi Platform and python programming are studied successfully.

EXP NO.: 08	INTERFACING SENSORS WITH RASPBERRY PI
DATE:	

AIM:

To interface the IR sensor and Ultrasonic sensor with Raspberry Pico.

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Hardware & Software Requirement	Quantity
1	Thonny IDE	1
2	Raspberry Pi 3	1
3	Jump wires	Few
4	Micro USB Cable	1
5	IR Sensor	1

PROCEDURE:

Step 1: Connect a sensor to Raspberry Pi and write Python code to read the data.

Step 2: Create a ThingSpeak account, make a channel, and note the Write API Key.

Step 3: Install the requests library on Raspberry Pi using sudo pip install requests.

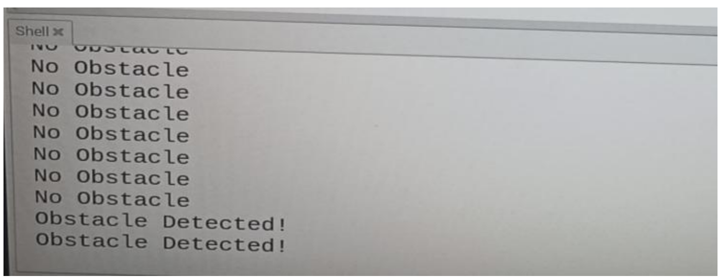
Step 4: Write Python code to send sensor data to ThingSpeak using the API Key.

Step 5: Run the code and check if data shows up on your ThingSpeak channel.

PROGRAM (IR SENSOR):

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
IR_PIN = 17
GPIO.setup(IR_PIN, GPIO.IN)
print("IR Sensor Test - Press Ctrl+C to exit")
try:
    while True:
        if GPIO.input(IR_PIN) == 0:
            print("Obstacle Detected!")
        else:
            print("No Obstacle")
        time.sleep(0.5)
except KeyboardInterrupt:
    print("Exiting Program")
finally:
    GPIO.cleanup()
```

OUTPUT:



RESULT:

Thus, the IR sensor and Ultrasound sensor are interfaced with Raspberry Pi executed successfully.

EXP NO.: 09	COMMUNICATE BETWEEN ARDUINO AND RASPBERRY PI USING ANY WIRELESS MEDIUM
DATE:	

AIM:

To study the program to Communicate between Arduino and Raspberry PI using any wireless medium (Bluetooth)

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Hardware & Software Requirement	Quantity
1	Thonny IDE	1
2	Raspberry Pi Pico Development Board	1
3	Jump wires	Few
4	Micro USB Cable	1
5	Arduino UNO Board	1

PROCEDURE:

Step 1: Connect Bluetooth modules to Arduino and Raspberry Pi Pico.

Step 2: Arduino sends 'A' and 'B' one after another every second.

Step 3: Raspberry Pi Pico receives data using UART.

Step 4: Check if received data is 'A' or 'B'.

Step 5: If 'A', turn LED ON; if 'B', turn LED OFF.

Step 6: Confirm LED status and data is received correctly.

PROGRAM:

MASTER- ARDUINO:

```
#include SoftwareSerial mySerial(2,3); //rx,tx
void setup() {
mySerial.begin(9600);
}
void loop() {
mySerial.write('A');
delay(1000);
mySerial.write('B');
delay(1000);
}
```

CONNECTIONS:

Arduino UNO Pin	Arduino Development Board	Bluetooth Module
2	-	Tx
3	-	Rx
-	GND	GND
-	5V	5V

SLAVE - RASPBERRY PI PICO

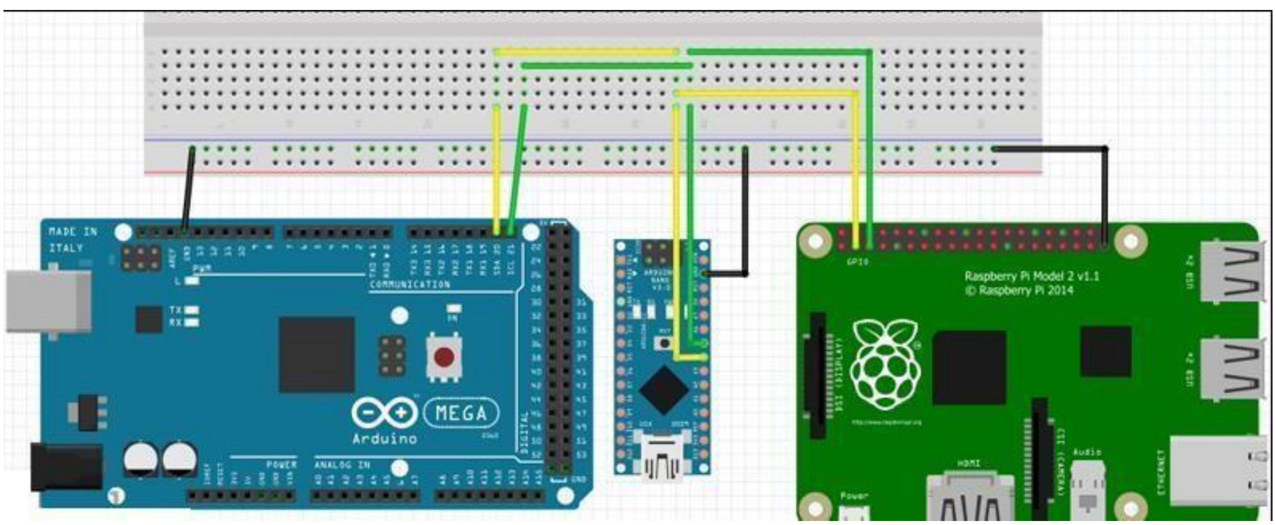
```
from machine import Pin, UART
uart = UART(0, 9600)
led = Pin(16, Pin.OUT)
while True:
if uart.any() > 0:
data = uart.read()
print(data)
if "A" in data:
led.value(1)
print('LED on \n')
uart.write('LED on \n')
elif "B" in data:
```

```
led.value(0)
print('LED off \n')
uart.write('LED off \n')
```

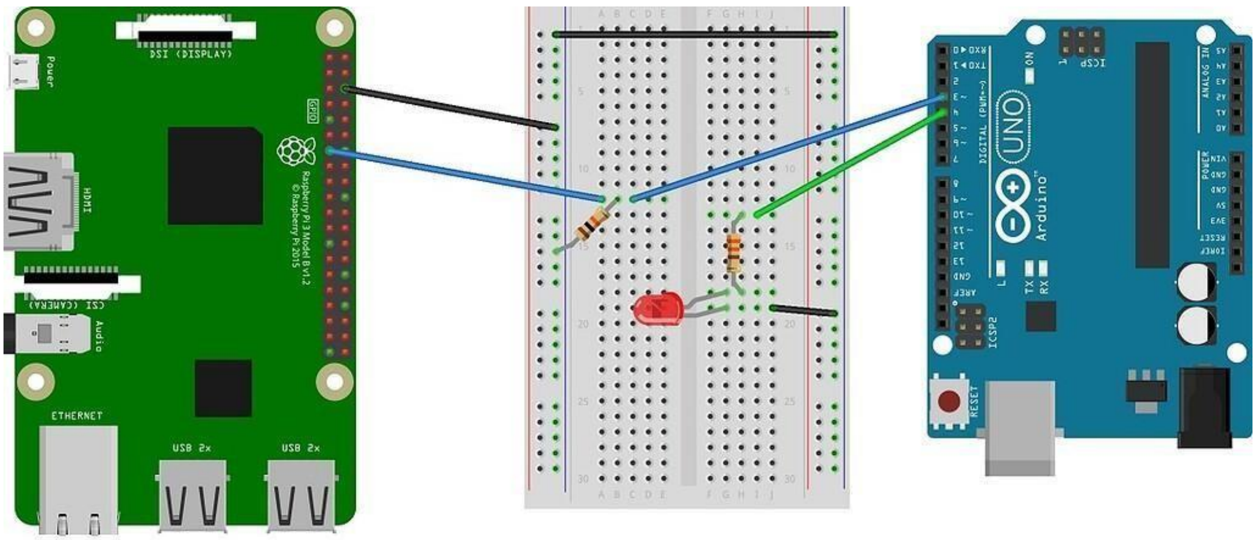
CONNECTIONS:

Raspberry Pi Pico Pin	Raspberry Pi Pico Development Board	Bluetooth Module
GP16	LED	-
VCC	-	+5V
GND	-	GND
GP1	-	Tx
GP0	-	Rx

OUTPUT:



INTERFACING FOR LED:



RESULT:

Thus the program to Communicate between Arduino and Raspberry PI using any wireless medium (Bluetooth) was written and executed successfully.

EXP NO.: 10	SETUP A CLOUD PLATFORM TO LOG THE DATA
DATE:	

AIM:

To set up a cloud platform to log the data from IoT devices.

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Software Requirement	Quantity
1	Thingspeak	1

CLOUD PLATFORM – THINGSPEAK

ThingSpeak is an open-source Internet of Things (IoT) cloud platform that enables users to collect, store, analyze, and visualize sensor data in real time. Developed by MathWorks, ThingSpeak integrates seamlessly with MATLAB for advanced analytics and predictive modeling, making it a powerful tool for engineers, researchers, and developers working on IoT applications. It allows users to set up channels to collect data from sensors or devices via HTTP or MQTT protocols, and it supports the creation of visualizations such as line graphs, bar charts, and maps directly within the platform. With built-in features like data aggregation, event scheduling, and alerting (via email or Twitter), ThingSpeak facilitates rapid prototyping and deployment of IoT systems without the need to build a backend from scratch.

In addition to its real-time data handling capabilities, ThingSpeak is widely appreciated for its accessibility and flexibility. It offers both free and paid versions, where the free tier is ideal for academic and hobbyist projects, and the paid licenses support higher data rates and commercial usage. The platform also supports integration with third-party services and devices such as Arduino, Raspberry Pi, and ESP8266, making it highly versatile for a broad range of IoT projects. Its MATLAB analytics integration allows users to process and analyze data directly on the cloud, enabling applications like predictive maintenance, anomaly detection, and machine learning-driven insights. Overall, ThingSpeak stands out as a comprehensive and user-friendly IoT cloud solution that bridges hardware, data analytics, and visualization effectively.

PROCEDURE:

Step 1: go to thingspeak.com and click sign up to create a free mathworks account by entering your email, name, and password, then verify your email to activate the account.

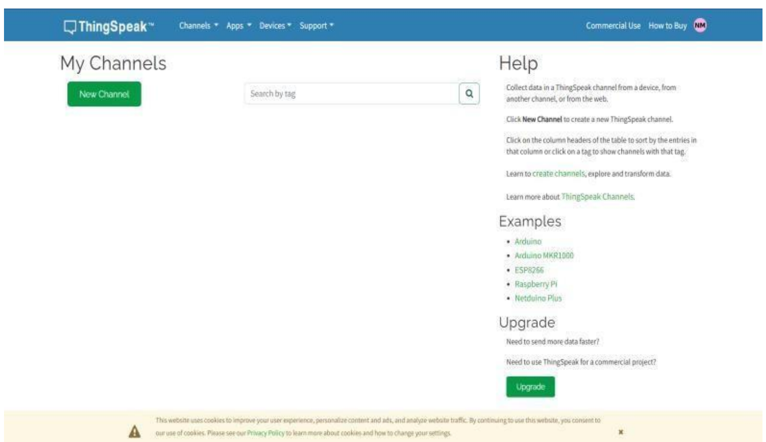
Step 2: after activating the account, log in to thingspeak.com using your mathworks credentials to access the main dashboard.

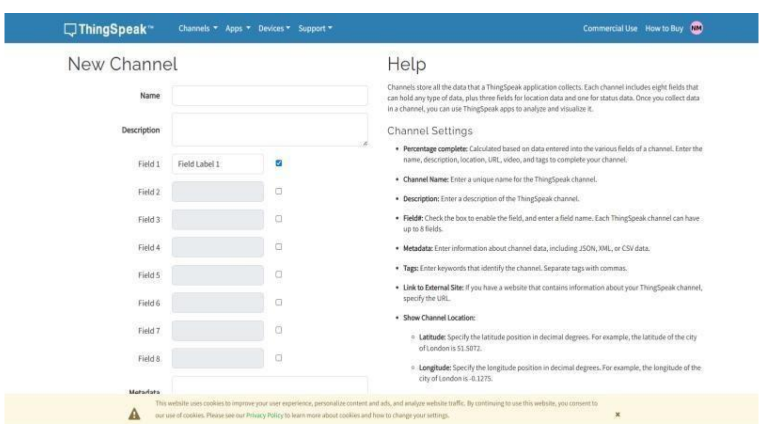
Step 3: click on channels, select my channels, and then click new channel to begin creating your own channel.

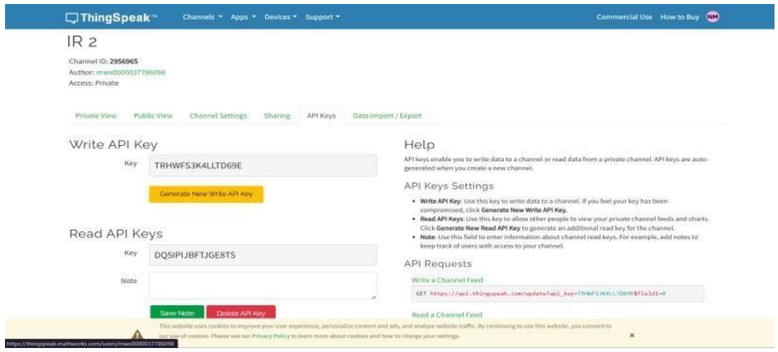
Step 4: enter a name for your channel, optionally add a description, enable the fields you need to store sensor data, and scroll down to click save channel.

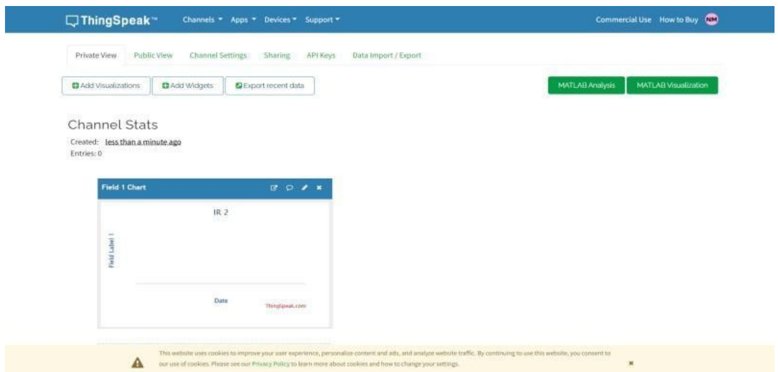
Step 5: once the channel is created, note the channel id and api keys which you will use to send or retrieve data, and now you can start sending data from your device and visualizing it with charts on thingspeak.

OUTPUT:









RESULT:

Thus, the cloud program to log the data from IoT devices is set up successfully.

EXP NO.: 11	LOG DATA USING RASPBERRY PI AND UPLOAD TO THE CLOUD PLATFORM
DATE:	

AIM:

To write and execute the program Log Data using Raspberry PI and upload it to the cloud platform

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Hardware & Software Requirement	Quantity
1	Thonny IDE	1
2	Raspberry Pi Pico Development Board	1
3	Jump wires	Few
4	Micro USB Cable	1

PROCEDURE:

step 1: connect the ir sensor to raspberry pi pin 17 and install needed libraries.

step 2: set the pin as input and check if obstacle is detected or not.

step 3: get the thingspeak api key ready and set value 1 for obstacle, 0 for no obstacle.

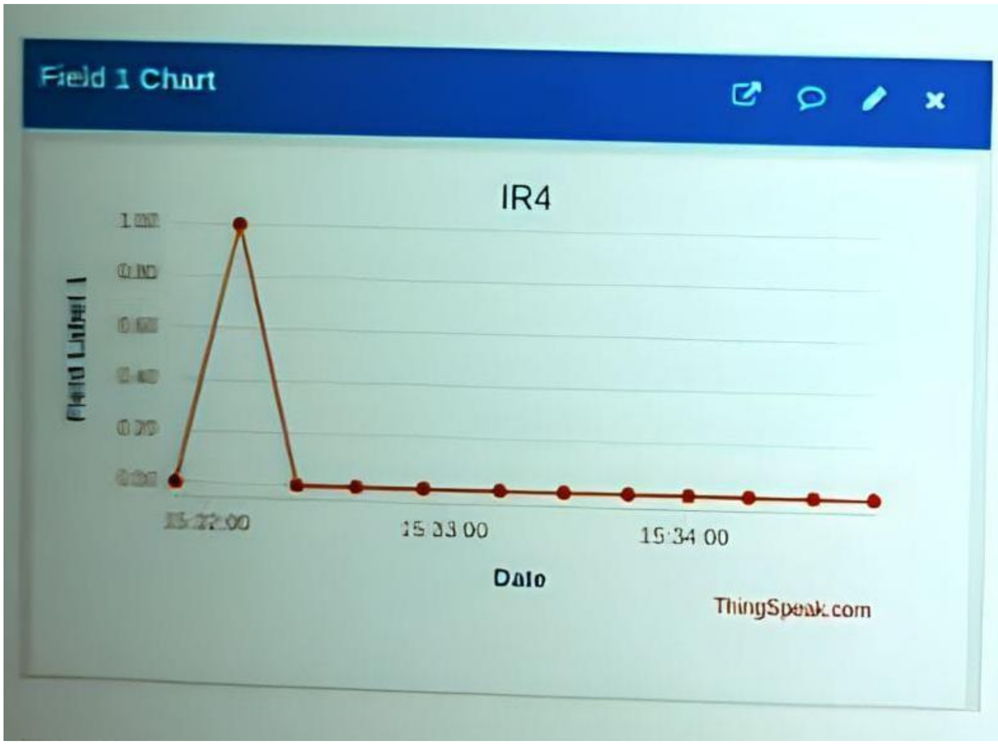
step 4: send the value to thingspeak using a get request.

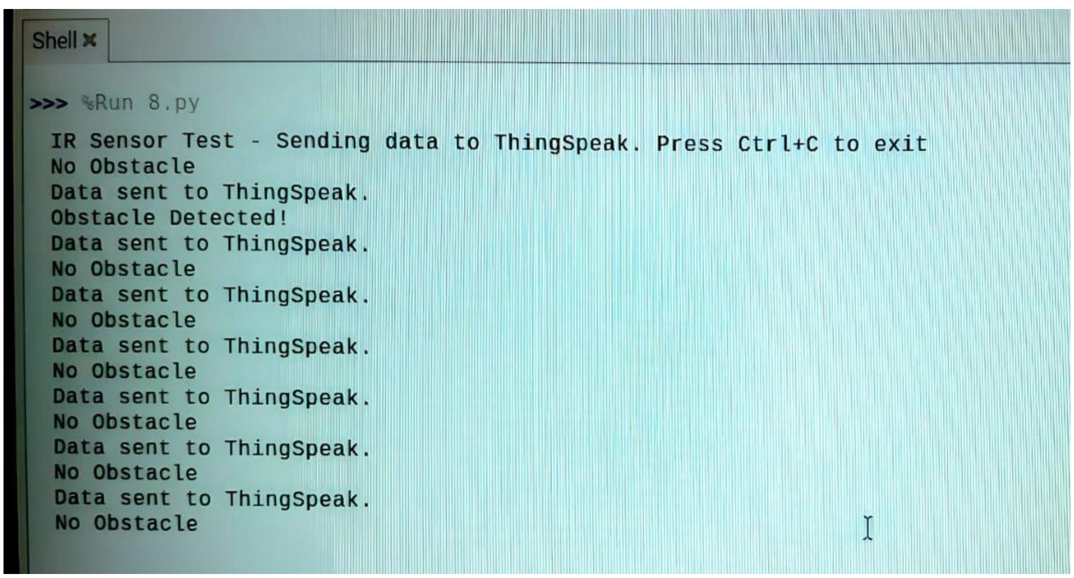
step 5: keep repeating and clean up gpio when you stop the program.

PROGRAM:

```
import RPi.GPIO as GPIO
import time
import requests
API_KEY = "Z7FKADF66448MH"
URL = f"https://api.thingspeak.com/update"
GPIO.setmode(GPIO.BCM)
IR_PIN = 17
GPIO.setup(IR_PIN, GPIO.IN)
print("IR Sensor Test - Sending data to Thingspeak. Press Ctrl+C to exit")
try:
    while True:
        if GPIO.input(IR_PIN) == 0:
            obstacle = 1
            print("Obstacle Detected!")
        else:
            obstacle = 0
            print("No Obstacle")
        payload = {'api_key': API_KEY, 'field1': obstacle}
        response = requests.get(URL, params=payload)
        if response.status_code == 200:
            print("Data sent to Thingspeak.")
        else:
            print(f"Failed to send data. HTTP {response.status_code}")
        time.sleep(0.5)
except KeyboardInterrupt:
    print("Exiting Program")
finally:
    GPIO.cleanup()
```

OUTPUT:





RESULT:

Thus, the program to execute Log Data using Raspberry pi and uploading it to the cloud platform is done successfully.

EXP NO.: 12	DESIGN AN IOT BASED SYSTEM
DATE:	

AIM:

To write a program using sensors for carparking assist.

HARDWARE & SOFTWARE TOOLS REQUIRED:

S.NO	Hardware & Software Requirement	Quantity
1	PC with windows	1
2	Arduino 1.6.5	1
3	Proteus 8.0 with Senor Library	1

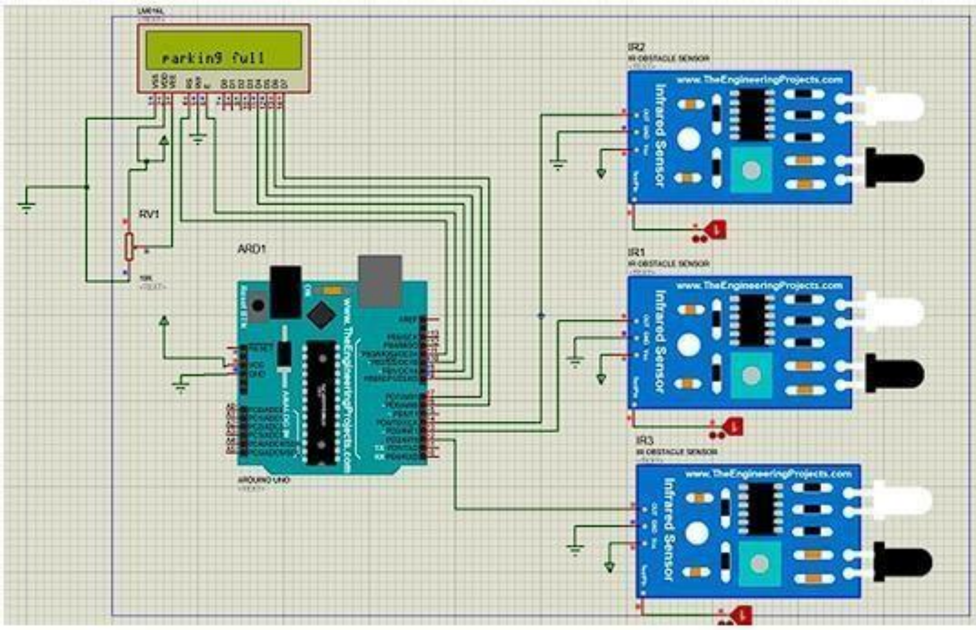
THEORY :

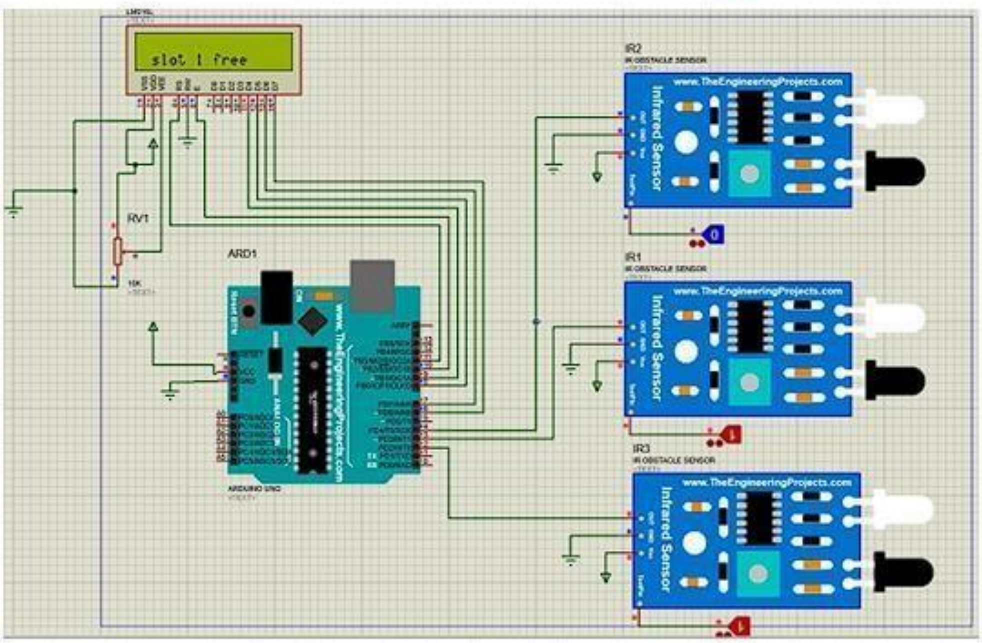
The infrared Obstacle Sensor Module has a built-in IR transmitter and IR receiver that sends out IR energy and looks for reflected IR energy to detect the presence of any obstacle in front of the sensor module. IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations. The program will give feedback to the driver based on the proximity of the car to the obstacle. We'll use an LED to indicate the presence of Free slot. The usage of this experiment becomes the major part in numerous and exclusive Malls, theatres, and Big Residencies.

PROGRAM:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(11, 10, 9, 8, 7, 6);
int sen1 = 4;
int  sen2  =  3;
int  sen3  =  2;
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(sen1, INPUT);
  pinMode(sen2, INPUT);
  pinMode(sen3, INPUT);
}
void loop() {
  int irValue1 = digitalRead(sen1);
  int irValue2 = digitalRead(sen2);
  int irValue3 = digitalRead(sen3);
  if (irValue1 == LOW && irValue2 == HIGH && irValue3 == HIGH) {
    lcd.clear();
    lcd.setCursor(1, 1);
    lcd.print("slot 1 free");
    delay(100);
  }
  else if (irValue2 == LOW && irValue1 == HIGH && irValue3 == HIGH) {
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("slot 2 free");
    delay(100);
  }
  else if (irValue3 == LOW && irValue1 == HIGH && irValue2 == HIGH) {
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("slot 3 free");
    delay(100);
  }
  else if (irValue1 == LOW || irValue2 == LOW || irValue3 == LOW) {
    lcd.clear();
    lcd.setCursor(1, 1);
    lcd.print("welcome");
    delay(100);
  }
  else if (irValue1 == HIGH && irValue2 == HIGH && irValue3 == HIGH) {
    lcd.clear();
    lcd.setCursor(1, 1);
    lcd.print("parking full");
    delay(100);
  }
}
```

OUTPUT:





RESULT:

Thus, the program for Carparking assist using IR Sensors was designed successfully.