SCHEME : K

Name : _____

Roll No. : _____ Year : 20___ 20___

Exam Seat No. : _____

LABORATORY MANUAL FOR

INTERNET OF THINGS- 314006

COMPUTER ENGINEERING GROUP

MAHARASHTRA STATE BOARD OF
TECHNICAL EDUCATION, MUMBAI
(Autonomous) (ISO 9001: 2015) (ISO/IEC 27001:2013)

Vision

To ensure that the Diploma level Technical Education constantly matches the latest requirements of Technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

Mission

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the challenging technological and environmental changes

Quality Policy

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

Core Values

MSBTE believes in the following:

- Skill development in line with industry requirements
- Industry readiness and improved employability of Diploma holders
- Synergistic relationship with industry
- Collective and Cooperative development of all stake holders
- Technological interventions in societal development
- Access to uniform quality technical education

A Laboratory Manual
For

Internet of Things

(314006)

Semester-IV

Diploma in Information Technology/ Computer Science & Information Technology

Maharashtra State Board of Technical Education, Mumbai

(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)

'K' Scheme Curriculum

Maharashtra State Board of Technical Education, Mumbai
(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)
4th Floor, Government Polytechnic Building
49, Kherwadi, Bandra (East), Mumbai – 400051

Maharashtra State Board of Technical Education
Certificate

This is to certify that Mr. /Ms. …… Roll No…… of the fourth Semester of Diploma in …… Engineering/Technology (Program Code - .....4K) of the Institute …… (Inst. Code……) has completed the practical work satisfactorily for the course Internet of Things (Course Code: 314006) for the academic year 20….. – 20…..... as prescribed in the curriculum.

Place ……. Enrollment No……

Date: …..... Exam Seat No. ……..…..

Course Teacher **Head of the Department** **Principal**

Seal of the Institute

Preface

Internet of Things (314006) laboratory manual is meticulously crafted to equip fourth semester diploma engineering students with valuable practical learning experiences aligned with MSBTE 'K' Scheme Curriculum.

The primary objective of this manual is to enhance technical skills of diploma engineering students which are vital to excel at the workplace. To achieve this, each practical is mapped with prescribed theory learning outcomes (TLOs), lab learning outcomes (LLOs) and course outcomes (COs). Course facilitators can adopt suitable pedagogical methods to impart the course with an aim to achieve the prescribed course outcomes effectively.

This laboratory manual on the Internet of Things is designed to provide students with a comprehensive understanding of the core concepts, technologies, and practical skills necessary to excel in this rapidly growing field. Through a series of hands-on experiments and projects, students will explore the fundamental aspects of IoT, including sensor integration, data acquisition, communication protocols, data analysis, and cloud computing.

This manual is intended for students of engineering who are keen to explore the dynamic world of IoT. It serves as a foundational resource for academic coursework.

Program Outcomes (POs) to be achieved through Practical:

PO1	Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems..
PO2	Problem analysis: Identify and analyses well-defined engineering problems using codified standard methods.
PO3	Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.
PO4	Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.
PO5	Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.
PO6	Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.
PO7	Life-long learning: Ability to analyses individual needs and engage in updating in the context of technological changes.

List of Relevant Skills

Knowledge of IoT can inspire students to develop innovative solutions IoT provides a rich platform for hands-on projects, allowing students to apply theoretical knowledge in real-world scenarios.

- Programming and Software Development: Essential for programming microcontrollers like Arduino, Raspberry Pi.
- Networking and Communication Protocols: Understanding and using MQTT, CoAP, and HTTP/HTTPS for device communication.
- Hardware and Electronics: Knowledge of different types of sensors and actuators used in IoT projects. Basics of designing and understanding electronic circuits.

Practical Course Outcome Matrix

Course Outcomes (COs)

CO1	Integrate hardware and software for simple IoT applications.
CO2	Create IoT applications by interfacing various sensors and embedded boards.
CO3	Create IoT applications by interfacing various actuators and embedded boards.
CO4	Develop IoT applications using IoT networking devices
CO5	Develop database based IoT application by integrating sensors with single board computer.

Sr. No.	Title of the Experiment	CO1	CO2	CO3	CO4	CO5
1	* Install any one embedded system(ex- Arduino IDE)and execute program to turn LED ON/OFF using delay	✓				
2	Change the color of LED	✓				
3	Control the brightness of LED using PWM Techniques	✓				
4	* Detect the presence or absence of Light using LDR Sensor		✓			
5	Measure the temperature of the object		✓			
6	Sense the touch of finger when it is placed on board		✓			
7	Detect the obstacle using IR sensor		✓			
8	* Measure the Distance between sensor and object using ultrasonic sensor		✓			
9	Detect the presence of Gas		✓			
10	Detect the vibration of an object using vibration detector sensor SW-420 with Arduino		✓			
11	Change the status of Buzzer ON/OFF	✓		✓		
12	* Display Humidity and Temperature on LCD using DHT11 sensor		✓	✓		
13	* Display the message as per detection of motion of object		✓	✓		
14	* Control relay state based on input from IR Sensor		✓	✓		
15	* Switch the LED ON/OFF on detection of obstacles using PIR sensor		✓	✓		

Sr. No.	Title of the Experiment	CO1	CO2	CO3	CO4	CO5
16	* Measure the Distance between sensor and object and sound buzzer when obstacle is detected in some specified range of distance		✓	✓		
17	Activate the alarm system if smoke detected		✓	✓		
18	* Display percentage of moisture in soil using soil moisture sensor		✓	✓		
19	Upon detection of fire, activate the LED indicator and initiate the alarm system		✓	✓		
20	* Display temperature value on serial monitor			✓		
21	* Activate the Melody audio output utilizing a Piezoelectric speaker.			✓		
22	* Control action using Relay based on temperature value			✓		
23	* Display 0 to 9 numbers continuously on seven segment display			✓		
24	Display simple message on I2C LCD			✓		
25	* Display POT value of potentiometer on LCD			✓		
26	* Transfer sensor collected data to smartphone using Bluetooth				✓	
27	Display the message on serial monitor when image is captured				✓	
28	* Create Web based IoT application using NodeMCU/ESP32/Raspberry Pi to display Temperature on Web Browser				✓	
29	* Setup Raspberry Pi as an Single board computer b with following accessories: a display a cable to connect Raspberry Pi to display a keyboard a mouse SD card					✓
30	* Install MariaDB database in Raspberry Pi and execute basic SQL querie					✓

Guidelines to Teachers

1. Teacher is expected to refer complete curriculum document and follow guidelines for implementation
2. Teacher should provide the guideline with demonstration of practical to the students with all features.
3. Teacher shall explain prior concepts to the students before starting of each practical.
4. Involve students in performance of each practical.
5. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
6. Teachers should give opportunity to students for hands on experience after the demonstration.
7. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected the students by the industry.
8. Give practical assignment and assess the performance of students based on task assigned to check whether it is as per the instructions.
9. Assess the skill achievement of the students and COs of each unit. Teachers should align the explanation of the topic to teaching learning outcome (TLOs).

Instructions for Students

1. Students should listen carefully to the lecture given by the teacher about course, curriculum, learning structure, skills to be developed.
2. Student shall read lab manual of related practical to be conducted before performing the practical
3. Students shall attempt to develop related hand-on skills and gain confidence.
4. Students should accomplish the requisites of Teamwork, Collaboration and Group Dynamics during the practical sessions.
5. Students shall develop the habits of evolving more ideas, innovations, skills etc. those included in scope of manual.
6. Student should develop habit to submit the practical on date and time.

Content Page

List of Practical's and Formative Assessment Sheet

Sr. No	Practical Title	Date of Performance	Date of Submission	Assessment Marks (25)	Teacher's Sign	Remark
1	* Install any one embedded system(ex- Arduino IDE)and execute program to turn LED ON/OFF using delay					
2	Change the colour of LED					
3	Control the brightness of LED using PWM Techniques					
4	* Detect the presence or absence of Light using LDR Sensor					
5	Measure the temperature of the object					
6	Sense the touch of finger when it is placed on Board					
7	Detect the obstacle using IR sensor					
8	* Measure the Distance between sensor and object using ultrasonic sensor					
9	Detect the presence of Gas					
10	Detect the vibration of an object using vibration detector sensor SW-420 with Arduino					
11	Change the status of Buzzer ON/OFF					
12	* Display Humidity and Temperature on LCD using DHT11 sensor					
13	* Display the message as per detection of motion of object					
14	* Control relay state based on input from IR Sensor					

Sr. No	Practical Title	Date of Performance	Date of Submission	Assessment Marks (25)	Teacher's Sign	Remark
15	* Switch the LED ON/OFF on detection of obstacles using PIR sensor					
16	* Measure the Distance between sensor and object and ring the buzzer when obstacle is detected in some specified range of distance					
17	Activate the alarm system if smoke detected					
18	* Display percentage of moisture in soil using soil moisture sensor					
19	Upon detection of fire, activate the LED indicator and initiate the alarm system					
20	* Display temperature value on serial monitor					
21	* Activate the Melody audio output utilizing a Piezoelectric speaker.					
22	* Control action using Relay based on temperature value					
23	* Display 0 to 9 numbers continuously on seven segment display					
24	Display simple message on I2C LCD					
25	* Display POT value of potentiometer on LCD					
26	* Transfer sensor collected data to smartphone using Bluetooth					
27	Display the message on serial monitor when image is captured					
28	* Create Web based IoT application using Node MCU/ESP32/Raspberry Pi to display Temperature on Web Browser					

Sr. No	Practical Title	Date of Performance	Date of Submission	Assessment Marks (25)	Teacher's Sign	Remark
29	* Setup Raspberry Pi as an Single board computer b with following accessories: a display a cable to connect Raspberry Pi to display a keyboard a mouse SD card					
30	* Install Maria DB database in Raspberry Pi and execute basic SQL queries					
				Total		

***Total marks to be transferred to proforma published by MSBTE**

Note:

- '*' Marked Practical (LLOs) Are mandatory.
- Minimum 80% of above list of lab experiment are to be performed.
- Judicial mix of LLOs is to be performed to achieve desired outcomes.

Practical No. 1: Install any one embedded system (ex- Arduino IDE) and execute program to turn LED ON/OFF using delay

I. Practical Significance

To install any one embedded system (ex- Arduino IDE) and execute program to turn LED ON/OFF using delay. You can use various software and techniques.

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO1 - Integrate hardware and software for simple IoT applications.

IV. Laboratory Learning Outcome(s)

LLO 1.1 Install any embedded system.

LLO 1.2 Write simple Arduino program using Arduino Uno IDE.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

• Arduino in IoT: In IoT applications the Arduino is used to collect the data from the sensors/devices to send it to the internet and receives data for purpose of control of actuators.

• Arduino IDE: The Arduino Software (IDE) is easy-to-use and is based on the Processing programming environment. The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board

VII. Actual Circuit diagram used in laboratory with related equipment rating

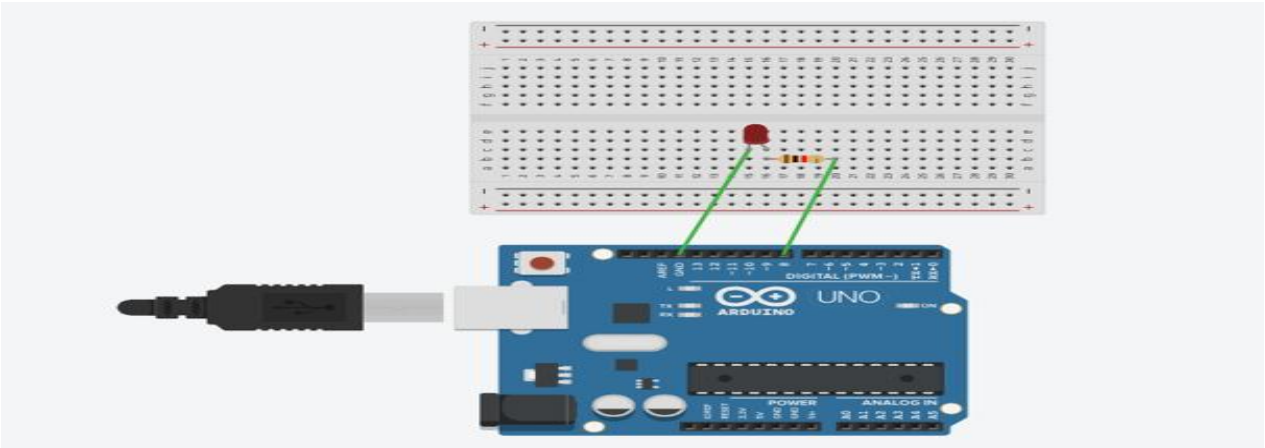


Fig 1.1.Ciruit diagram to turn LED ON/OFF using delay

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Led		01
4	Resistor	1K	01

IX. Precautions to be followed

Use always current limiting resistor before LED connected to Arduino.

X. Procedure

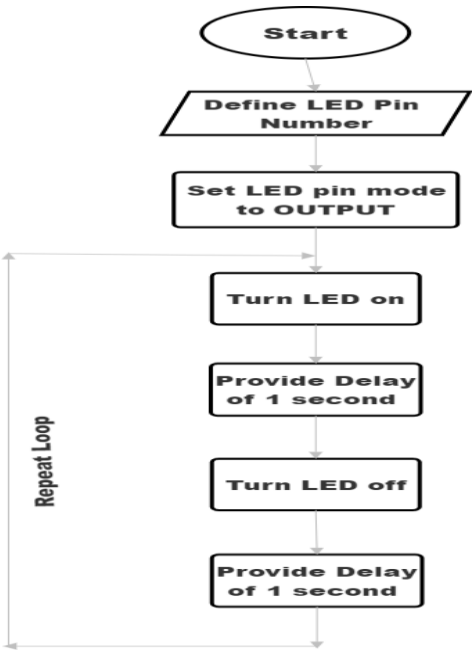
1. Interface LED to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the LED on/ off status.

XI. Sample program

Step 1: Algorithm

i. Define the pin number for the LED.
ii. Set the LED pin mode to Output.
iii. Inside the Loop Turn the LED ON.
iv. Provide a delay for 1second.
v. Turn the LED off and wait for 1second.
vi. Repeat the loop indefinitely.

Step 2: Flowchart



Step 3: Program

```
int led = 8;

void setup() {
      pinMode (led, OUTPUT);
}
void loop() {
      digitalWrite(led, HIGH);
      delay (1000);
      digitalWrite(led, LOW);
     delay(1000);
}
```

XII. Results (Output of the Program)

……

……

……

……

……

……

……

XIII Conclusion

……

……

……

……

……

XIV Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. State applications of IOT

……

……

……

……

……

2. Draw IOT architecture

……

……

……

……

XV: References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)
3. Cornel M Amariei: Arduino Development Cookbook

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 2: Change the colour of LED

I. Practical Significance

To change colour of RGB LED

.

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO1 - Integrate hardware and software for simple IoT applications.

IV. Laboratory Learning Outcome(s)

LLO 2.1 Interface RGB LED with Arduino

LLO 2.2 Write program to change the color of LED

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

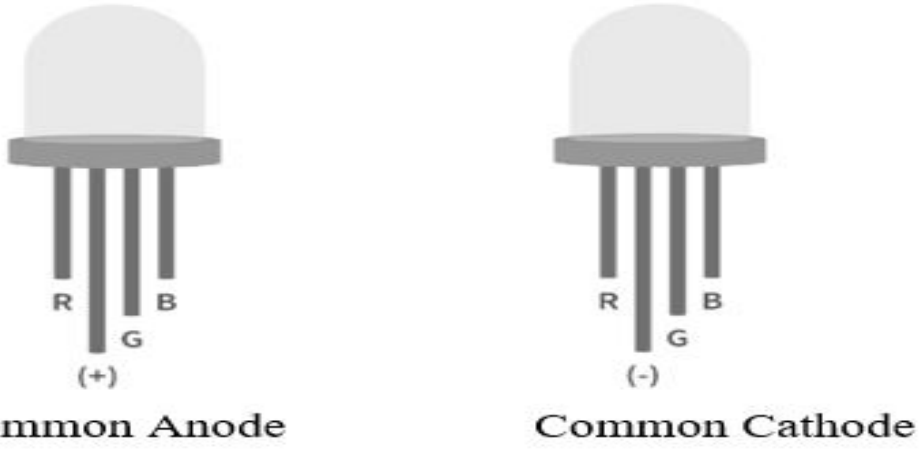
c. Follow ethical practices.

VI.Relevant Theoretical Background

There are actually two types of RGB LED's; the common cathode one and the common anode one. In the common cathode RGB led, the cathode of all the LED's is common and we give PWM signals to the anode of LED's while in the common anode RGB led, the anode of all the LED's is common and we give PWM signals to the cathode of LED's. To change colour of RGB led, we can use the PWM pins of Arduino

Structure of Common Anode and Common Cathode LED:

The structure of Common Anode and Common Cathode LED consists of 4 terminals, where the first terminal is "R" the second terminal is "Anode +" or "Cathode –", the third terminal is "G" and the fourth terminal is "B" as shown below



VII. Actual Circuit diagram used in laboratory with related equipment rating

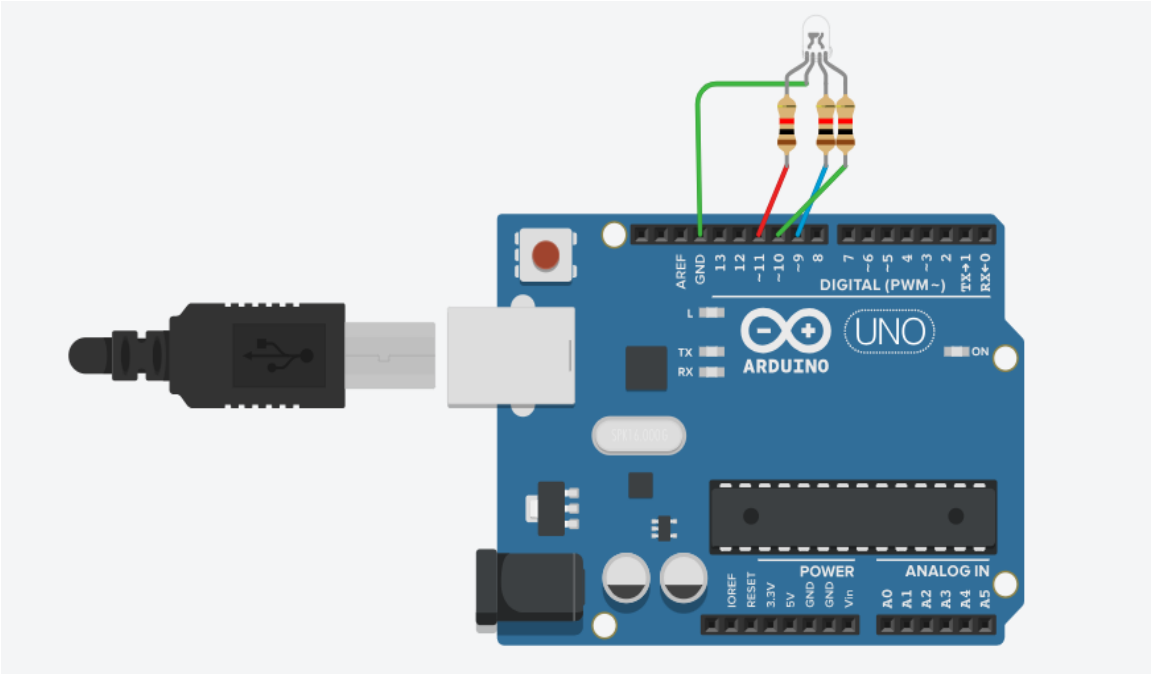


Fig 2.1.Ciruit diagram to interfacing RGB LED

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	RGB LED		01

IX Precautions to be followed

Use always current limiting resistor before LED connected to Arduino.

X. Procedure

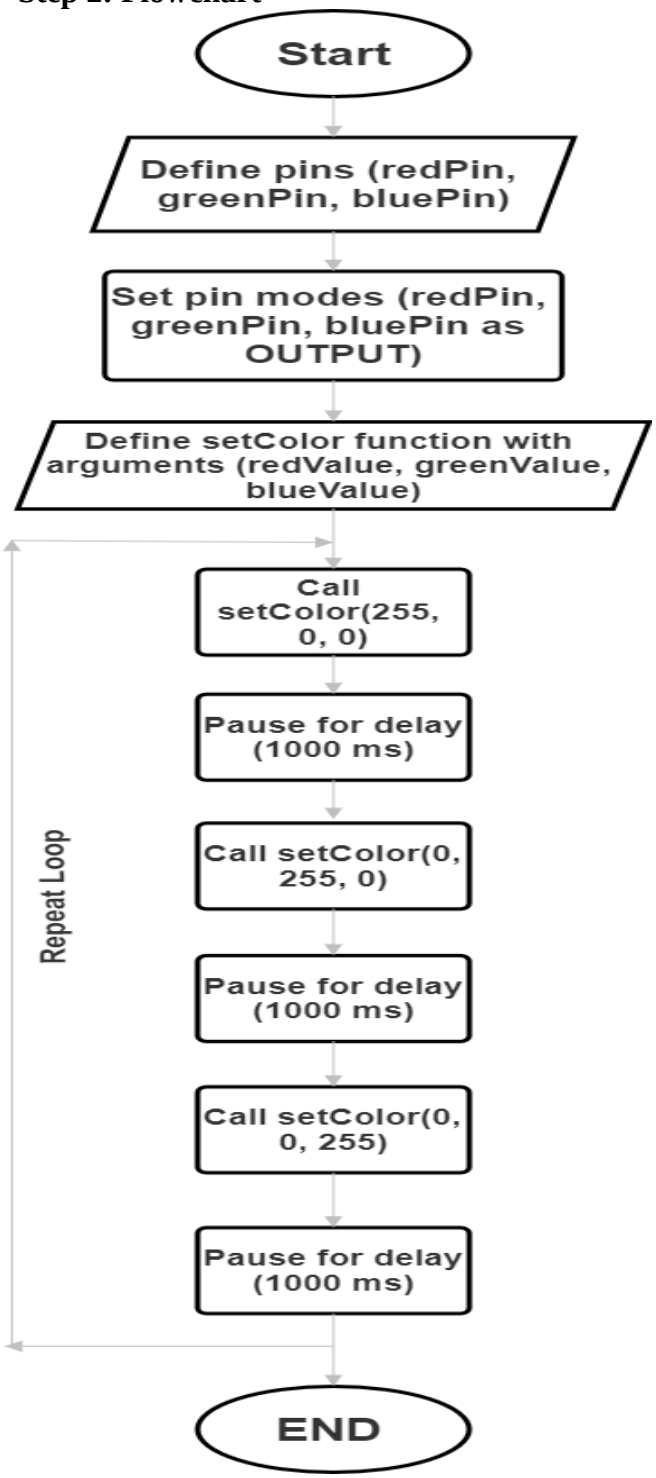
1. Interface RGB LED to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the RGB LED status.

XI. Sample program

Step 1: Algorithm

1. Assign pins for red, green, and blue LEDs.
2. Configure pins as outputs.
3. Map RGB values to corresponding pins.
4. Pause to display each color for a set duration.
5. Continuously cycle through colors.

Step 2: Flowchart



Step 3: Program

```
 int redPin= 11;
int greenPin = 10;
int bluePin = 9;

void setup() {
      pinMode(redPin, OUTPUT);
      pinMode(greenPin, OUTPUT);
      pinMode(bluePin, OUTPUT);
}
void loop() {
       setColor(255, 0, 0); // Red Color
       delay(1000);
       setColor(0, 255, 0); // Green Color
       delay(1000);
       setColor(0, 0, 255); // Blue Color
       delay(1000);
      setColor(170, 0, 255); // Purple Color
      delay(1000);
   }
void setColor(int redValue, int greenValue, int blueValue) {
        analogWrite(redPin, redValue);
        analogWrite(greenPin, greenValue);
        analogWrite(bluePin, blueValue);
}
```

XII Results (Output of the Program)

……

……

……

……

……

XIII Conclusion

……

……

……

……

……

XIV Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. How the different colors can be obtained using RGB LED?

……

……

……

……

……

XV References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook
3. Instructables.com/RGB-LED-Interfacing-With-Arduino/

XVI Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 3: Control the brightness of LED using PWM Techniques

I. Practical Significance

To Control the brightness of LED using PWM Techniques

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO1 - Integrate hardware and software for simple IoT applications

IV. . Laboratory Learning Outcome(s)

LLO 3.1 Interface Potentiometer and LED with Arduino.
LLO 3.2 Write a program to control the brightness of LED.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

The Arduino PWM is very useful for controlling things. We can control the brightness of a led, speed of a motor, direction of a Servo motor, and many other things using PWM.

Pulse Width Modulation or PWM, is a technique to generate an analog like signal within a digital pin.

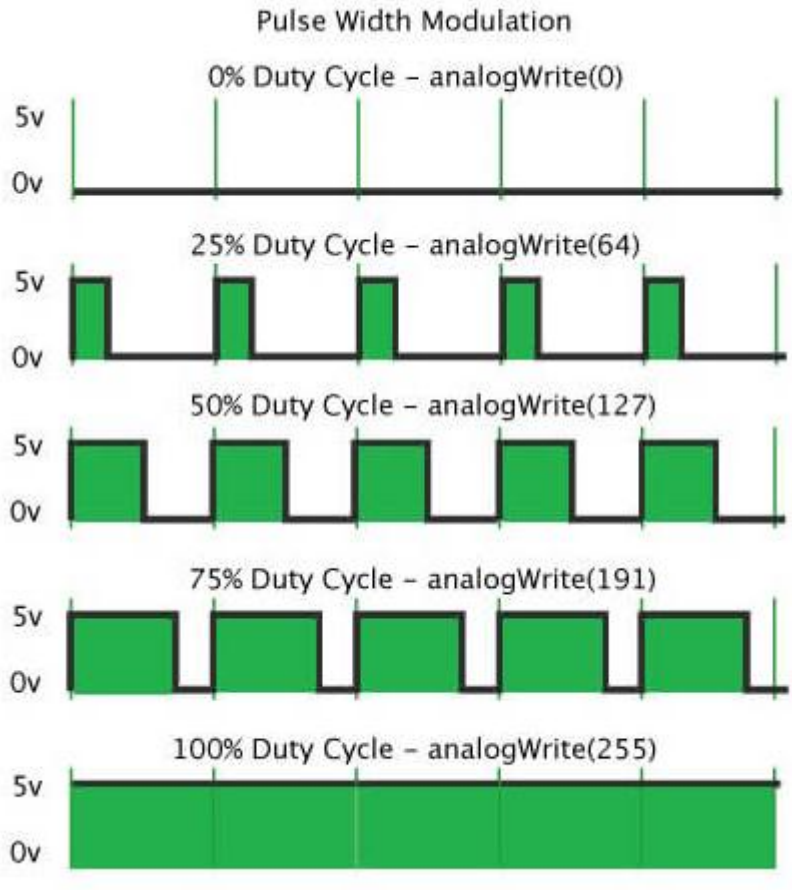
Here are some key terms related to PWM.

Period: It is the time required to complete a full cycle. The period of a PWM signal is inverse of the PWM frequency.

TON (On Time): It is the time when the signal remains high within a period of time.

TOFF (Off Time): It is the time when the signal remains low within a period of time.

Duty Cycle: It is the percentage of time when the signal was high during the time of a period.



To get a 50% or 20% duty cycle on Arduino code, we will use Arduino's analogWrite() function.

Analog Write () works on a scale of 0 – 255. That means we can have 256 different voltages from 0v to 5v. And the difference between each step would be 5v/255 = 0.0196v.
So we can have voltages like 0v, 0.0196v, 0.0392v, …. 5v. It is almost like an analog signal.

analogWrite(0) gives a signal of 0% duty cycle i.e 0v output.
analogWrite(50) gives a signal of ~ 20% duty cycle i.e 1v output.
analogWrite(63) gives a signal of 25% duty cycle i.e 1.25v output.
analogWrite(127) gives a signal of 50% duty cycle i.e 2.5v output.
Arduino Uno has six PWM pins, pin 3, 5, 6, 9, 10 and11. Pin 5 and 6 have a frequency of 980Hz and pins 3,9,10 and 11 have a frequency of 490Hz.

a) Control the brightness of LED using PWM Techniques:

In this example, we will control the brightness of an LED using Arduino PWM. First, we will increase the analogWrite() value from 0 to 255 to gradually increase the voltage of the output pin from 0v to 5v.

It will increase the brightness of the led from zero brightness to the fullest.
Then we will decrease the analogWrite() value from 255 to 0. Which will decrease the brightness of the LED from the fullest to the off state. Overall we will get a LED Dimming effect.

b) Control the brightness of LED using potentiometer

In this example, we will use a potentiometer to control the PWM value so that we can control the brightness of a LED manually

VII. Actual Circuit diagram used in laboratory with related equipment rating

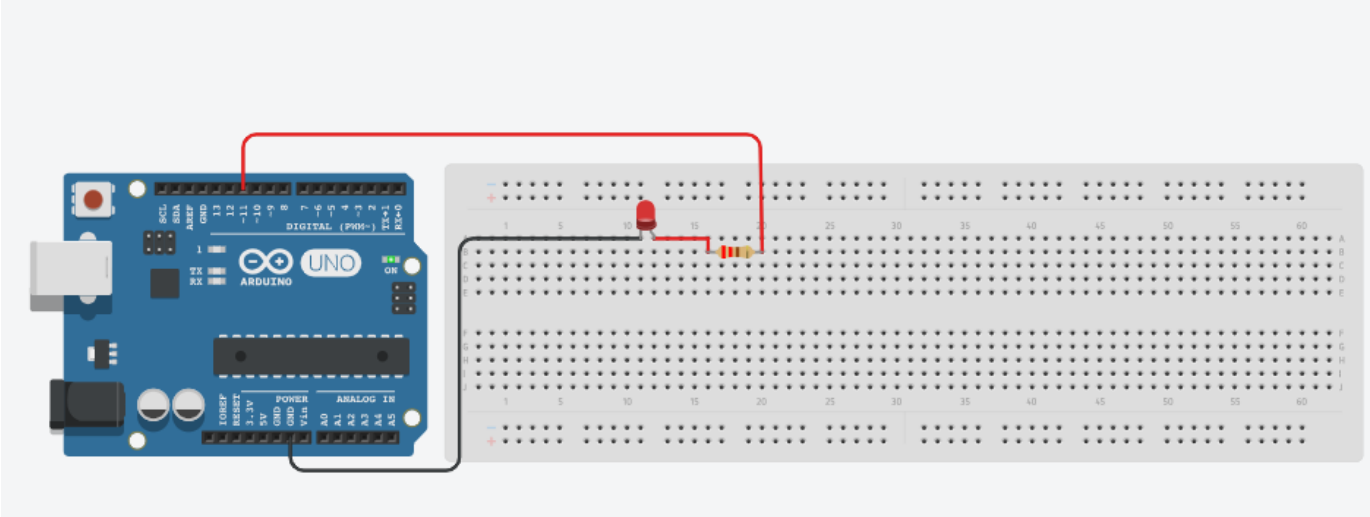


Fig 3.1 Control the brightness of LED using PWM Techniques

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	LED		01
4	Resistor		01

IX .Precautions to be followed

Use always current limiting resistor before LED connected to Arduino.

X. Procedure

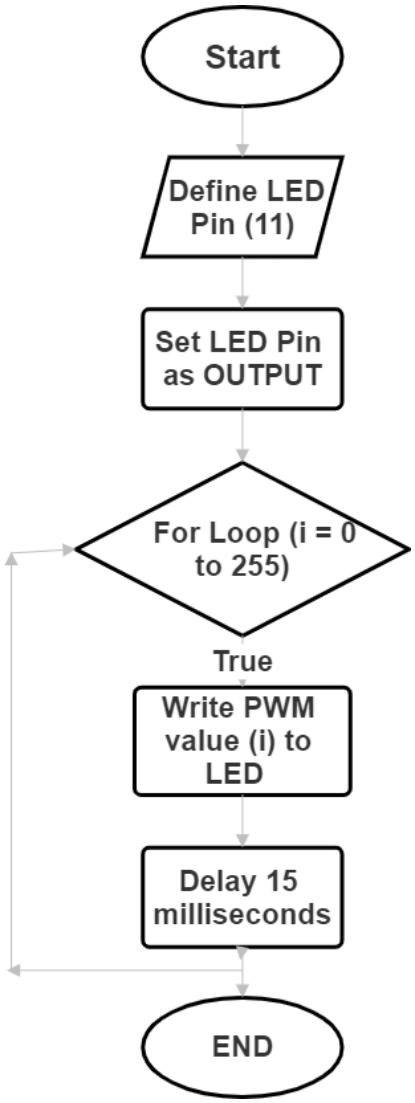
1. Interface LED to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the LED status

XI. Sample program

Step 1: Algorithm

1. Define LED pin as 11.
2. Set the LED pin as an output in setup ().
3. Enter the loop () function.
4. Use a for loop to increment PWM value from 0 to 255.
5. Write the PWM value to the LED and delay for 15 milliseconds.
6. Repeat the loop continuously

Step 2: Flowchart



Step 3: Program

```
#define led 11
 void setup()
{
 pinMode(led, OUTPUT);
}

void loop()
{
  for(int i = 0; i <= 255; i++)
 {
   analogWrite(led, i);
   delay(15);
 }
}
```

XII Results (Output of the Program)

……

……

……

……

……

XIII Conclusion

……

……

……

……

……

XIV Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Write a program to control Brightness of LED using potentiometer.

……

……

……

……

……

XV References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 4: Detect the presence or absence of Light using LDR Sensor

I. Practical Significance

To detect the presence or absence of Light using LDR Sensor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards

IV. . Laboratory Learning Outcome(s)

LLO 4.1 Interface LDR sensor with Arduino.
LLO 4.2 Write a program for detection of Light.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

LDR Sensor Module Relay is used to detect the presence of light / measuring the intensity of light. The output of the module goes high in the presence of light and it becomes low in the absence of light

VII. Actual Circuit diagram used in laboratory with related equipment rating

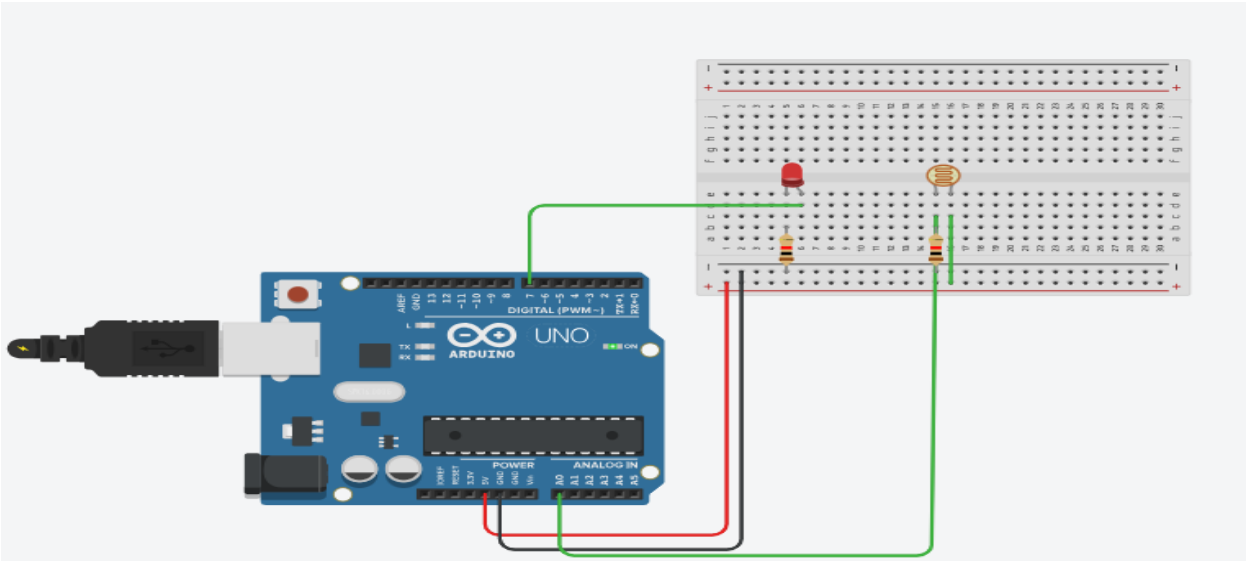


Fig 4.1 Detect presence or absence of light using LDR sensor

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	LDR sensor		01
4	RED Led		01

IX .Precautions to be followed

Use always current limiting resistor before LED connected to Arduino.

X. Procedure

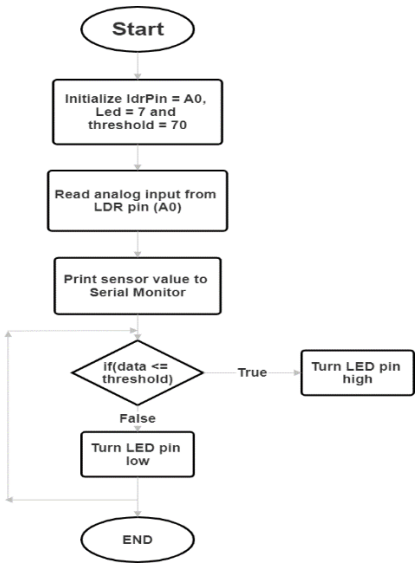
1. Interface LED and LDR to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the LED status

XI. Sample program

Step 1: Algorithm

1. Read analog input from LDR pin (connected to pin A0).
2. Print sensor value to serial monitor (baud rate: 9600).
3. Check if sensor value is less than or equal to threshold value.
4. If true, turn LED pin (connected to pin 7) high.
5. If false, turn LED pin low.
6. Repeat loop.

Step 2: Flowchart



Step 3: Program

```
int ldrPin = A0;
int led = 7;
int threshold = 70;

void setup()
{
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop()
{
  int data = analogRead(ldrPin);
  Serial.println("");
  Serial.print("Light Sensor ");
  Serial.print("Value = ");
  Serial.print(data);

  if(data <= threshold)
  {
    digitalWrite(led, HIGH);
  }
  else
  {
    digitalWrite(led, LOW);
  }
}
```

XII . Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. How LDR sensor works?

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook
3. Arshdeep Bahga, Vijay Madisetti Internet of Things: A Hands-On Approach

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 5: Measure the temperature of the object

I. Practical Significance

To Measure the temperature of the object using temperature Sensor like LM35

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 5.1 Interface Analog Temperature Sensor (e.g. LM35) with Arduino.
LLO 5.2 Write a program to sense Temperature of Object.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Temperature Sensor is a device that converts temperature into an electrical signal that can be read by the Arduino. Common types include:

- Thermistor: A type of resistor whose resistance varies significantly with temperature.
- LM35: An analog linear temperature sensor.
- DHT11/DHT22: Digital temperature and humidity sensors.
- DS18B20: A digital thermometer

The analog signal from the sensor (voltage or resistance) is read by an Analog-to-Digital Converter (ADC) in the Arduino. For example, the Arduino's ADC converts the analog input voltage (0-5V) to a digital value (0-1023).

VII. Actual Circuit diagram used in laboratory with related equipment rating

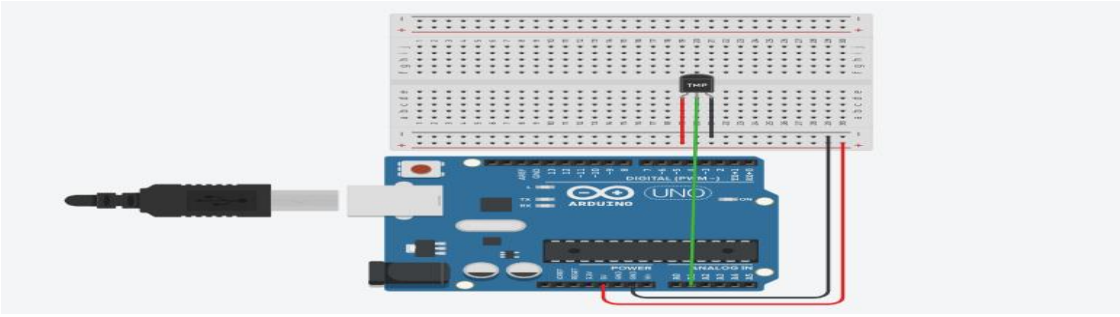


Fig 5.1Ciruit diagram to measure the temperature

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quanity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Temperature sensor		01

IX .Precautions to be followed

Connect temperature sensor in accurate manner

X .Procedure

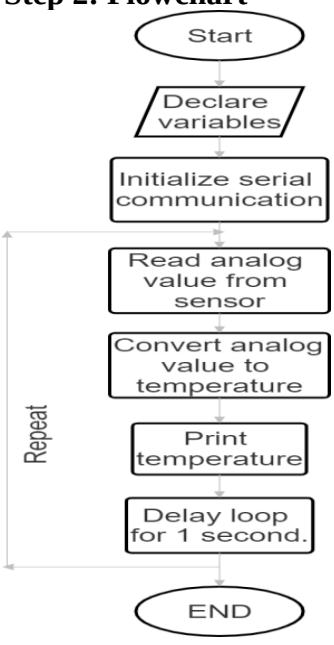
1. Interface temperature sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Declare variables: float temp, int tempPin = A1.
2. Initialize serial communication: Serial.begin(9600).
3. Read analog value from sensor: temp = analogRead(tempPin).
4. Convert analog value to temperature: temp = temp * 0.48828125.
5. Print temperature.
6. Delay loop for 1 second.
7. Repeat loop.

Step 2: Flowchart



Step 3: Program

```
float temp;
int tempPin = A1;

void setup() {
   Serial.begin(9600);
}

void loop() {
  temp = analogRead(tempPin);
  // read analog volt from sensor and save to variable temp
  temp = temp * 0.48828125;
  // convert the analog volt to its temperature equivalent
  Serial.print("TEMPERATURE = ");
  Serial.print(temp); // display temperature value
  Serial.print("*C");
  Serial.println();
  delay(1000); // update sensor reading each one second
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. List and explain different types of temperature sensors

……

……

……

……

……

2. Write a program to turn on LED when temperature is reached to some threshold value.

……

……

……

……

……

XV: References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)
3. Cornel M Amariei: Arduino Development Cookbook

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 6: Sense the touch of finger when it is placed on board

I. Practical Significance

To sense the touch of finger when it is placed on board.

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards..

IV. Laboratory Learning Outcome(s)

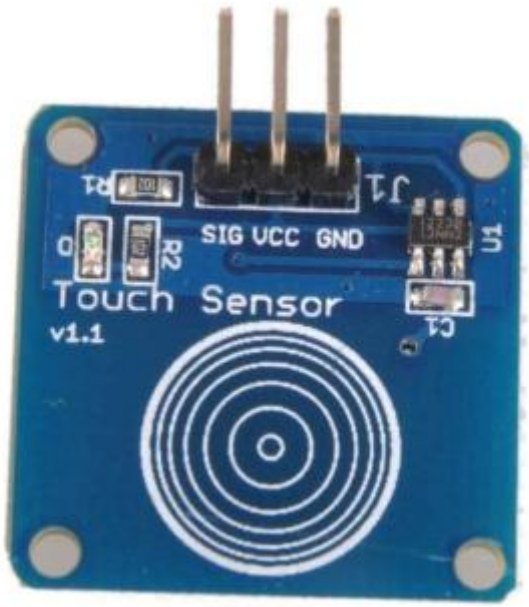
LLO 6.1 Interface touch Sensor with Arduino.
LLO 6.2 Write program to sense the touch when finger is placed on board

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI.Relevant Theoretical Background

A touch sensor is a type of device that captures and records physical touch or embrace on a device and/or object. A touch sensor primarily works when an object or individual gets in physical contact with it. Touch sensors are also called as tactile sensors and are sensitive to touch, force or pressure. It can be implemented using Capacitive or Resistive sensing technology.



VII. Actual Circuit diagram used in laboratory with related equipment rating

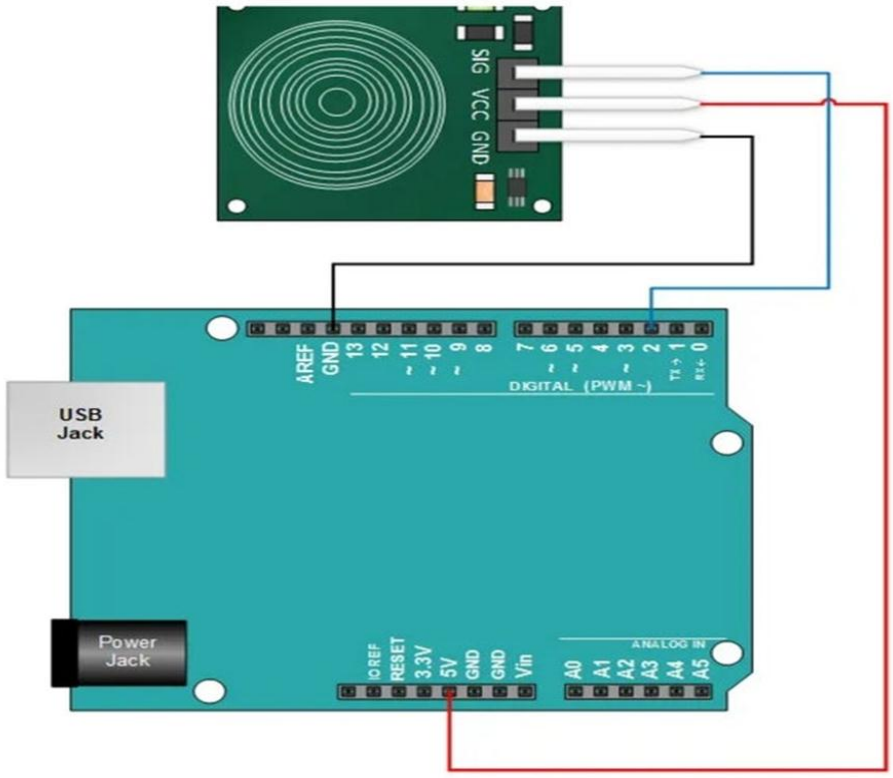


Fig.6.1 interfacing of touch sensor

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Touch sensor		01

IX. Precautions to be followed

Connect Touch sensor in accurate manner.

X. Procedure

1. Interface touch sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.

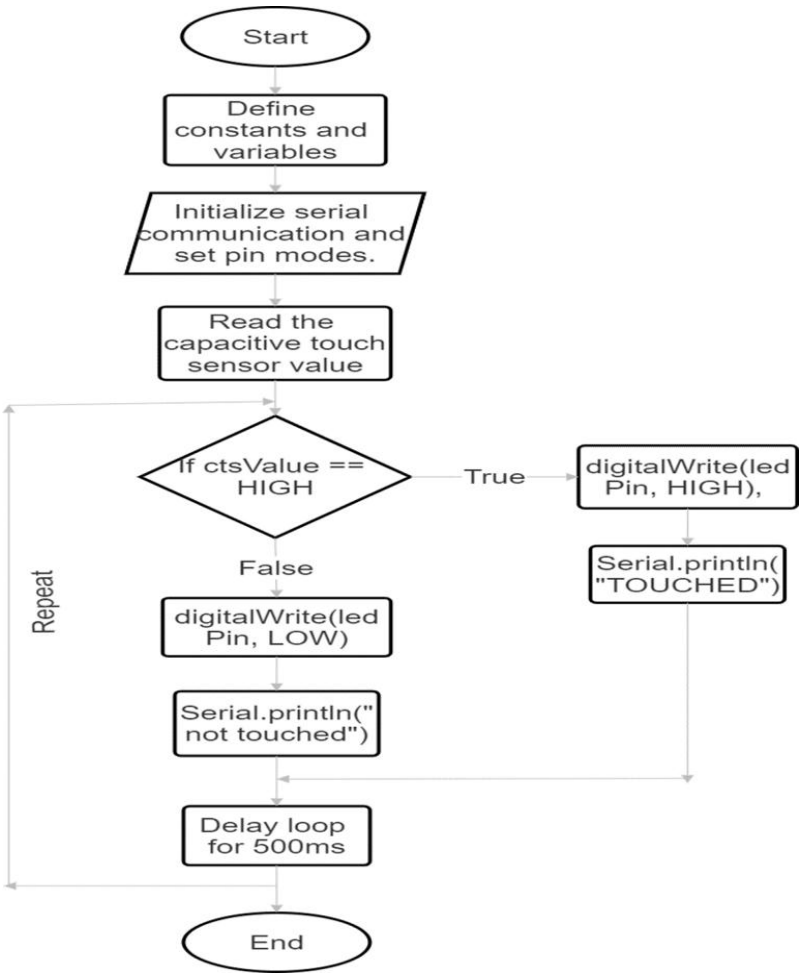
6. Upload the program
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Define constants and variables: define ctsPin 2, int ledPin = 13.
2. Initialize serial communication and set pin modes.
3. Read the capacitive touch sensor value: int ctsValue =digitalRead(ctsPin).
4. If ctsValue == HIGH, then digitalWrite(ledPin, HIGH),
5. Serial.println("TOUCHED").
6. Else, digitalWrite(ledPin, LOW), Serial.println("not touched").
7. Delay loop for 500ms.

Step 2: Flowchart



Step 3: Program

```
#define ctsPin 2
int ledPin = 13;

void setup() {
```

```
        Serial.begin(9600);
        pinMode(ledPin, OUTPUT);
        pinMode(ctsPin, INPUT);
}
void loop() {
      int ctsValue = digitalRead(ctsPin);
      if (ctsValue == HIGH) {
            digitalWrite(ledPin, HIGH);
            Serial.println("TOUCHED");
}
else{
digitalWrite(ledPin,LOW);
Serial.println("not touched");
}
delay(500);
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. **Conclusion**

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. How to interface buzzer and touch sensor?

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 7: Detect the obstacle using IR sensor

I. Practical Significance

To detect the obstacle using IR sensor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards..

IV. . Laboratory Learning Outcome(s)

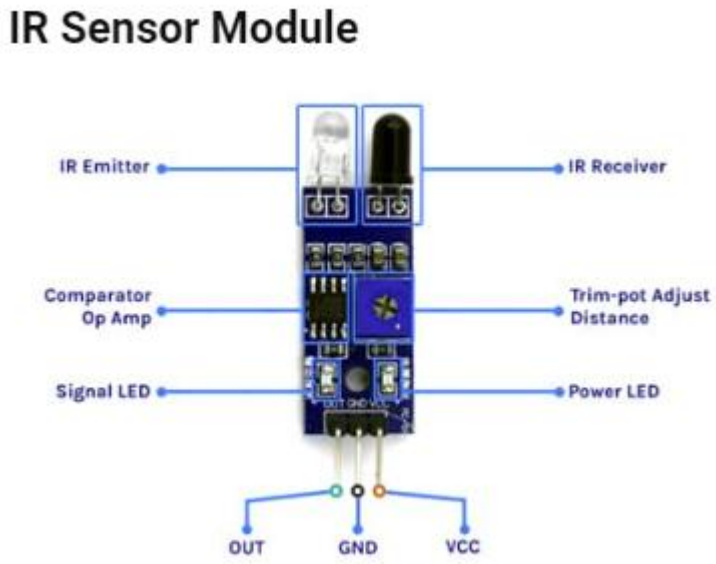
LLO 7.1 Interface IR Sensor with Arduino/ Raspberry Pi.
LLO 7.2 Write program to detect the obstacle.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI.Relevant Theoretical Background

IR sensors work by detecting infrared radiation emitted by objects in the environment. They typically use a photodiode or a phototransistor to detect the IR radiation and convert it into an electrical signal that can be processed and analyzed.



The IR transmitter continuously emits the IR light and the IR receiver keeps on checking for the reflected light. If the light gets reflected back by hitting any object in front it, the IR receiver receives this light. This way the object is detected in the case of the IR sensor.

VII. Actual Circuit diagram used in laboratory with related equipment rating

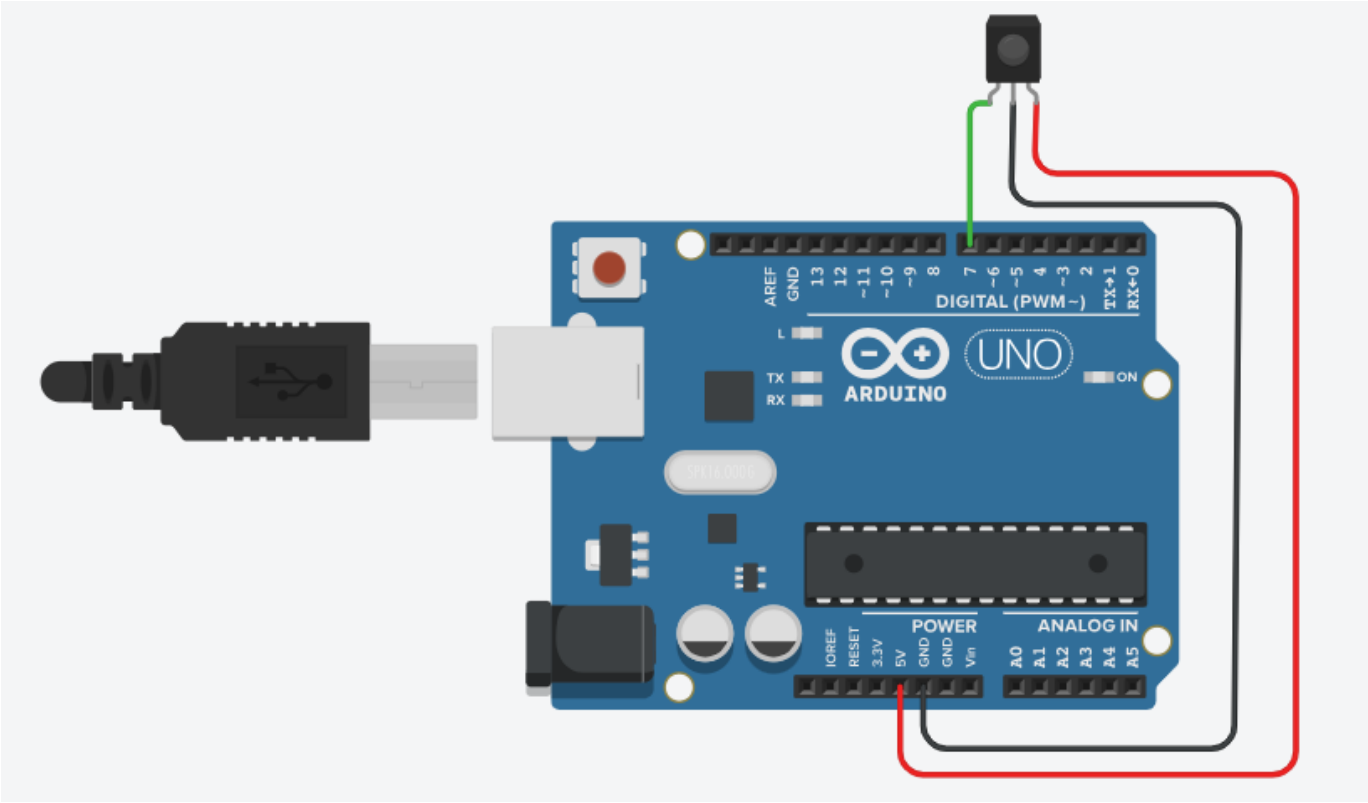


Fig 7.1 Detect the obstacle using IR sensor

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	IR sensor		01

IX .Precautions to be followed

Connect pins of IR sensor in accurate manner.

X. Procedure

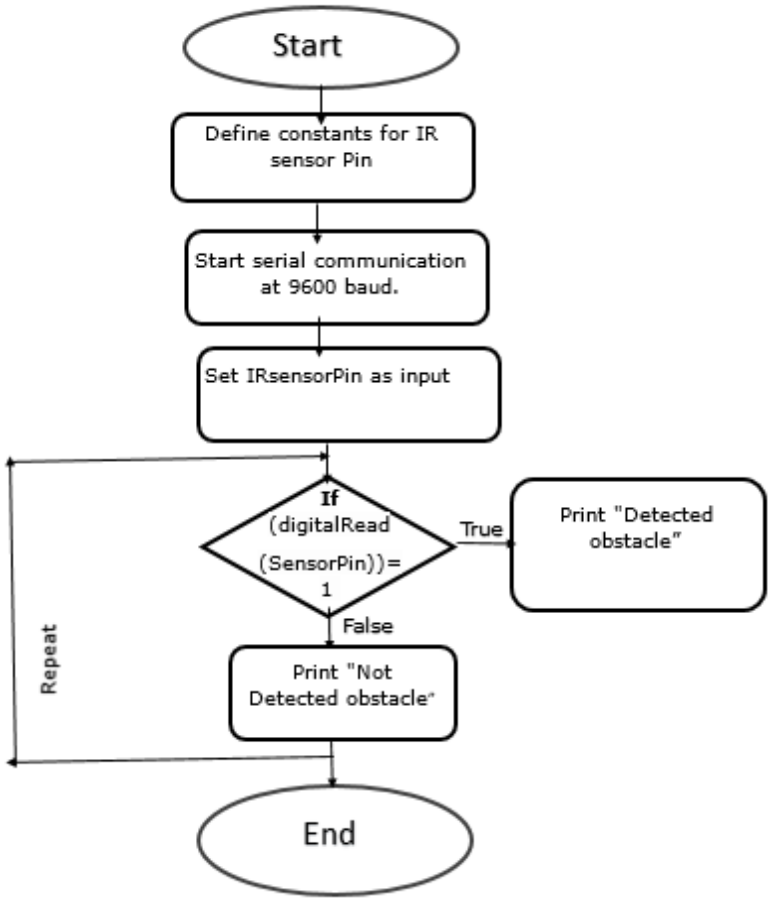
1. Interface IR sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Start
2. Initialize pin 7 as IRSensor
3. Set IRSensor pin mode to INPUT
4. Start Serial communication at 9600 baud rate
5. Read the status of the IRSensor
6. If the status is 1
    a. Print "obstacle detected" to the Serial Monitor
7. Else
    a. Print "No obstacle detected" to the Serial Monitor

Step 2: Flowchart



Step 3: Program

```
int IRSensor = 7; // connect ir sensor to arduino pin 7
void setup()
{
  pinMode (IRSensor, INPUT); // sensor pin INPUT
  Serial.begin(9600);
```

```
}

void loop()
{
  int statusSensor = digitalRead (IRSensor);

  if (statusSensor == 1)
{
    Serial.println("obstacle detected");
  }
else
  {
    Serial.println(" No obstacle detected");
  }
  }
```

XII Results (Output of the Program)

……

……

……

……

……

XIII Conclusion

……

……

……

……

……

XIV Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain different types of IR sensors

……

……

……

……

……

XV References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 8: * Measure the Distance between sensor and object using ultrasonic sensor

I. Practical Significance

To Measure the Distance between sensor and object using ultrasonic sensor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards

IV. . Laboratory Learning Outcome(s)

LLO 8.1 Interface Ultrasonic Sensor with Arduino/Raspberry Pi.
LLO 8.2 Write a program to measure the Distance between sensor and object.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

An ultrasonic Sensor is a device used to measure the distance between the sensor and an object without physical contact. This device works based on time-to-distance conversion.

Working Principle of Ultrasonic Sensor: Ultrasonic sensors measure distance by sending and receiving the ultrasonic wave. The ultrasonic sensor has a sender to emit the ultrasonic waves and a receiver to receive the ultrasonic waves. The transmitted ultrasonic wave travels through the air and is reflected by hitting the Object. Arduino calculates the time taken by the ultrasonic pulse wave to reach the receiver from the sender.

We know that the speed of sound in air is nearly 344 m/s,
So, the known parameters are time and speed (constant). Using these parameters, we can calculate the distance traveled by the sound wave.
Formula: Distance = Speed * Time
In the code, the "time" variable stores the time taken by the sound wave traveling from the emitter to the receiver. That is double the time to reach the object, whereas the sensor returns the total time including sender to object and object to receiver. Then, the time taken to reach the object is half of the time taken to reach the receiver.
so we can write the expression as,
Distance = Speed of Sound in Air * (Time Taken / 2)
Note: Speed of sound in air = 344 m/s.

VII. Actual Circuit diagram used in laboratory with related equipment rating

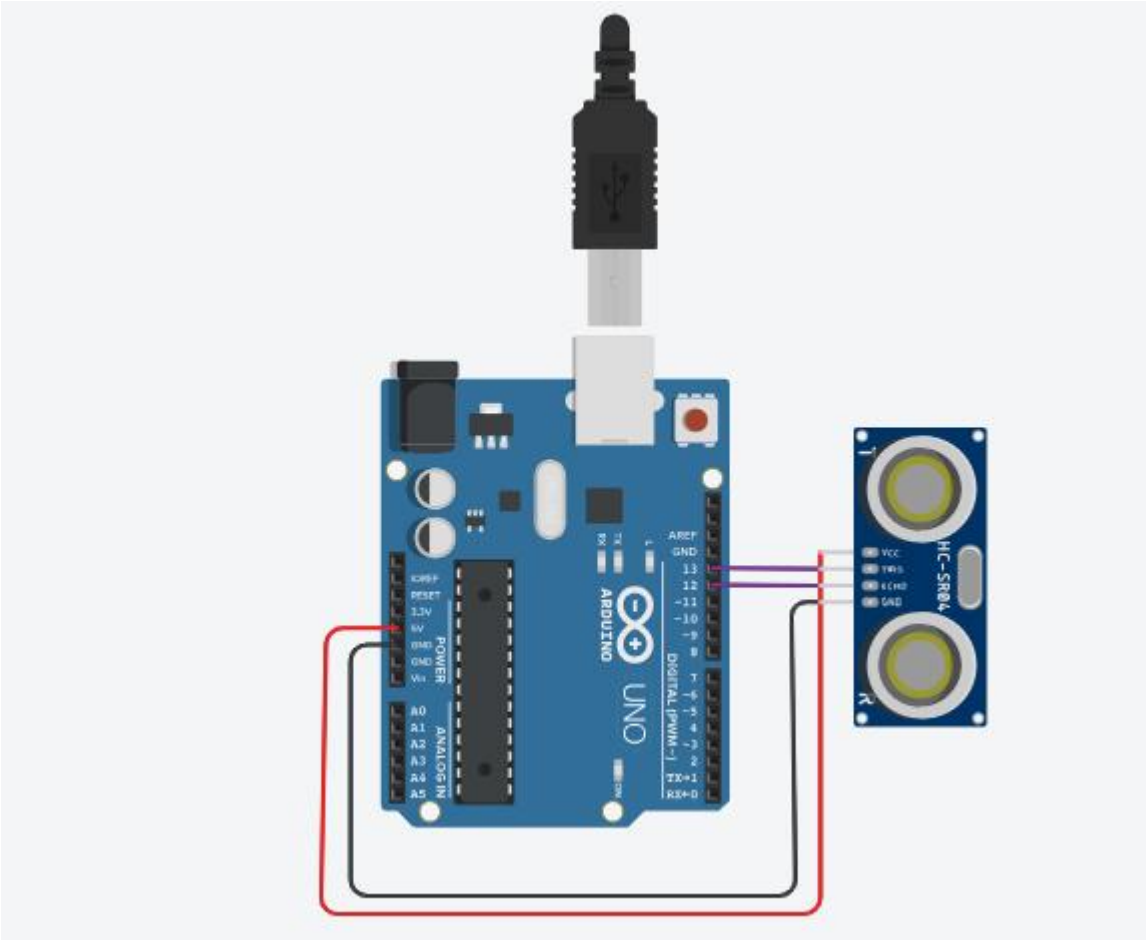


Fig 8.1 Measure distance between ultrasonic sensors

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Ultrasonic sensor		01

IX .Precautions to be followed

Connect pins of ultrasonic sensor in accurate manner.

X .Procedure

1. Interface ultrasonic sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.

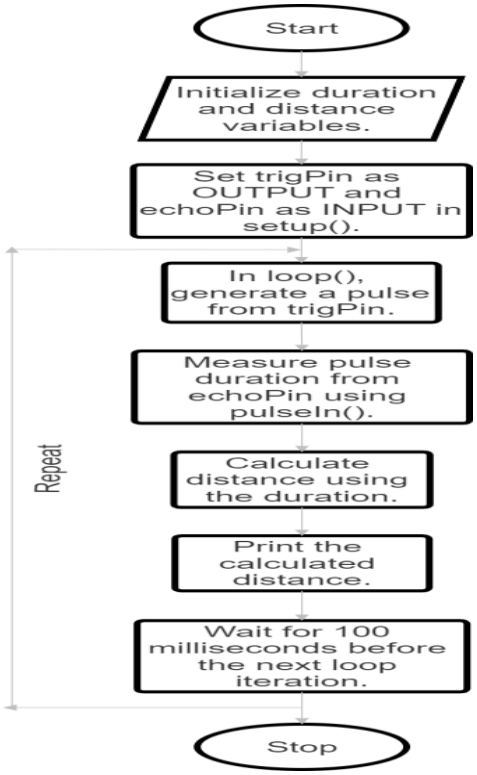
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Define echoPin to 12 and trigPin to 13.
2. Initialize time and distance variables.
3. Set trigPin as OUTPUT and echoPin as INPUT in setup().
4. In loop(), generate a pulse from trigPin.
5. Measure pulse duration from echoPin using pulseIn().
6. Calculate distance using the duration.
7. Print the calculated distance.
8. Wait for 100 milliseconds before the next loop iteration.

Step2: Flowchart



Step3: program

```
#define echoPin 12
#define trigPin  13

long duration;
int distance;

void setup()
{
```

```
   pinMode(trigPin, OUTPUT);
   pinMode(echoPin, INPUT);
   Serial.begin(9600);
   Serial.println("Distance measurement using Arduino Uno.");
   delay(500);
}

void loop()
{
   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);
   duration = pulseIn(echoPin, HIGH);
   distance = duration * 0.0344 / 2;
   Serial.print("Distance: ");
   Serial.print(distance);
   Serial.println(" cm");
   delay(100);
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain working principle of ultrasonic sensor

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 9: Detect the presence of Gas

I. Practical Significance

To detect the presence of gas

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 9.1 Interface Gas Sensor with Arduino/ Raspberry Pi.
LLO 9.2 Write program to detect the presence of Gas.
.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

A **gas sensor** is a device which detects the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. Based on this voltage value the type and concentration of the gas can be estimated. The type of gas the sensor can detect depends on the **sensing material** present inside the sensor. Normally these sensors are available as modules When the concentration of the gas exceeds this threshold the digital pin goes high. The analog pin can be used to measure the concentration of the gas.

For Module		
1	Vcc	This pin powers the module, typically the operating voltage is +5V
2	Ground	Used to connect the module to system ground
3	Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas

VII. Actual Circuit diagram used in laboratory with related equipment rating

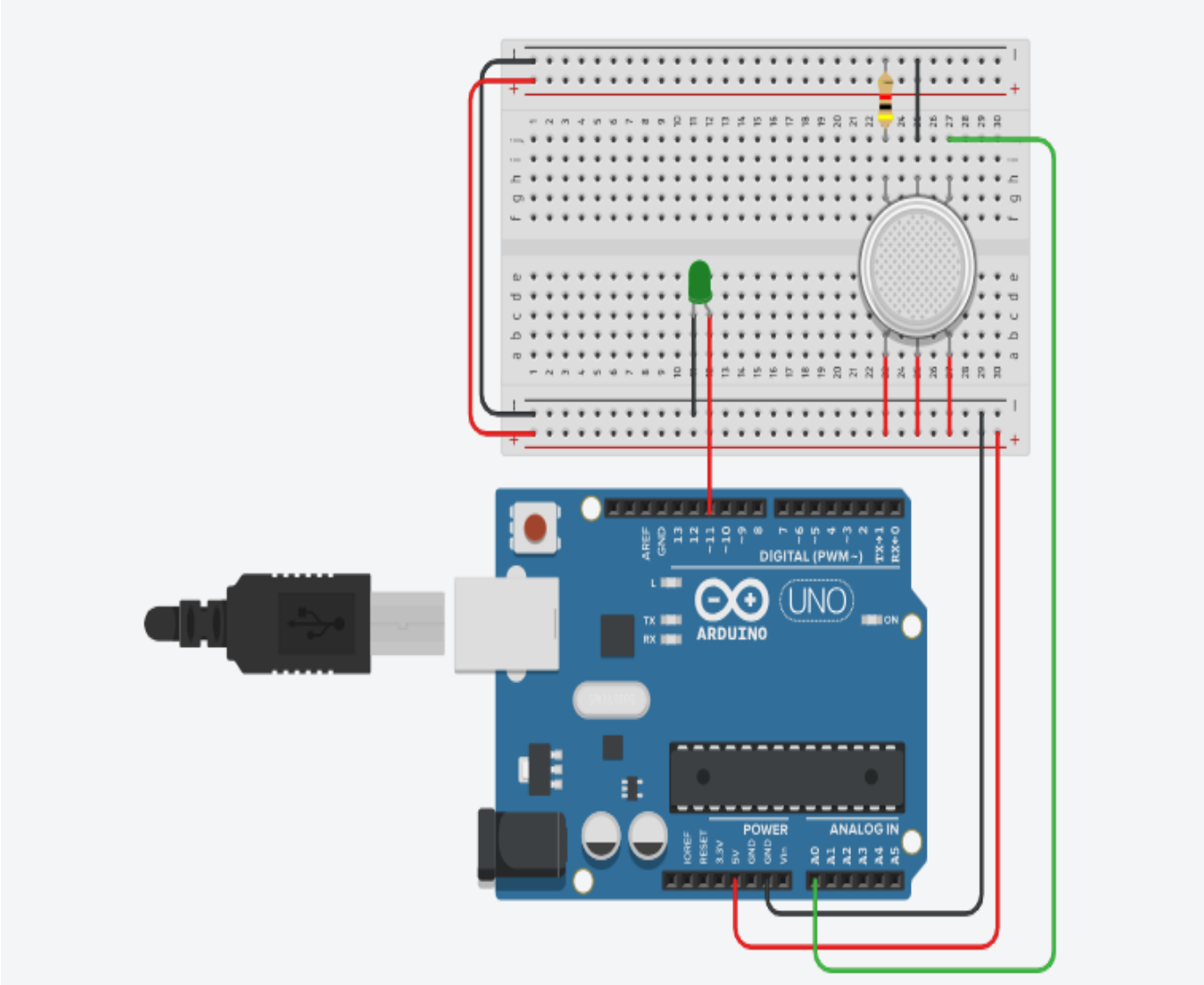


Fig 9.1Ciruit diagram to detect gas

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quntity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Gas sensor		01

IX .Precautions to be followed

Connect gas sensor in accurate manner

X .Procedure

1. Interface gas sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Set the LED pin (11) as an output and the MQ2 gas sensor pin (A0) as an analog input.
2. Read the analog value from the MQ2 gas sensor pin
3. Checks if the sensor value is greater than or equal to 470, which indicates the presence of smoke.
4. If smoke is detected, it turns on the LED and prints "SMOKE DETECTED" to the serial console.
5. If no smoke is detected, it turns off the LED and prints the current sensor value to the serial console.

Step 2: Flowchart

Step 3: Program

```
   int LED = 11;
const int gas = 0;
int MQ2pin = A0;


void setup() {
  Serial.begin(9600);
}

void loop() {
  float sensorValue,MQ2pin;
sensorValue = analogRead(MQ2pin); // read analog input pin 0

  if(sensorValue >= 470){
    digitalWrite(LED,HIGH);
    Serial.println(" |SMOKE DETECTED");


  }
  else{
          digitalWrite(LED,LOW);
    Serial.println("Sensor Value: ");
    Serial.println(sensorValue);
  }
  delay(1000);
}
          float getsensorValue(int pin){
          return (analogRead(pin));
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. List and explain different types of gas sensors

……

……

……

……

……

2. Write a program to turn on buzzer when gas is detected.

……

……

……

……

……

XV. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)
3. Cornel M Amariei: Arduino Development Cookbook

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 10: Detect the vibration of an object using vibration detector sensor SW-420 with Arduino

I. Practical Significance

To detect the vibration of an object using vibration detector sensor SW-420 with Arduino

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards..

IV. Laboratory Learning Outcome(s)

LLO 10.1 Interface vibration detector sensor with Arduino.
LLO 10.2 Write a program to interface vibration detector.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

A vibration sensor is a device that detects mechanical vibrations. It measures the vibration levels in your machine and alerts you to any potential problems, like equipment failure or worn parts that need replacement.

Vibration Sensor (SW-420) is a high sensitivity non-directional vibration sensor. When the module is stable, the circuit is turned on and the output is high. When the movement or vibration occurs, the circuit will be briefly disconnected and output low

VII. Actual Circuit diagram used in laboratory with related equipment rating

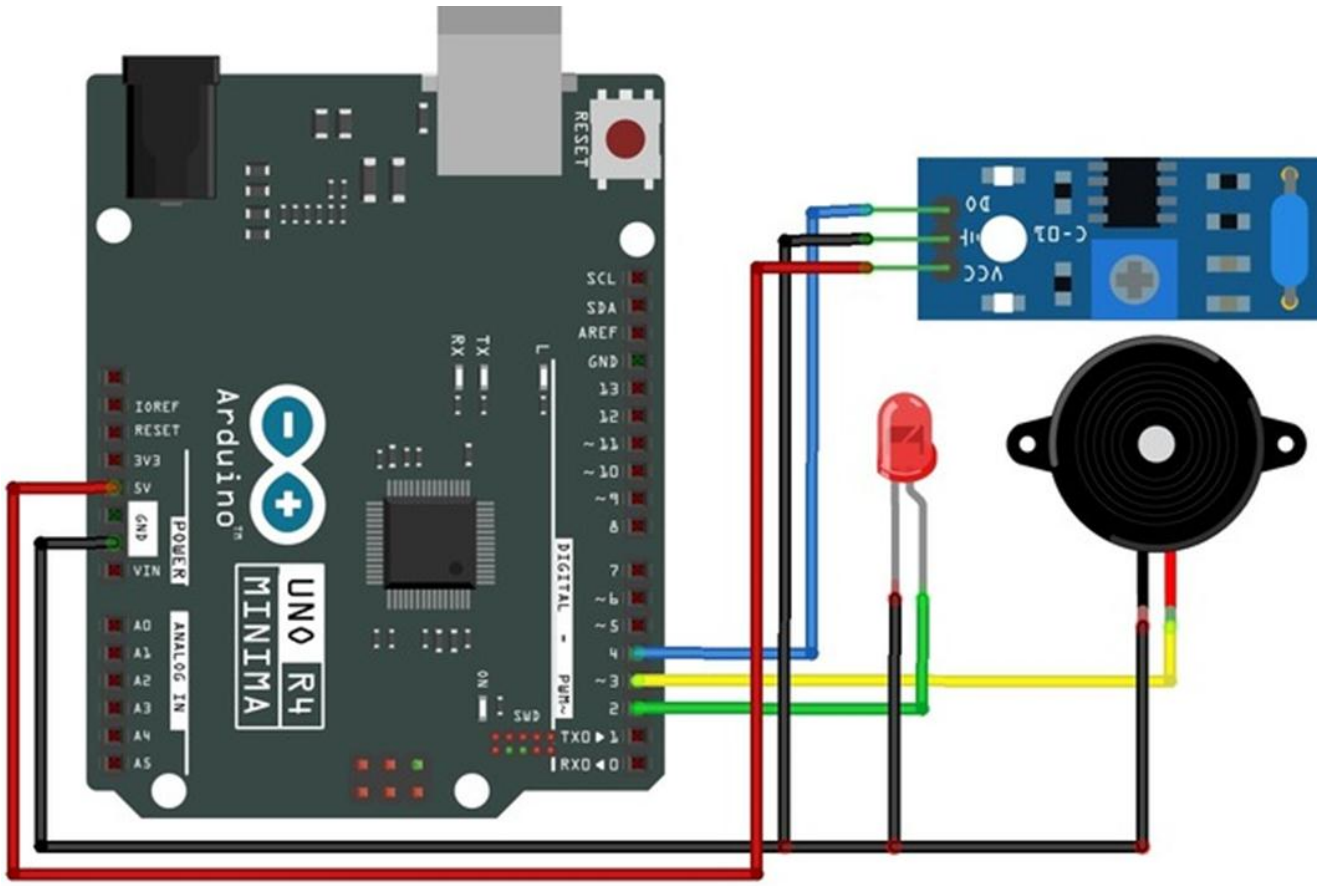


Fig.10.1 interfacing of vibration sensor

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Vibration detection sensor		01
4	Led		01
4	Buzzer		01

IX. Precautions to be followed

Connect Vibration detection sensor sensor in accurate manner.

X. Procedure

1. Interface vibration detection sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.

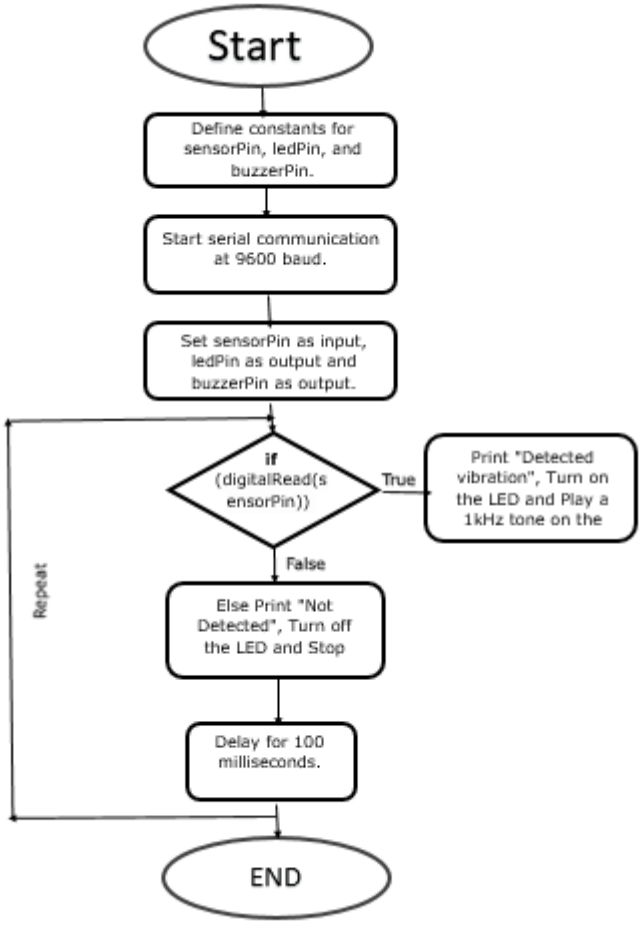
6. Upload the program
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Define constants for sensorPin, ledPin, and buzzerPin.
2. Start serial communication at 9600 baud.
3. Set sensorPin as input, ledPin as output and buzzerPin as output.
4. If vibration detected on sensorPin is True.
5. Print "Detected vibration...",Turn on the LED and Play a 1kHz tone on the buzzer.
6. Else Print "Not detected", Turn off the LED and Stop the tone on the buzzer.
7. Delay for 100 milliseconds.

Step 2: Flowchart



Step 3: Program

```
const int sensorPin = 4;
const int ledPin = 2;
const int buzzerPin = 3;

void setup()
```

```
{
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop()
{
  if (digitalRead(sensorPin))
  {
    Serial.println("Detected vibration...");
    digitalWrite(ledPin, HIGH);
    tone(buzzerPin, 1000);
  }
  else
  {
    Serial.println("...");
    digitalWrite(ledPin, LOW);
    noTone(buzzerPin);
  }

  delay(100);
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Write application of vibration sensor

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 11: Change the status of Buzzer ON/OFF

I. Practical Significance

To change the status of buzzer when motion is detected

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO1 - Integrate hardware and software for simple IoT applications.

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. . Laboratory Learning Outcome(s)

LLO 11.1 Interface buzzer.

LLO 11.2 Write a program to display motion detected or not.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

c. Follow ethical practices

VI. Relevant Theoretical Background

An electric buzzer is an audio signaling device. It produces sound within 1 to 7 kHz frequency range. Buzzers are simple in use, light in weight, and available in affordable price.

VII. Actual Circuit diagram used in laboratory with related equipment rating

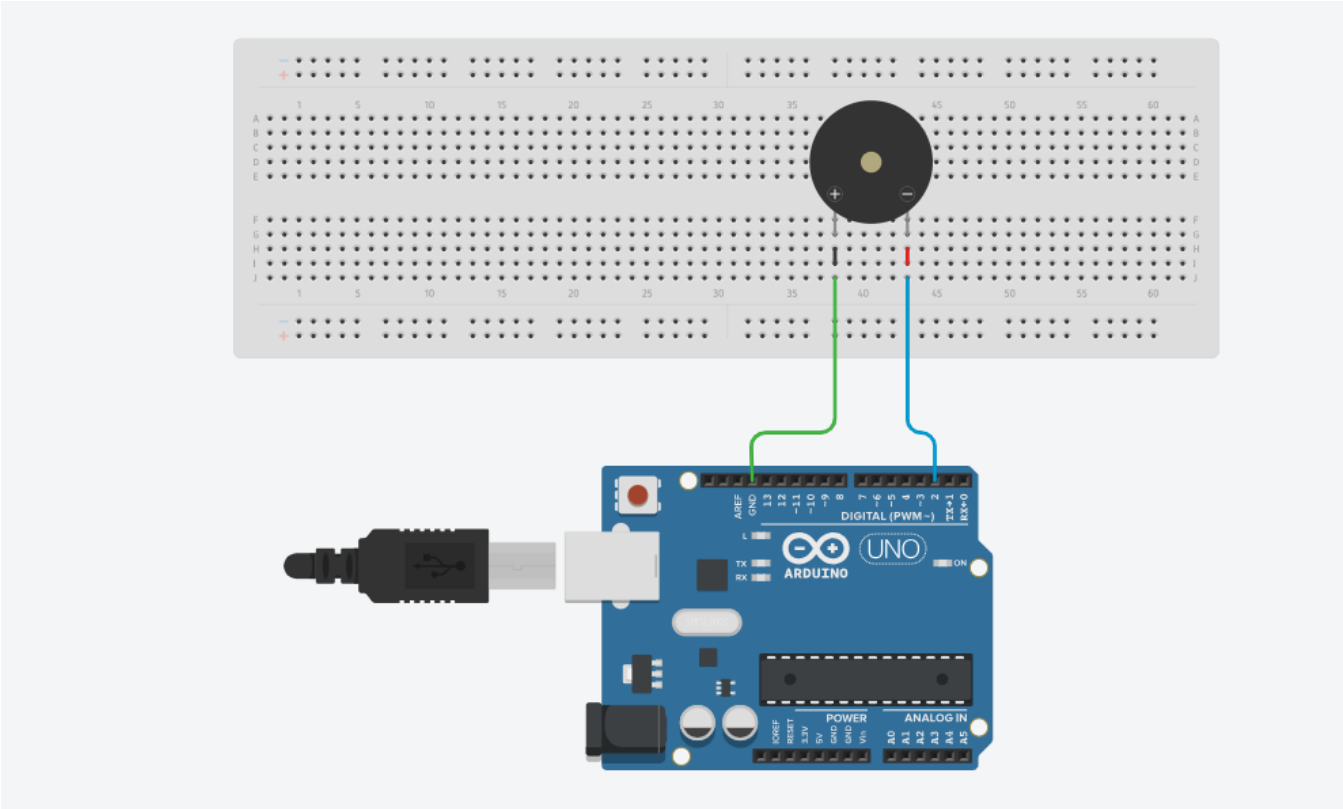


Fig 11.1 Interfacing of buzzer

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Buzzer		01

IX .Precautions to be followed

Connect pins of buzzer in accurate manner.

X. Procedure

1. Interface buzzer to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program

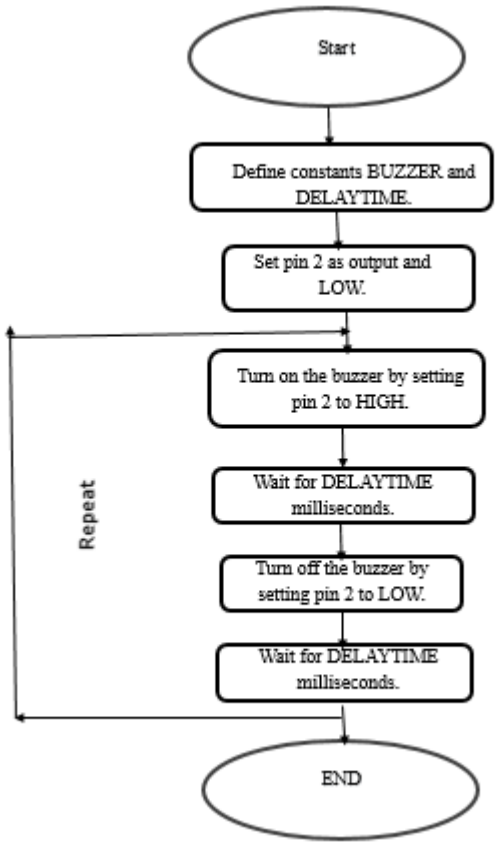
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Define constants BUZZER and DELAYTIME.
2. Set pin 2 as output and LOW.
3. Turn on the buzzer by setting pin 2 to HIGH.
4. Wait for DELAYTIME milliseconds.
5. Turn off the buzzer by setting pin 2 to LOW.
6. Wait for DELAYTIME milliseconds.

Step 2: Flowchart



Step 3: Program

```
#define BUZZER 2
#define DELAYTIME 1000

void setup() {
  pinMode(2, OUTPUT);
  digitalWrite(2, LOW);
}

void loop(){
  digitalWrite(2, HIGH);
  delay(DELAYTIME);
  digitalWrite(2, LOW);
```

```
  delay(DELAYTIME);
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII . Conclusion

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain different types of buzzer and their applications

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1	Coding and Debugging ability	
2	Making connections of hardware	
3	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4	Relevance of output of the problem definition	
5	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 12: Display Humidity and Temperature on LCD using DHT11 sensor

I. Practical Significance

To display Humidity and Temperature on LCD using DHT11 sensor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards
CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. . Laboratory Learning Outcome(s)

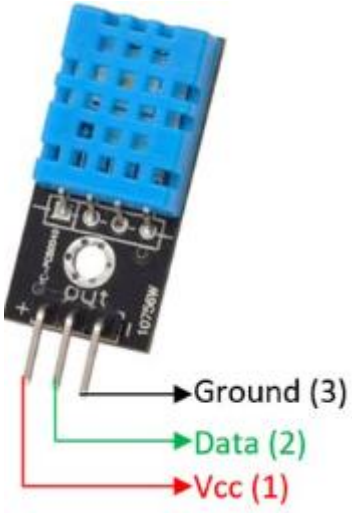
LLO 12.1 Interface DHT11 sensor and I2C LCD with Arduino.
LLO 12.2 Write a program to display Humidity and Temperature on LCD.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

The **DHT11** is a commonly used **Temperature and humidity sensor that** comes with a dedicated NTC to measure temperature and Humidity



VII. Actual Circuit diagram used in laboratory with related equipment rating

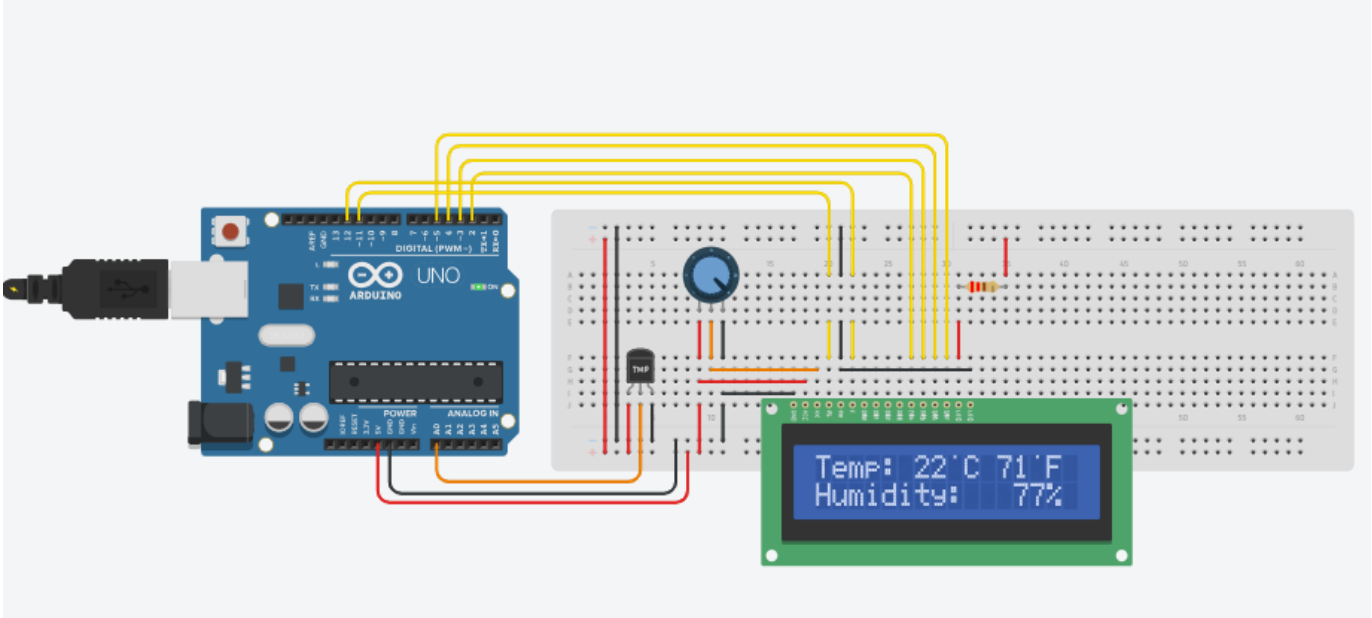


Fig 12.1 Interfacing of temperature sensor and LCD

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quanity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Temperature and Humidity sensor		01

IX .Precautions to be followed

Do accurate connections.

X. Procedure

1. Interface Humidity and Temperature sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step3: program

```
#include<LiquidCrystal.h>
#include<SoftwareSerial.h>

LiquidCrystal lcd(11,12,2,3,4,5);
```

```
int degreeCelsius = 0;
int degreeFahrenheit = 0;
float percentHumidity = 0;
float tempCalib=5.3;
float humidCalib=2.8654;

int convertCtoF(float c) { return c * 1.8 + 32; }

void clear(void){
lcd.setCursor(0,0);
lcd.print("               ");
lcd.setCursor(0,1);
lcd.print("               ");
delay(500);
}

void printResultsLCD(void){
degreeFahrenheit = convertCtoF(degreeCelsius);
lcd.setCursor(0,0);
lcd.print("Temp:  ");
lcd.setCursor(6,0);
lcd.print(degreeCelsius);
lcd.print("\260C ");
lcd.setCursor(11,0);
lcd.print(degreeFahrenheit);
lcd.print("\260F ");
lcd.setCursor(0,1);
lcd.print("Humidity:   ");
lcd.print((int)percentHumidity);
lcd.setCursor(14,1);
lcd.print("\045 ");  // octal for percent - ascii
delay(1500);
}

// Global Variables
int DHT11_Pin = A0; // DHT11 Data Pin

int Humidity = 0;
int Temp = 0;
int TempComma = 0;
bool P_DHTError = false; // Checksum Error

// a Delay routine. Call DelayTimer(time in uSec)

void DelayTimer(long int DelayValue){
long int DelayTime = micros();
do {

}while (micros()-DelayTime < DelayValue);
}
```

```
// Main DHT Void

void DHT11(){

long int DataTime = 0;

byte Result[45];
byte DataArray = 0;
byte DataCounter = 0;
byte DHTData[4];

bool BlockDHT=false;

// Trigger Sensor - Handshake between
// MCU and DHT  (described in the Datasheet)

pinMode(DHT11_Pin,OUTPUT);
digitalWrite(DHT11_Pin,HIGH);
DelayTimer(250000); //Wait 250millisec
digitalWrite(DHT11_Pin,LOW);
DelayTimer(30000); //Wait 30millisec
digitalWrite(DHT11_Pin,HIGH);
DelayTimer(50); //Wait 50microsec
pinMode(DHT11_Pin,INPUT);
// read the Bits and put them into a Result array (It will count 42 bits. The first two one are
unused)

do {
if (digitalRead(DHT11_Pin) == 0 && BlockDHT == false) {BlockDHT =
true;Result[DataArray]=(micros()-DataTime);DataArray++;DataTime=micros();} //If DHT
pin is low, go to next Dataset
if (digitalRead(DHT11_Pin) == 1) {BlockDHT = false;} // As long as DHT pin is Hight add
time in Microseconds to Result

}while((micros()-DataTime) < 150); // if DTH Sensor high for more than 150 usec, leave loop

// Asign 1 or 0 to Result variable. If more than 80uS Data as "1"
// Starting at Data set 02. First two Datasets are ignored!

for (int i=2; i< DataArray; i++) {
if (Result[i] <= 90) Result[i]=0; else Result[i]=1;
}

for (int j=0; j< 5; j++){ // redo it for the 5 Bytes (40 Databits /8 = 5)
for (int i=0; i< 8; i++) {bitWrite(DHTData[j], 7-i, Result[i+2+(j*8)]);} // Create 5 Databytes
from the 40 Databits (Ignoring the 2 first Databits)

}
// check checksum }
Temp = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```

```
degreeCelsius=Temp;
percentHumidity = DHTData[0]/humidCalib;
Temp = DHTData[2]/tempCalib;
degreeCelsius=Temp;
if (DHTData[4] == (DHTData[0]+DHTData[1]+DHTData[2]+DHTData[3])){Humidity =
DHTData[0];Temp = DHTData[2];TempComma = DHTData[3];P_DHTError=false;} else
P_DHTError=true; //If Checksum is worng, Temp=99 (Dataset 0-3 in addition = Dataset 4 =
Checksum OK)

}

void setup() {
Serial.begin(9600);
lcd.begin(16,2);

lcd.setCursor(0,0);
lcd.print("Temp \046 Humidity");
lcd.setCursor(0,1);
lcd.print("    Display    ");
delay(1000);
lcd.clear();
}

void loop() {
DHT11();
printResultsLCD();
DelayTimer(500000); //wait 2,5 sec
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain working principle of temperature and Humidity sensor

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 13: Display the message as per detection of motion of object

I. Practical Significance

To Display the message as per detection of motion of object

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards.

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 13.1 Interface PIR Sensor with Arduino/ Raspberry Pi to Detect Motion of object.
LLO 13.2 Write a program to display motion detected or not.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

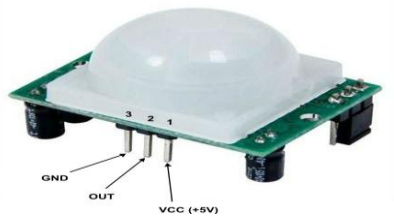
VI. Relevant Theoretical Background

.

PIR sensor: PIR sensors allow you to sense motion. They are small, inexpensive, low-power, easy to use

A passive infrared sensor is an electronic sensor that measures infrared light radiating from objects. PIR sensors mostly used in PIR-based motion detectors

The below image shows a typical pin configuration of the PIR sensor, which is quite simple to understand the pinouts



- Pin1 corresponds to the drain terminal of the device, which connected to the positive supply 5V DC.
- Pin2 corresponds to the source terminal of the device, which connects to the ground terminal via a 100K or 47K resistor. The Pin2 is the output pin of the sensor. The pin 2 of the sensor carries the detected IR signal to an amplifier.
- Pin3 of the sensor connected to the ground.

PIR sensor i.e. Passive Infrared Sensor, passive word indicates PIR Sensor does not generate or radiate any energy for detection purposes.

PIR Sensors don't detect or measure "**HEAT**"; they detect the infrared radiation emitted or reflected from objects.

They are small, inexpensive, low power and easy to use. They are commonly found at home, medical, factories etc. areas

VII. Actual Circuit diagram used in laboratory with related equipment rating

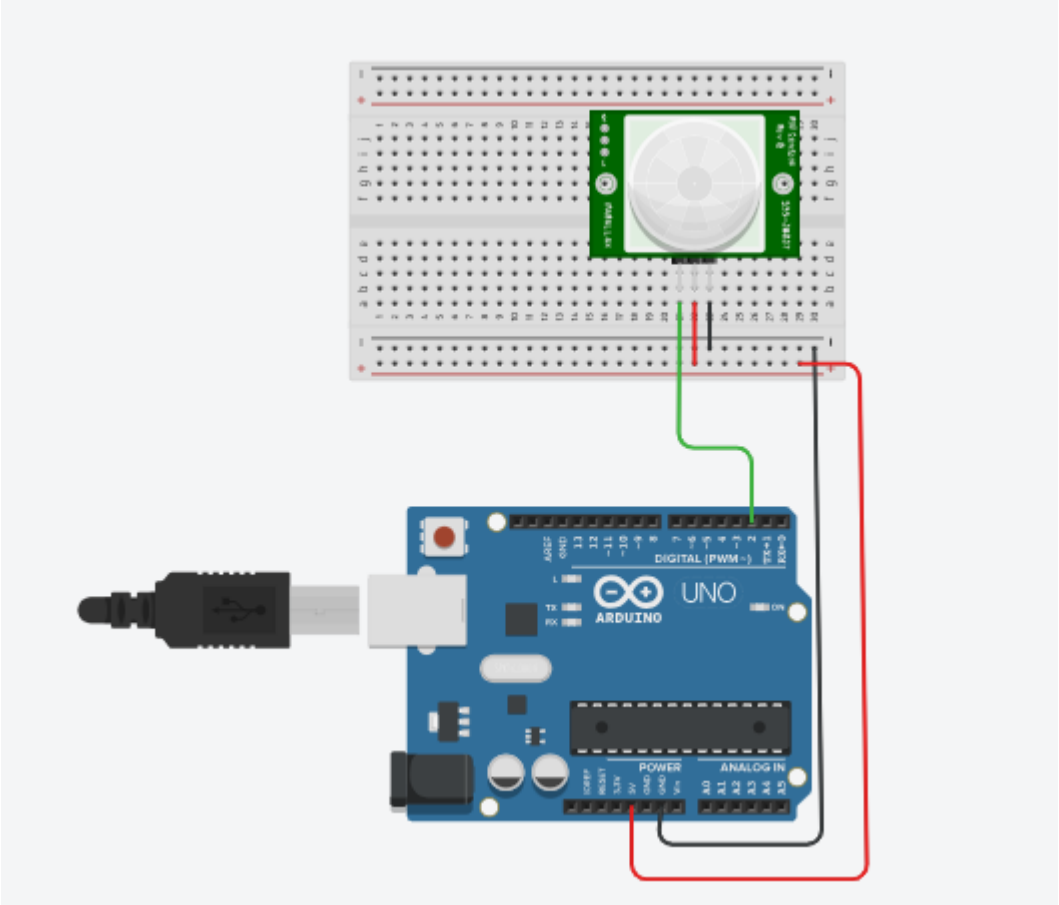


Fig 13.1Ciruit diagram to detect motion with PIR sensor

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quanity
1	Arduino IDE software		01
2	Arduino Uno		01
3	PIR sensor		01

IX .Precautions to be followed

Connect PIR sensor in accurate manner

X .Procedure

1. Interface PIR sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.

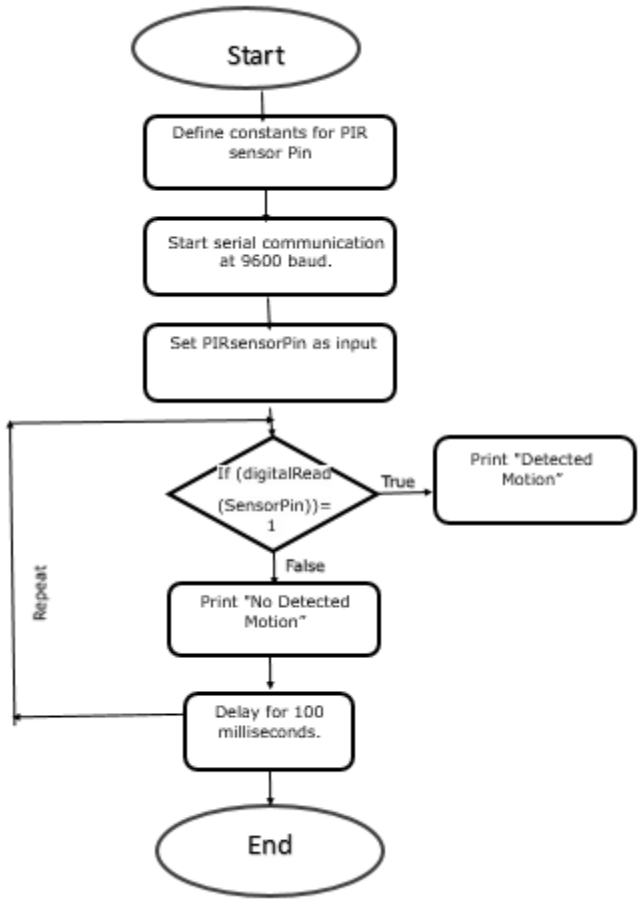
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Define PIR sensor pin
2. In the setup function, set PIRpin as INPUT.
3. In the loop function, read the state of PIRpin and store it in PIRread.
4. If PIRread is 0, display'Motion Detected ' on Serial monitor
5. Repeat loop.

Step 2: Flowchart



Step 3: Program

```
int sensorState = 0;

void setup()
{
  pinMode(2, INPUT);

  Serial.begin(9600);
}

void loop()
{
```

```
  // read the state of the sensor/digital input
  sensorState = digitalRead(2);
  // check if sensor pin is HIGH. if it is, set the
  // display message on Serial monitor.
  if (sensorState == HIGH)
 {

    Serial.println("motion detected");
  } else
{
       Serial.println("No motion detected");

  }
  delay(10); // Delay a little bit to improve simulation performance
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Perform the same experiment with interfacing LCD display

……

……

……

……

……

XV.References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)
3. https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview
4. Cornel M Amariei: Arduino Development Cookbook

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total25	
	Dated Signature of Course Teacher	

Practical No. 14: Control relay state based on input from IR sensor

I. Practical Significance

To Control relay state based on input from IR sensor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards.

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 10.1 Interface IR sensor and relay with Arduino.
LLO 10.2 Write a program to control relay state based on input from IR sensor.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Relay modules are simply circuit boards that house one or more relays. They come in a variety of shapes and sizes, but are most commonly rectangular with 2, 4, or 8 relays mounted on them, sometimes even up to a 16 relays.

Relay modules contain other components than the relay unit. These include indicator LEDs, protection diodes, transistors, resistors, and other parts. But what is the module relay, which makes the bulk of the device? You may ask. Here are facts to note about it:

- A relay is an electrical switch that can be used to control devices and systems that use higher voltages. In the case of module relay, the mechanism is typically an electromagnet.
- The relay module input voltage is usually DC. However, the electrical load that a relay will control can be either AC or DC, but essentially within the limit levels that the relay is designed for.
- A relay module is available in an array of input voltage ratings: It can be a 3.2V or 5V relay module for low power switching, or it can be a 12 or 24V relay module for heavy-duty systems.
- The relay module information is normally printed on the surface of the device for ready reference. This includes the input voltage rating, switch voltage, and current limit.

VII. Actual Circuit diagram used in laboratory with related equipment rating

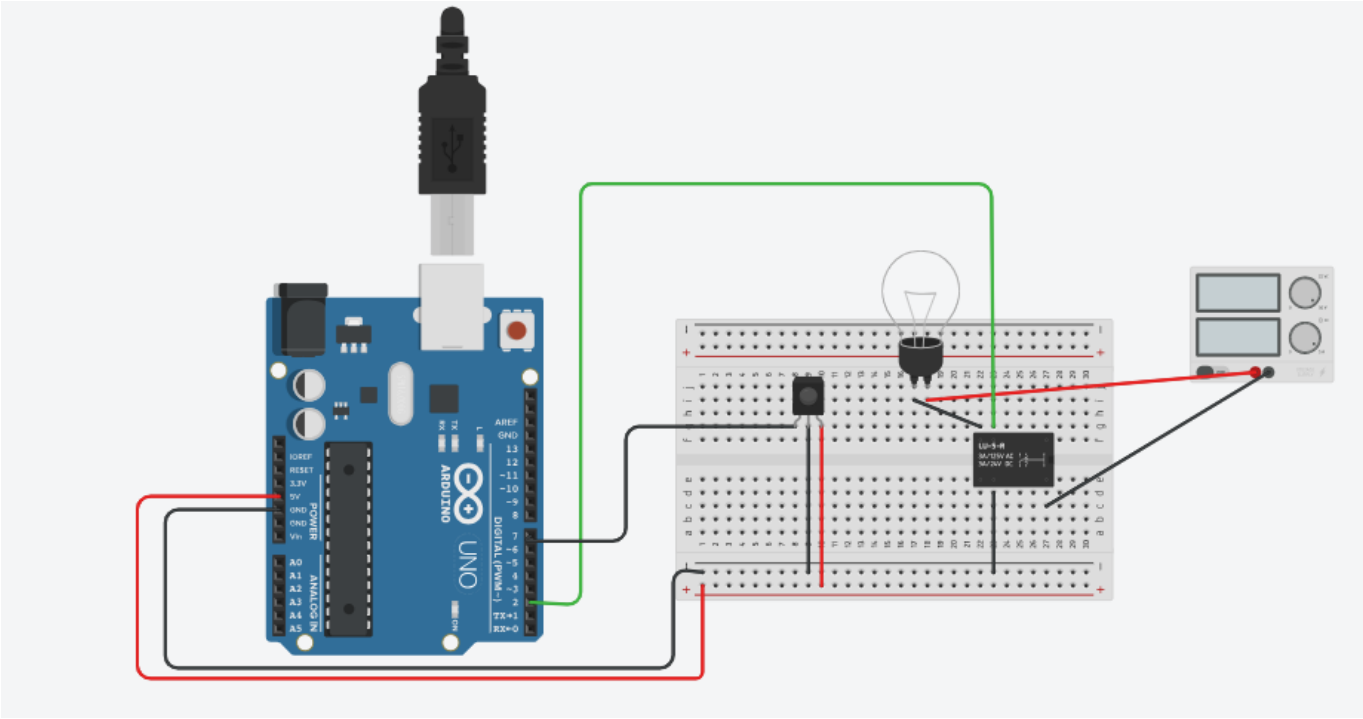


Fig.14.1 interfacing of IR sensor and relay module

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	IR sensor		01
4	Relay module		01

IX Precautions to be followed

Connect IR sensor and relay as per circuit diagram.

X Procedure

1. Interface IR sensor and relay to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

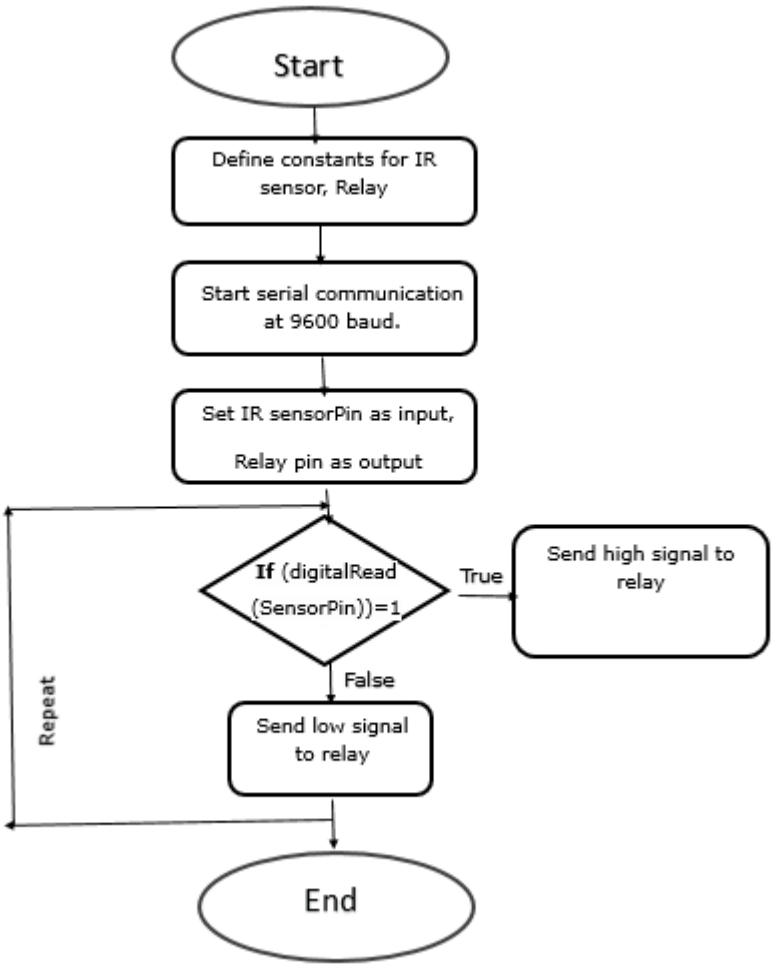
XI. Sample program

Step 1: Algorithm

8. Start
9. Initialize pin 7 as IRSensor and pin 2 as relay

10. Set IRSensor pin mode to Input
11. Set relay pin mode to output
12. Read the status of the IRSensor
13. If the status is 1
    a. Turn on relay
14. Else
    a. Turn off relay

Step 2: Flowchart



Step 3: Program

```
int IRSensor = 7;// connect ir sensor to arduino pin 7

  int relay = 2;//connect relay  to arduino pin 2

void setup()

{
```

```
  pinMode (IRSensor, INPUT);
   // sensor pin INPUT
   pinMode (relay, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int statusSensor = digitalRead (IRSensor);

  if (statusSensor == 1)
{
    digitalWrite(relay,HIGH);
  }
else
  {
    digitalWrite(relay,LOW);
  }
 }
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. State applications of IR sensor

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 15: Switch the LED ON/OFF on detection of obstacles using PIR

I. Practical Significance

To change the status of LED on detection of obstacles using PIR

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. . Laboratory Learning Outcome(s)

LLO 15.1: Interface PIR sensor and LED with Arduino.

LLO 15.2: Switch the LED ON/OFF on detection of obstacles using PIR sensor

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

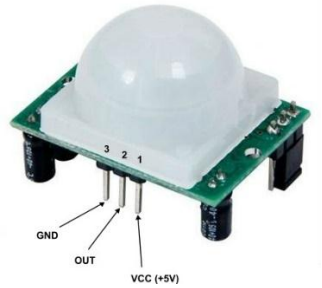
c. Follow ethical practices

VI. Relevant Theoretical Background

PIR sensor: PIR sensors allow you to sense motion. They are small, inexpensive, low-power, easy to use

A passive infrared sensor is an electronic sensor that measures infrared light radiating from objects. PIR sensors mostly used in PIR-based motion detectors

The below image shows a typical pin configuration of the PIR sensor, which is quite simple to understand the pinouts



- Pin1 corresponds to the drain terminal of the device, which connected to the positive supply 5V DC.
- Pin2 corresponds to the source terminal of the device, which connects to the ground terminal via a 100K or 47K resistor. The Pin2 is the output pin of the sensor. The pin 2 of the sensor carries the detected IR signal to an amplifier.
- Pin3 of the sensor connected to the ground.

PIR sensor i.e. Passive Infrared Sensor, passive word indicates PIR Sensor does not generate or radiate any energy for detection purposes.

PIR Sensors don't detect or measure "**HEAT**"; they detect the infrared radiation emitted or reflected from objects.

They are small, inexpensive, low power and easy to use. They are commonly found at home, medical, factories etc. areas.

VII. Actual Circuit diagram used in laboratory with related equipment rating

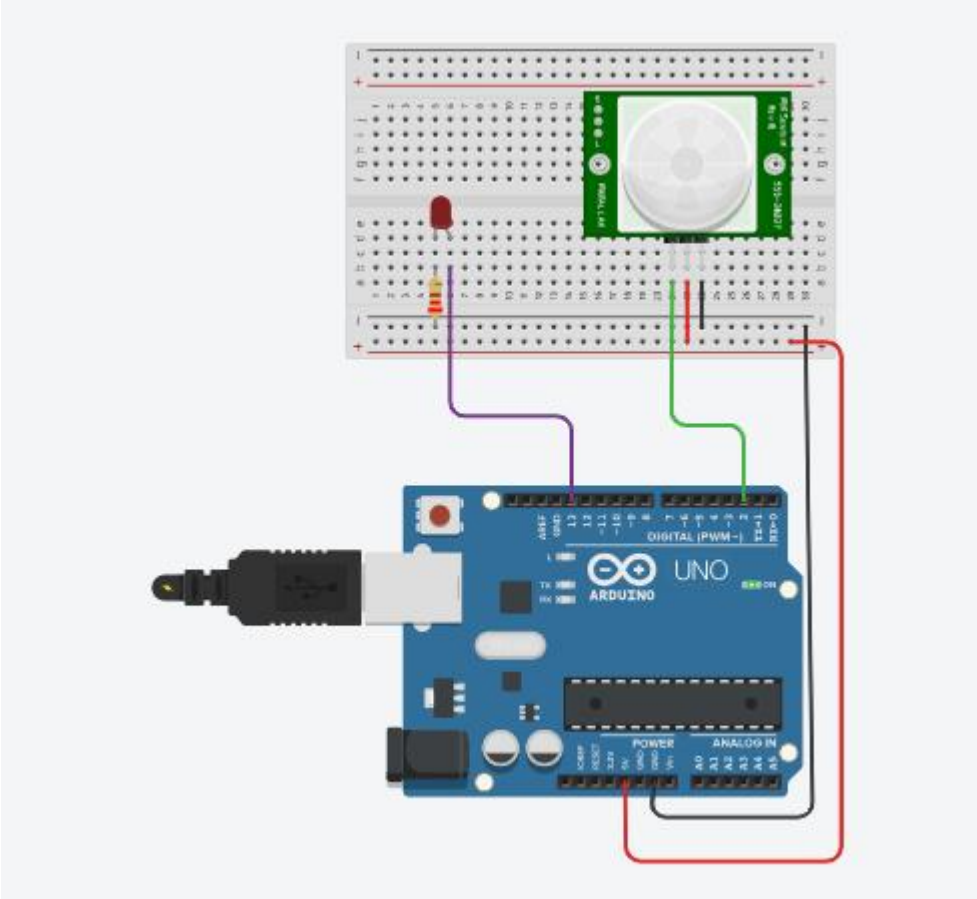


Fig 15.1 Detect motion using PIR sensor and LED

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quanity
1	Arduino IDE software		01
2	Arduino Uno		01
3	PIR sensor		01
4	LED		01

IX. Precautions to be followed

Connect pins of PIR sensor and buzzer in accurate manner.

X . Procedure

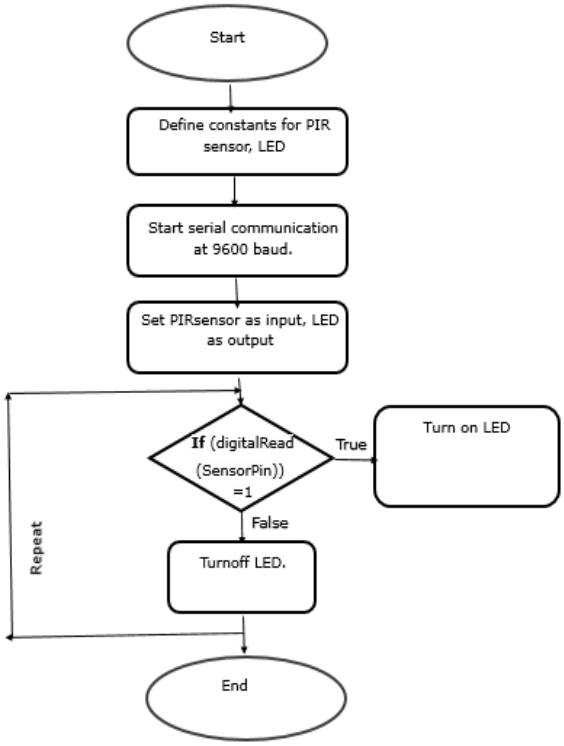
1. Interface PIR sensor and LED to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Set the PIRSesosor pin (2) as an input and LED pin (13) output.
2. Read the value from PIRSesosor pin
3. Checks if the sensor value is high which indicates the presence of motion.
4. If motion is detected, it turns on the led.
5. If no motion is detected, it turns off the led.

Step 2: Flowchart



Step 3: Program

```
int PIRSensor = 2; // connect PIR sensor to arduino pin 2
int LED  = 13; // connect LED  to arduino pin 13


 void setup()
{
 pinMode(LED, OUTPUT);
 pinMode(PIRSensor, INPUT);
 Serial.begin(9600);
}

void loop()
{
 digitalWrite(2, LOW);
 Serial.println(digitalRead(2));
 if (digitalRead(2) == 1) {
  digitalWrite(13, HIGH);
 } else {
  digitalWrite(13, LOW);
 }
 delay(10); // Delay a little bit to improve simulation performance
}
```

XII. Results (Output of the Program)

……

……

……

XIII. Conclusion

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain difference between PIR sensor and IR sensor

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook
3. https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview
4. Cornel M Amariei: Arduino Development Cookbook

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 16: Measure the Distance between sensor and object and ring the buzzer when obstacle is detected in some specified range of distance.

I. Practical Significance

To Measure the Distance between sensor and object and ring the buzzer when obstacle is detected in some specified range of distance

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards.

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 16.1 Interface Ultrasonic Sensor, Buzzer with Arduino.
LLO 16.2 Write program to start the buzzer when obstacle is detected in some specified range of distance.

.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

An ultrasonic Sensor is a device used to measure the distance between the sensor and an object without physical contact. This device works based on time-to-distance conversion.

Working Principle of Ultrasonic Sensor: Ultrasonic sensors measure distance by sending and receiving the ultrasonic wave. The ultrasonic sensor has a sender to emit the ultrasonic waves and a receiver to receive the ultrasonic waves. The transmitted ultrasonic wave travels through the air and is reflected by hitting the Object. Arduino calculates the time taken by the ultrasonic pulse wave to reach the receiver from the sender.

We know that the speed of sound in air is nearly 344 m/s,
So, the known parameters are time and speed (constant). Using these parameters, we can calculate the distance traveled by the sound wave.
Formula: Distance = Speed * Time

VII. Actual Circuit diagram used in laboratory with related equipment rating

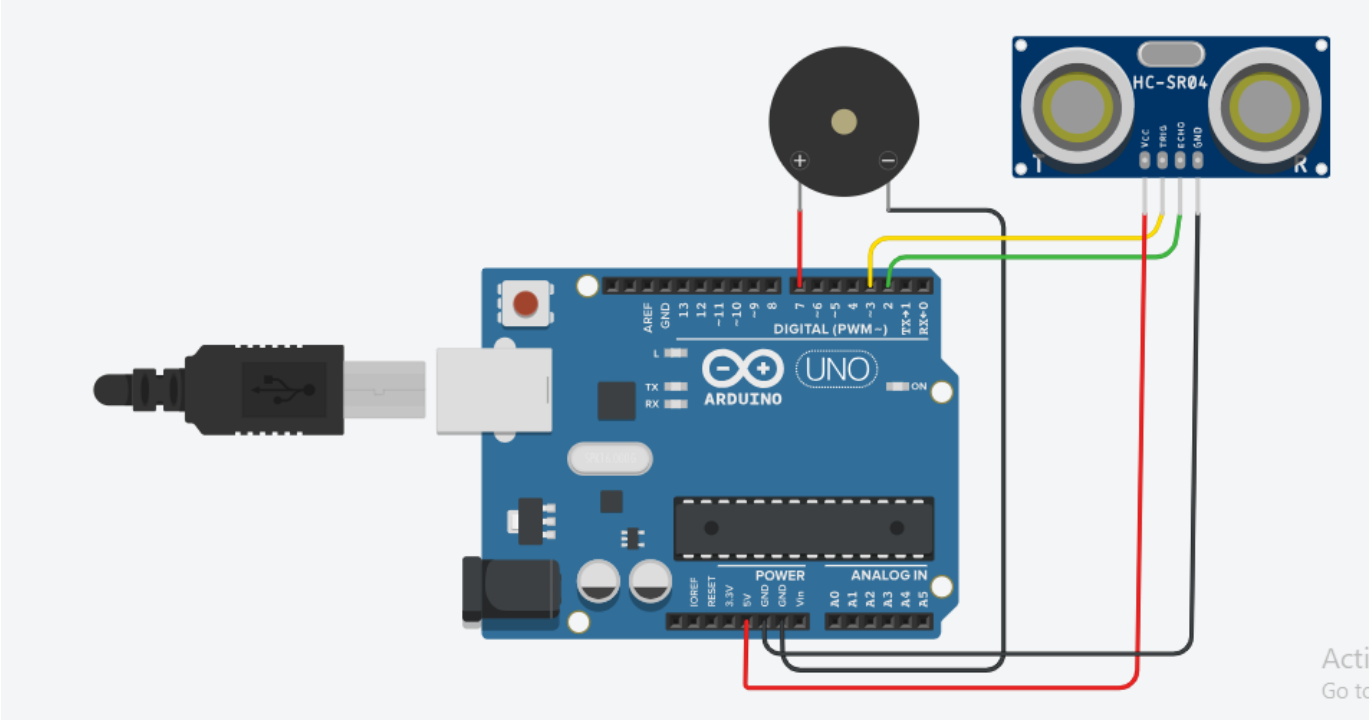


Fig 16.1Ciruit diagram to detect obstacle and ring the buzzer

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Ultrasonic sensor		01
4	Buzzer		01

IX .Precautions to be followed

Connect ultrasonic sensor and buzzer as per circuit diagram

X .Procedure

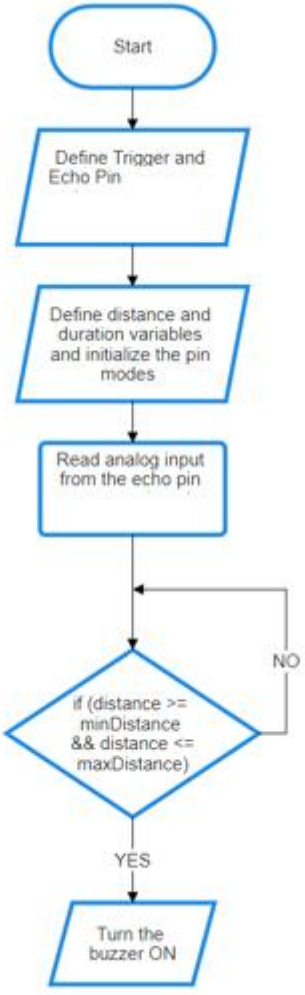
1. Interface ultrasonic sensor and buzzer to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop assembly program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Define trigger and echo pin
2. Define distance and duration variable
3. Initialize pin mode
4. Read input from echo pin
5. Check if distance >= mindistance && distance<= max distance
6. if true turn buzzer on
7. if false turn buzzer off
8. Repeat loop .

Step 2: Flowchart



Step 3: Program

```
  // Define pins
const int trigPin = 3;     // Trigger pin of ultrasonic sensor
const int echoPin = 2;     // Echo pin of ultrasonic sensor
const int buzzerPin = 7;  // Buzzer pin

// Define variables
long duration;
int distance;
int minDistance = 10;  // Minimum distance for obstacle detection (in centimeters)
int maxDistance = 50;  // Maximum distance for obstacle detection (in centimeters)

void setup() {

  // Initialize pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  // Trigger ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read duration of echo pulse
  duration = pulseIn(echoPin, HIGH);

  // Calculate distance in cm
  distance = duration * 0.034 / 2;

  // Check if distance is within specified range
  if (distance >= minDistance && distance <= maxDistance) {
    // If obstacle detected within range, activate buzzer
    digitalWrite(buzzerPin, HIGH);
  } else {
    // If no obstacle detected or outside specified range, turn off buzzer
    digitalWrite(buzzerPin, LOW);
  }

  // Delay for stability
  delay(100);
}
```

XII . Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Explain ultrasonic sensor pin diagram

……

……

……

……

……

XV. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)
3. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 17: Activate the alarm system if smoke detected

I. Practical Significance

To Play the Burglar Alarm if smoke detected

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards.

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 17.1 Interface Smoke Sensor with Arduino/ Raspberry Pi.
LLO 17.2 Write program to detect smoke and play Burglar Alarm if smoke detected.

V. Relevant Affective Domain related Outcomes

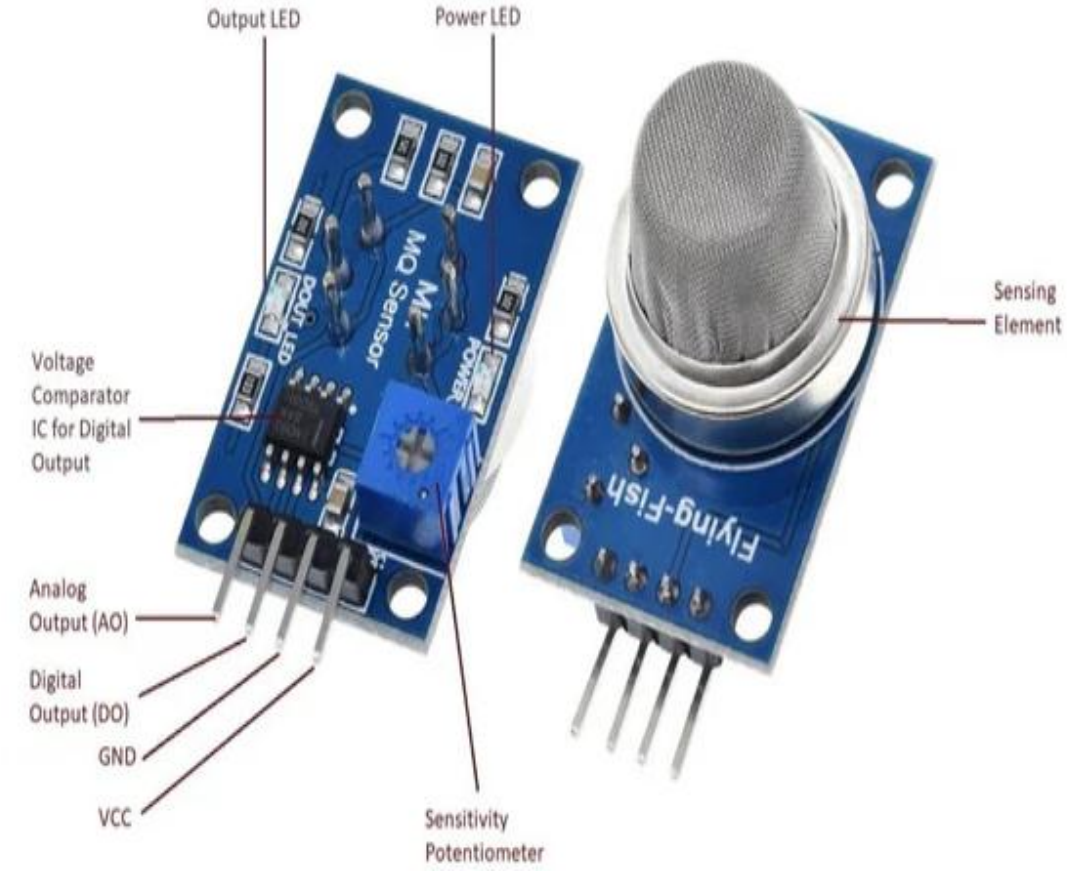
a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

MQ2 gas sensor is an electronic sensor used for sensing the concentration of gases in the air such as LPG, propane, methane, hydrogen, alcohol, smoke and carbon monoxide.

MQ2 gas sensor is also known as chemiresistor. It contains a sensing material whose resistance changes when it comes in contact with the gas. This change in the value of resistance is used for the detection of gas.

MQ2 is a metal oxide semiconductor type gas sensor. Concentrations of gas in the gas is measured using a voltage divider network present in the sensor. This sensor works on 5V DC voltage. It can detect gases in the concentration of range 200 to 10000ppm.



VII. Actual Circuit diagram used in laboratory with related equipment rating

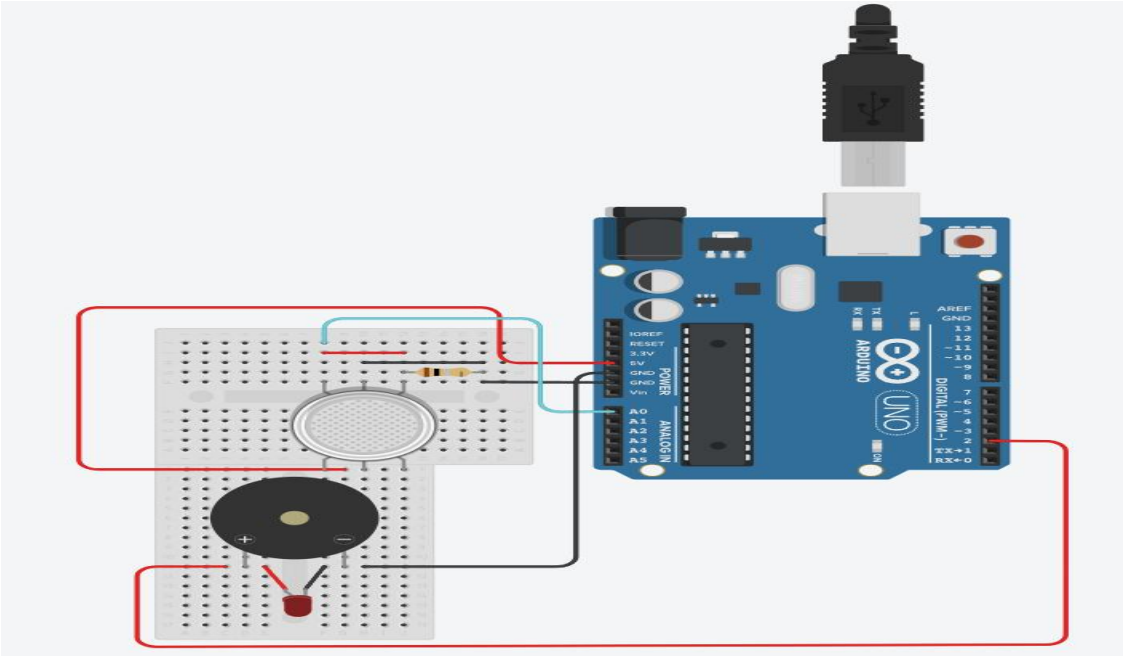


Fig 17.1 Interfacing of smoke sensor and buzzer

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Vibration detection sensor		01

IX Precautions to be followed

Connect smoke detection sensor in accurate manner.

X. Procedure

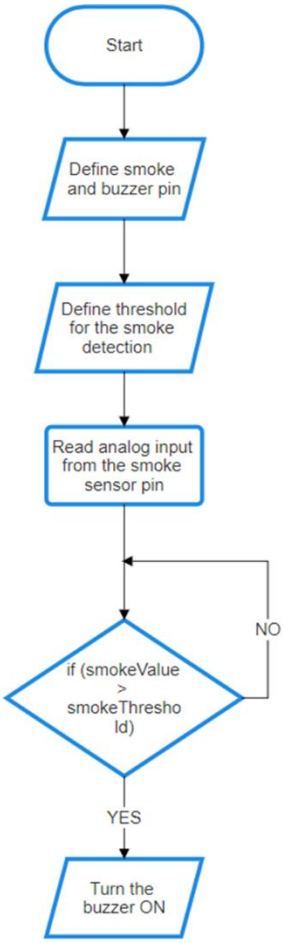
1. Interface smoke sensor and buzzer to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop assembly program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Define smoke and buzzer pin
2. Define threshold for the smoke detection
3. Initialize the pin mode.
4. Read analog input from the smoke sensor pin
5. Check if (smoke Value > smoke Threshold)
6. If true, turn the buzzer on
7. If false, turn the buzzer off
8. Repeat loop.

Step 2: Flowchart



Step 3: Program

```
int val = A0;


void setup()
{
  pinMode (2,OUTPUT);
  pinMode (A0,INPUT);
  Serial.begin(9600);
}
void loop()
{
  val = analogRead (A0);
  if (val <500 )
  {digitalWrite (2, LOW);}
  else
  {digitalWrite(2, HIGH);}
```

XII Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain the different types of gas sensors and their applications

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 18: Display percentage of moisture in soil using soil moisture sensor.

I. Practical Significance

To display percentage of moisture in soil using soil moisture sensor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 -Create IoT applications by interfacing various sensors and embedded boards.

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 18.1 Interface soil moisture Sensor with Arduino/ Raspberry Pi.

LLO 18.2 Write program to measure percentage of moisture in soil

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

Soil moisture sensors are devices used to measure the water content in soil. They are crucial for applications in agriculture, horticulture, and environmental science to ensure optimal watering of plants, manage irrigation systems, and study soil properties. Here's a detailed look at soil moisture sensors

By providing accurate and timely data on soil moisture levels, they help in making informed decisions that benefit both plants and the environment.

VII. Actual Circuit diagram used in laboratory with related equipment rating

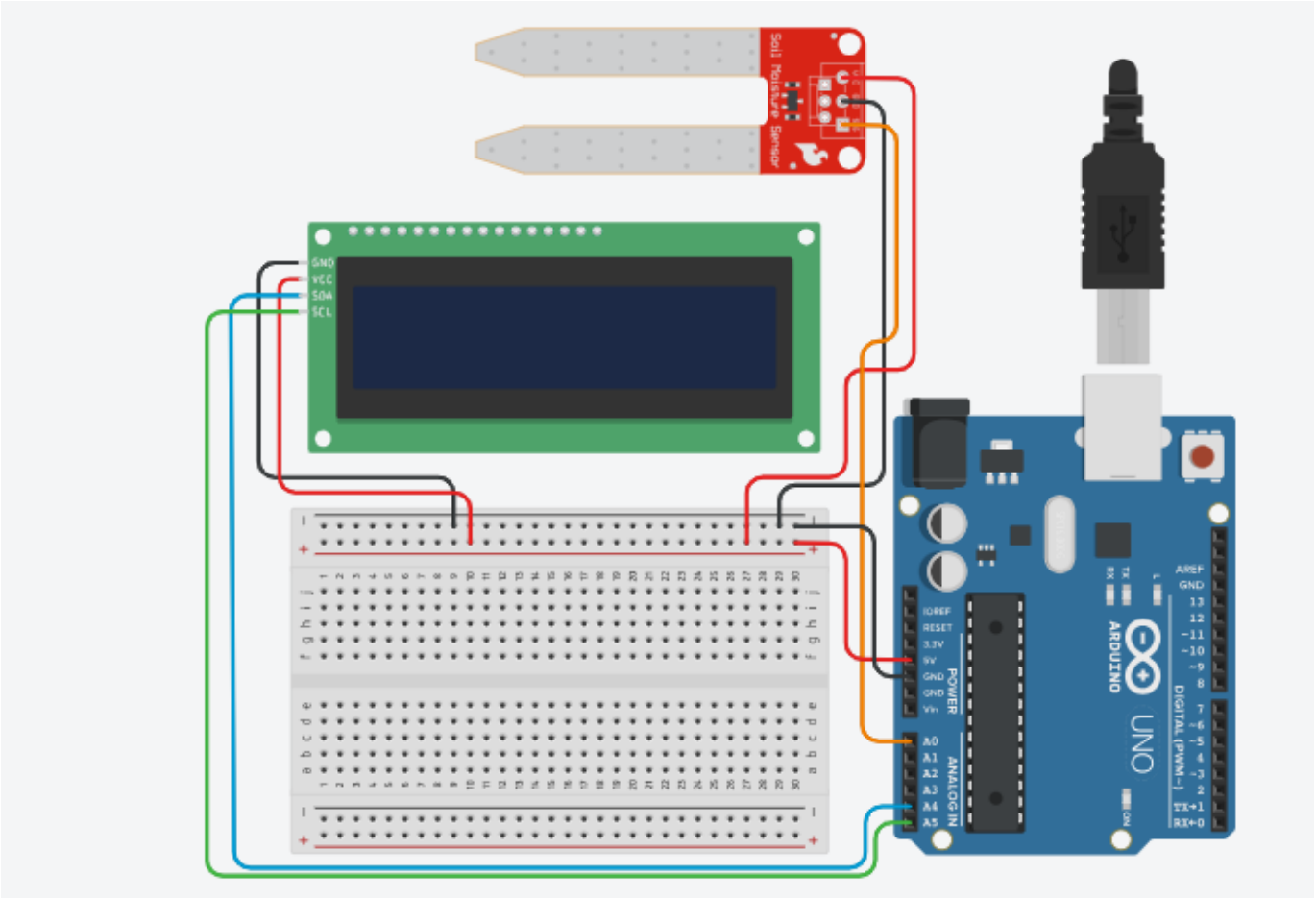


Fig 18.1 measure soil moistu

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Soil moisture sensor		01

IX .Precautions to be followed

Connect pins of soil moisture sensor and buzzer in accurate manner.

X. Procedure

1. Interface soil moisture sensor and buzzer to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.

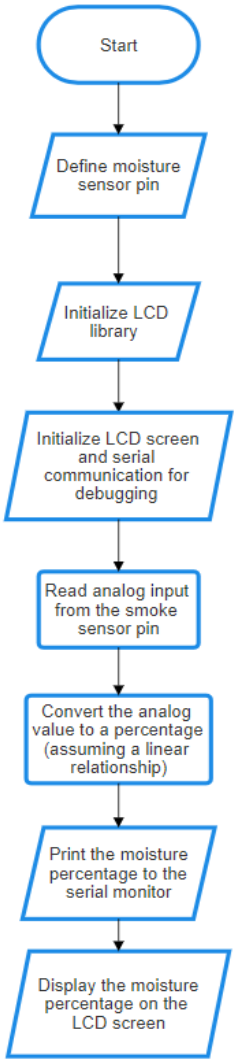
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

XI. Sample program

Step 1: Algorithm

1. Define moisture sensor pin
2. Initialize LCD library
3. Initialize LCD screen and serial communication for debugging
4. Read the analog value from the soil moisture sensor
5. Convert the analog value to a percentage (assuming a linear relationship)
6. Print the moisture percentage to the serial monitor
7. Display the moisture percentage on the LCD screen
8. Repeat loop.

Step 2: Flowchart



Step 3: Program

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(32, 16, 2);

const int moisturePin = A0;

void setup() {
  Serial.begin(9600); // initialize serial monitor
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Soil Moisture:");
}

void loop() {
  int moistureValue = analogRead(moisturePin);
  int moisturePercent = map(moistureValue, 0, 1023, 0, 100);

  lcd.setCursor(0, 1);
  lcd.print(moisturePercent);
  lcd.print("%");
  lcd.print(" ");

  Serial.println(moistureValue);
  //Serial.print("Moisture: ");
  //Serial.print(moisturePercent);
  //Serial.println("%");

  delay(200);
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain different types of soil moisture sensor

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 19: Upon detection of fire, activate the LED indicator and initiate the alarm system

I. Practical Significance

To detect the fire and turn ON LED and play the alarm

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO2 - Create IoT applications by interfacing various sensors and embedded boards
CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. . Laboratory Learning Outcome(s)

LLO 19.1 Interface fire detector sensor with NodeMCU.
LLO 19.2 Write program to display to glow LED and play the alarm when fire detected.

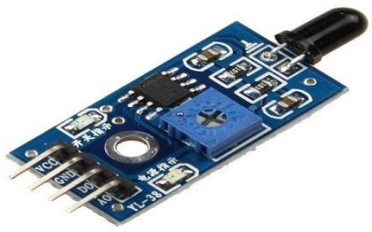
V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

. A flame-sensor is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting. It includes an alarm system, a natural gas line, propane & a fire suppression system. This sensor is used in industrial boilers. The main function of this is to give authentication whether the boiler is properly working or not. The response of these sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.
Flame Sensor Module

The pin configuration of this sensor is shown below. It includes four pins which include the following. When this module works with a microcontroller unit then the pins are



VII. Actual Circuit diagram used in laboratory with related equipment rating

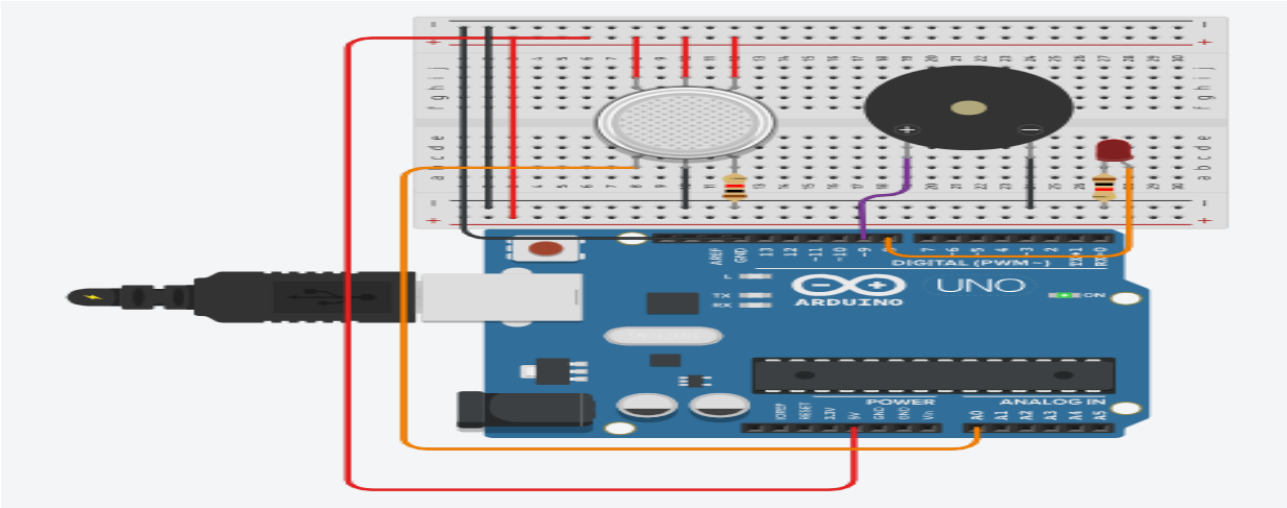


Fig 19.1 Interfacing flame sensor

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Fire sensor		01
4	Buzzer		01
5	LED		01

IX .Precautions to be followed

Connect pins of fire sensor in accurate manner.

X. Procedure

1. Interface fire sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

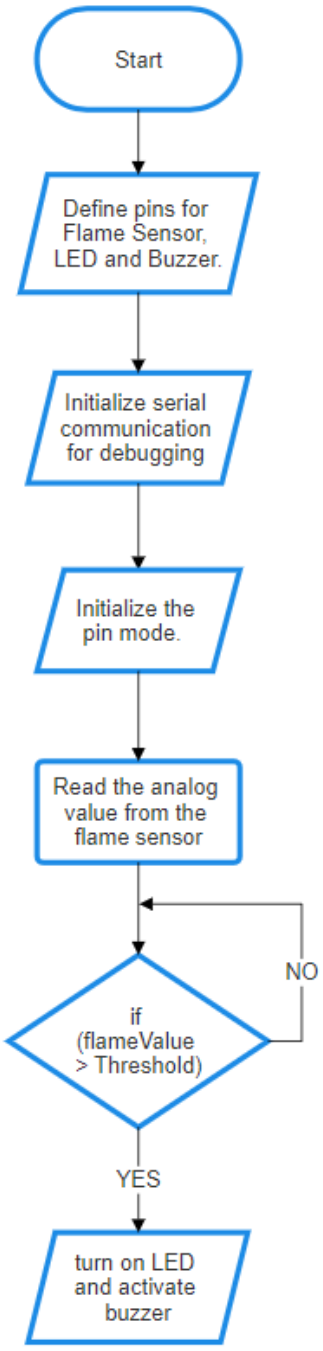
XI. Sample program

Step 1: Algorithm

1. Define pins for Flame Sensor, LED and Buzzer.
2. Initialize serial communication for debugging
3. Initialize the pin mode.
4. Read the analog value from the flame sensor
5. Check if the flame sensor value crosses the threshold

6. If true, turn on LED and activate buzzer
7. If false, turn off LED and stop the buzzer
8. Repeat loop.

Step2: Flowchart



Step3: program

```
int sensorValue = 0;// Set the initial sensorValue to 0

// The setup routine runs once when you press reset
void setup() {
  // Initialize the digital pin 8 as an output
  pinMode(8, OUTPUT);
```

```
  pinMode(9, OUTPUT);
  // Initialize serial communication at 9600 bits per second
  Serial.begin(9600);
}

// The loop routine runs over and over again forever
void loop() {
  // Read the input on analog pin 0(A0)
  int sensorValue = analogRead(A0);
  // Print out the value you read
  Serial.println(sensorValue, DEC);
  // If sensorValue is greater than 250
  if (sensorValue > 250) {
    // Activate digital output pin 8 - the LED will light up
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
  }
  else {
    // Deactivate digital output pin 8 - the LED will not light up
    digitalWrite(9, LOW);
    digitalWrite(8, LOW);
  }
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain how LCD can be interfaced in fire detection system

……

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 20: Display temperature value on serial monitor

I. Practical Significance

To display temperature value on serial monitor

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 20.1 Interface temperature sensor with Arduino.
LLO 20.2 Write program to sense temperature and display it to serial monitor.
.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Temperature Sensor is a device that converts temperature into an electrical signal that can be read by the Arduino. Common types include:

- Thermistor: A type of resistor whose resistance varies significantly with temperature.
- LM35: An analog linear temperature sensor.
- DHT11/DHT22: Digital temperature and humidity sensors.
- DS18B20: A digital thermometer

The analog signal from the sensor (voltage or resistance) is read by an Analog-to-Digital Converter (ADC) in the Arduino. For example, the Arduino's ADC converts the analog input voltage (0-5V) to a digital value (0-1023).

VII. Actual Circuit diagram used in laboratory with related equipment rating

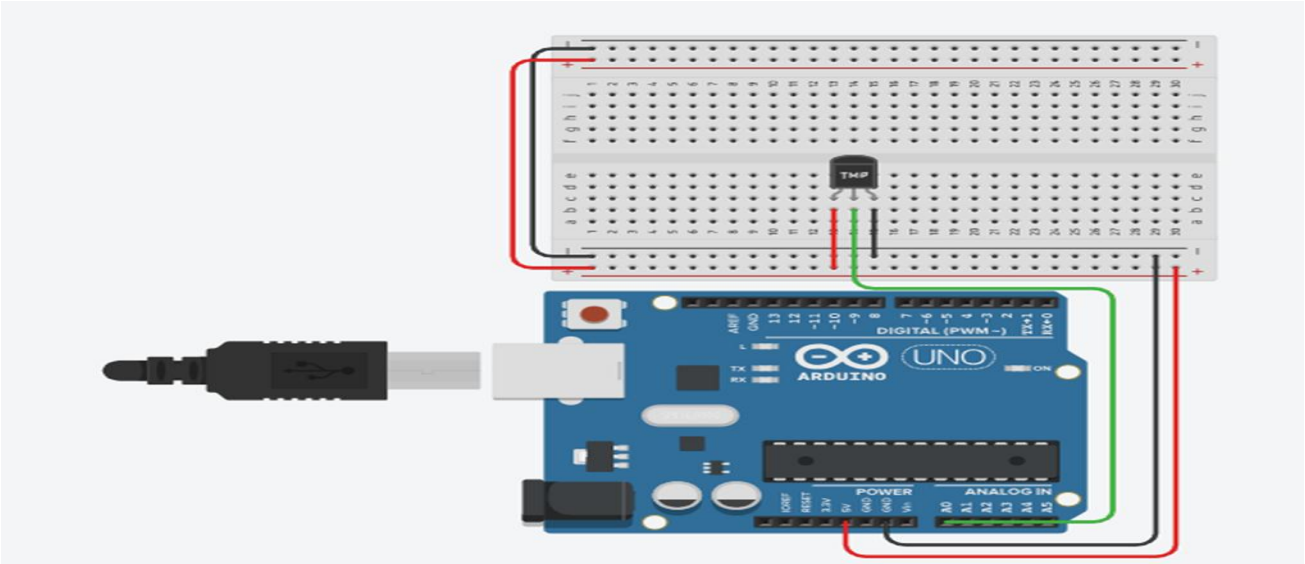


Fig 20.1Ciruit diagram to sense temperature

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Analog Temperature sensor		01

IX .Precautions to be followed

Connect temperature sensor as per circuit diagram

X .Procedure

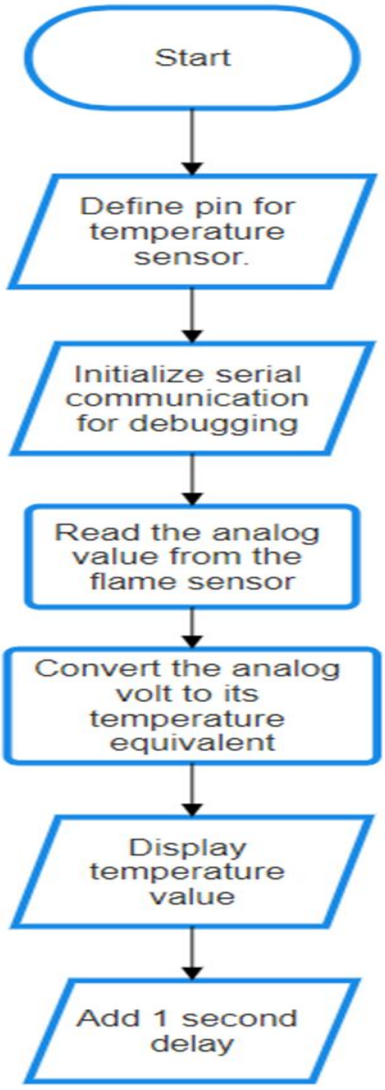
1. Interface temperature sensor to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Define pin for temperature sensor
2. Initialize serial communication for debugging
3. Read the analog volt from the sensor
4. Convert the analog volt to its temperature equivalent
5. Display temperature value
6. Add 1 second delay
7. Repeat loop.

Step 2: Flowchart



Step 3: Program

```
   float temp;
int tempPin = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  temp = analogRead(tempPin);
  // read analog volt from sensor and save to variable temp
  temp = temp * 0.48828125;
  // convert the analog volt to its temperature equivalent
  Serial.print("TEMPERATURE = ");
```

```
  Serial.print(temp); // display temperature value
  Serial.print("*C");
  Serial.println();
  delay(1000); // update sensor reading each one second
}
```

XII. Results (Output of the Program)

……

…..……

……..

……

……

XIII . Conclusion

……

…..……

……..

……

……..

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Write a program to make buzzer on when temperature reaches to certain threshold value

……

…..……

……..

……

……..

……

…..……

……..

XV. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)
3. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 21: Activate the Melody audio output utilizing a piezoelectric

I. Practical Significance

To Play Melody sound with a Piezo speaker

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains.

III. Course Level Learning Outcome(s)

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 21.1 Interface Piezo speaker with Arduino.
LLO 21.2 Write program to play Melody.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Piezo buzzers are used to beep, make sounds or play a melody. Piezo buzzers usually have two pins: a negative pin that connects to GND on the Arduino and a positive pin (connected to one of the digital pin slots on the Arduino) that receives control signals from the Arduino. The piezo buzzer can play sounds at different frequencies to create a melody. This can be achieved by using the tone() function to change the frequency of the voltage sent to the buzzer from the Arduino

VII. Actual Circuit diagram used in laboratory with related equipment rating

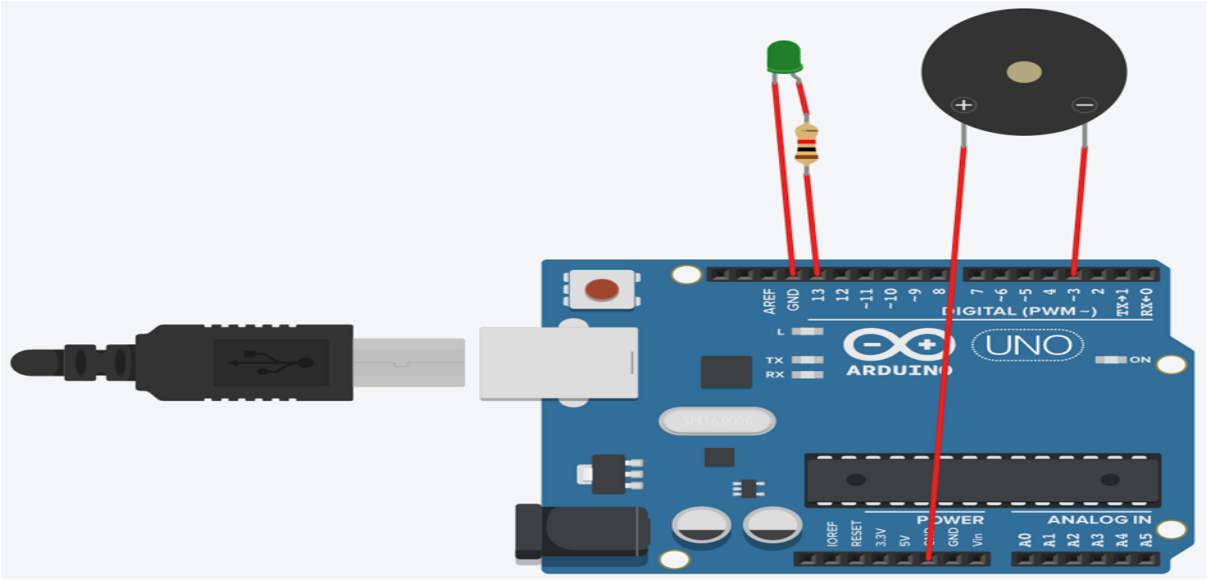


Fig 21.1 Interfacing of Piezo speaker

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qunatity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Piezo speaker		01

IX. Precautions to be followed

Connect Piezo speaker in accurate manner.

X. Procedure

1. Interface Piezo speaker to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop assembly program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

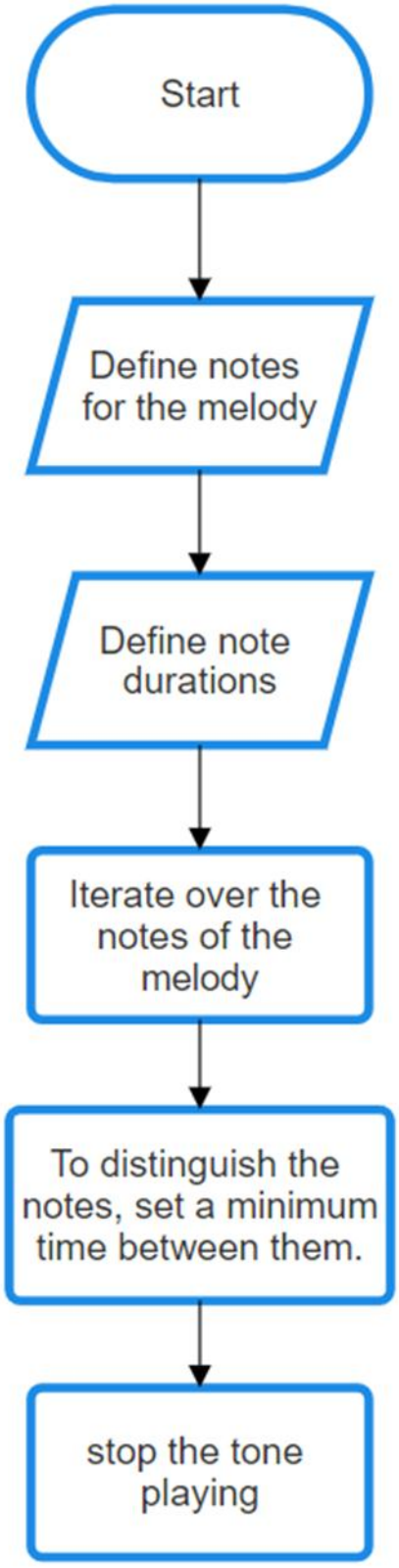
XI. Sample program

Step 1: Algorithm

1 Define notes for the melody
2 Define note durations
3 Iterate over the notes of the melody
4 To distinguish the notes, set a minimum time between them.

5 stop the tone playing

Step 2: Flowchart



Step 3: Program

```
#define NOTE_C4  262
```

```
#define NOTE_D4  294
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_G4  392
#define NOTE_A4  440
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_D5  587
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_G5  784
#define NOTE_A5  880
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_R   0

int melody[] = {
  NOTE_E4, NOTE_G4, NOTE_A4, NOTE_A4, NOTE_B4, NOTE_C5, NOTE_B4, NOTE_A4,
  NOTE_G4, NOTE_A4, NOTE_B4, NOTE_B4, NOTE_C5, NOTE_D5, NOTE_C5, NOTE_B4,
  NOTE_A4, NOTE_G4, NOTE_E4, NOTE_E4, NOTE_G4, NOTE_A4, NOTE_A4, NOTE_B4,
  NOTE_C5, NOTE_B4, NOTE_A4, NOTE_G4, NOTE_A4, NOTE_B4, NOTE_B4, NOTE_C5,
  NOTE_D5, NOTE_E5, NOTE_C5, NOTE_A4, NOTE_G4, NOTE_E4, NOTE_G4, NOTE_A4,
  NOTE_A4, NOTE_B4, NOTE_C5, NOTE_B4, NOTE_A4, NOTE_G4, NOTE_A4, NOTE_B4,
  NOTE_B4, NOTE_C5, NOTE_D5, NOTE_E5, NOTE_C5, NOTE_A4, NOTE_G4, NOTE_E4,
  NOTE_C4, NOTE_C4, NOTE_C4, NOTE_C4
};

int noteDurations[] = {
  8, 8, 4, 4, 4, 4, 8, 8,
  8, 8, 4, 4, 4, 4, 8, 8,
  8, 8, 4, 4, 4, 4, 8, 8,
  8, 8, 4, 4, 4, 4, 8, 8,
  8, 8, 4, 4, 4, 4, 8, 8,
  8, 8, 4, 4, 4, 4, 8, 8,
  8, 8, 4, 4, 4, 4, 8, 8,
  1, 1, 1, 1
};
void setup() {
  for (int thisNote = 0; thisNote < sizeof(melody) / sizeof(melody[0]); thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(7, melody[thisNote], noteDuration);

    delay(1000);

  }
}
void loop() {
  // No need for anything here, we're just playing the melody once in setup()
}
```

XII. Results (Output of the Program)

……

……

……

……

……

XIII. Conclusion

……

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Describe tone() function

……

……

……

……

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook
3. https://www.instructables.com/How-to-use-a-Buzzer-Arduino-Tutorial/

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 22: Control action using Relay based on temperature value

I. Practical Significance

To Control action using Relay based on temperature value

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 22.1 Interface Temperature sensor, Relay with Arduino.
LLO 22.2 Write a program to turn it ON/OFF when Temperature increases or decreases.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices

VI. Relevant Theoretical Background

Relays are a type of switch within an electronic system. Relays control one electrical circuit by opening and closing contacts in another circuit. As relay diagrams show, when a relay contact is normally open (NO), there is an open contact when the relay is not energized. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized.

VII. Actual Circuit diagram used in laboratory with related equipment rating

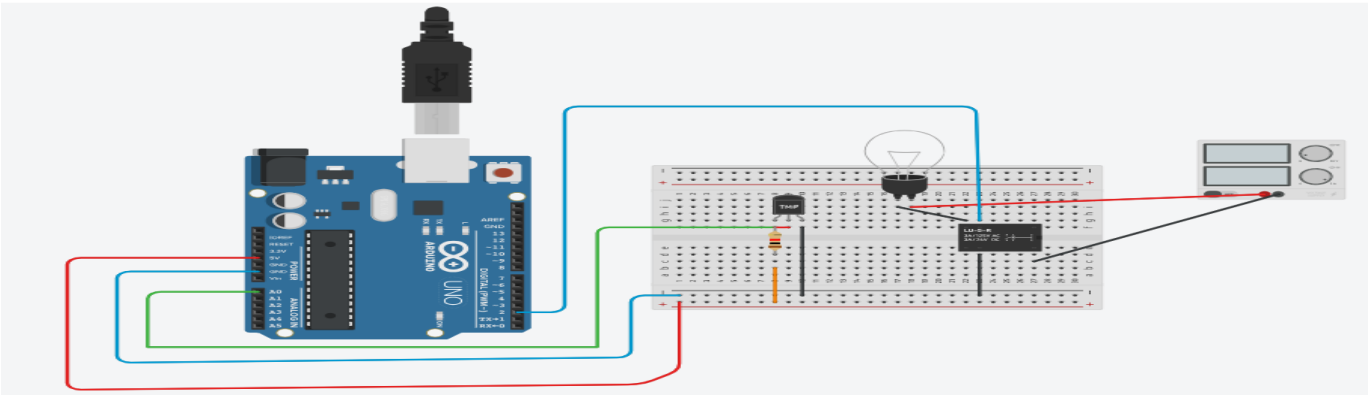


Fig 22.1 Control action using Relay based on temperature value

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quanity
1	Arduino IDE software		01
2	Arduino Uno		01
3	Temperature sensor		01
4	Relay		01
5	Light bulb		01

IX .Precautions to be followed

Connect pins of temperature sensor and relay in accurate manner.

X . Procedure

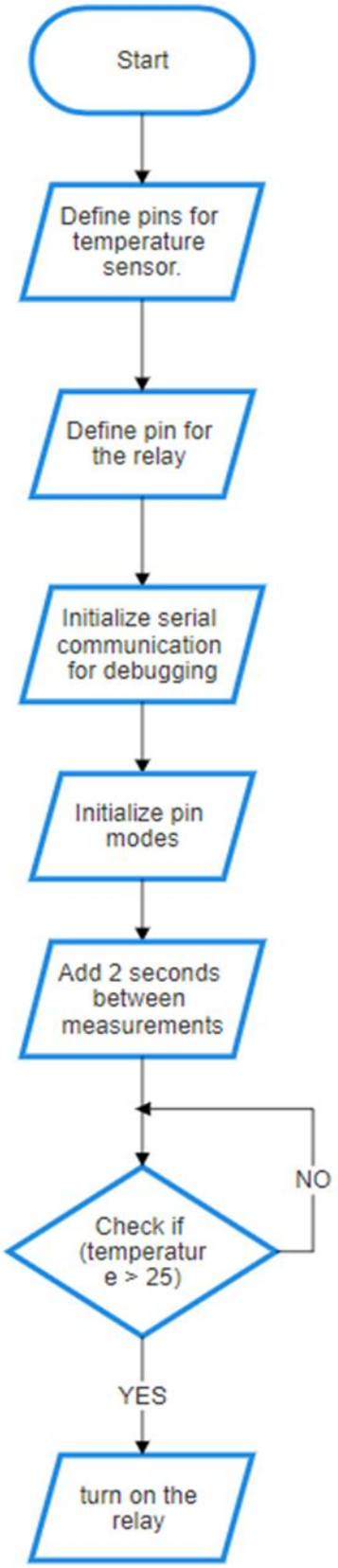
1. Interface temperature sensor and relay to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

XI.. Sample program

Step 1: Algorithm

1. Define pins for temperature sensor
2. Define pin for the relay
3. Initialize serial communication for debugging
4. Initialize pin modes
5. Add 2 seconds between measurements
6. Check if (temperature > 25)
7. If true, turn on the relay if temperature is above 25°C, and print “Relay turned ON”.
8. If false, turn off the relay if temperature is below or equal to 25°C, and print “Relay turned OFF”.
9. Repeat loop.

Step 2: Flowchart



Step 3: Program

int sensorValue = 0;

```
void setup()
{
 pinMode(A0, INPUT);
 Serial.begin(100);
 pinMode(2, OUTPUT);
}
void loop()
{
 sensorValue = analogRead(A0);
 int celsius = sensorValue/2;
 Serial.println( celsius);
 if(celsius == 25)
 {
 digitalWrite(2, HIGH);
 }
 else{
 digitalWrite(2, LOW);
 }
 delay(500);

}
```

XII. Results (Output of the Program)

……

……

……..

……..…

XIII . **Conclusion**

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. State and explain different types of relay

……..……

……

……..

……..…

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 23: Display 0 to 9 numbers continuously on seven segment display

I. Practical Significance

Display 0 to 9 numbers continuously on seven segment display

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. . Laboratory Learning Outcome(s)

LLO 23.1 Interface seven segment display with Arduino.

LLO 23.2 Write a program to display 0 to 9 numbers continuously.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

c. Follow ethical practices

VI. Relevant Theoretical Background

. 7-segment display is commonly used in electronic display devices for decimal numbers from 0 to 9 and in some cases, basic characters. The use of light-emitting diodes (LEDs) in seven-segment displays made it more popular, whereas of late liquid crystal displays (LCD) have also come into use.

Seven-Segment Display Working

Seven-segment devices are generally made up of LEDs. These LEDs will glow when they are forward-biased. The intensity of the LEDs depends on the forward current.

So, a sufficient forward current has to be provided to these LEDs to glow with full intensity. This is provided by the driver and is applied to the seven segments.

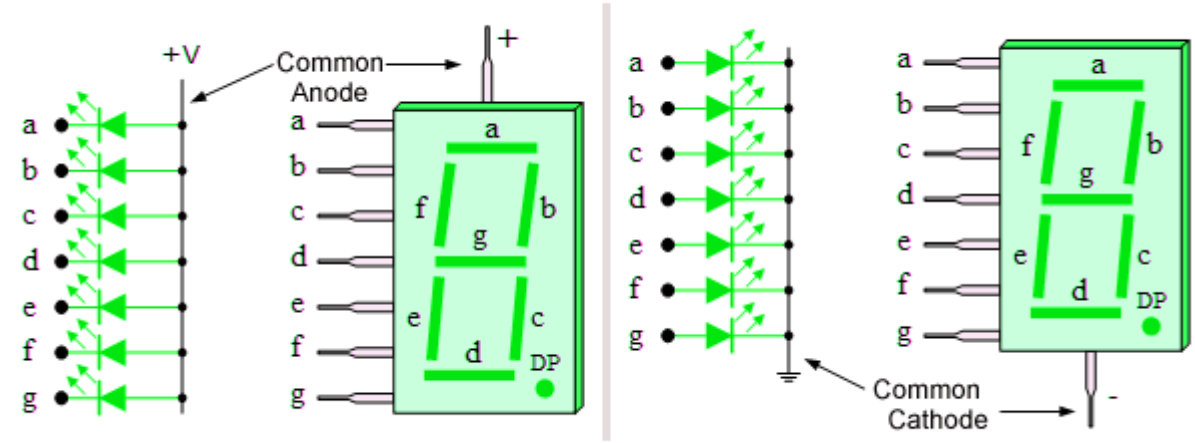
There are basically 2 types of seven-segment LED display

1. Common Anode 7 Segment Display:

All the Negative terminals (Anode) of all the 8 LEDs are connected together. All the positive terminals are left alone.

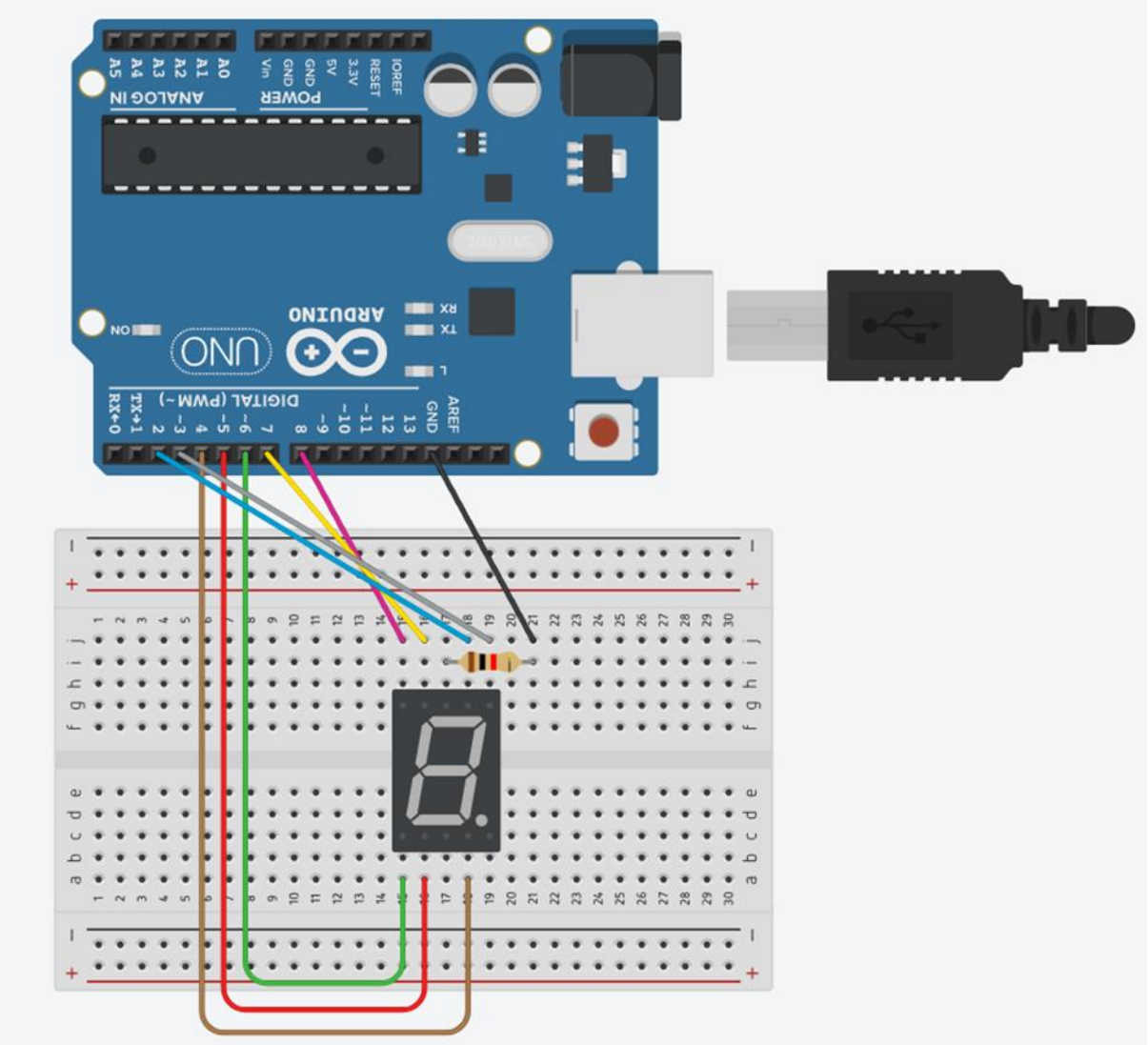
2. Common Cathode 7 Segment Display:

All the positive terminals (Cathode) of all the 8 LEDs are connected together. All the negative thermals are left alone



Common Anode and Common Cathode 7 Segment LED Display

VII. Actual Circuit diagram used in laboratory with related equipment rating



Fig, 23.1 interfacing of seven segment display

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qty
1	Arduino IDE software		01
2	Arduino Uno		01
3	7 segment display		01

IX .Precautions to be followed

Connect pins of seven segment display in accurate manner.

X .Procedure

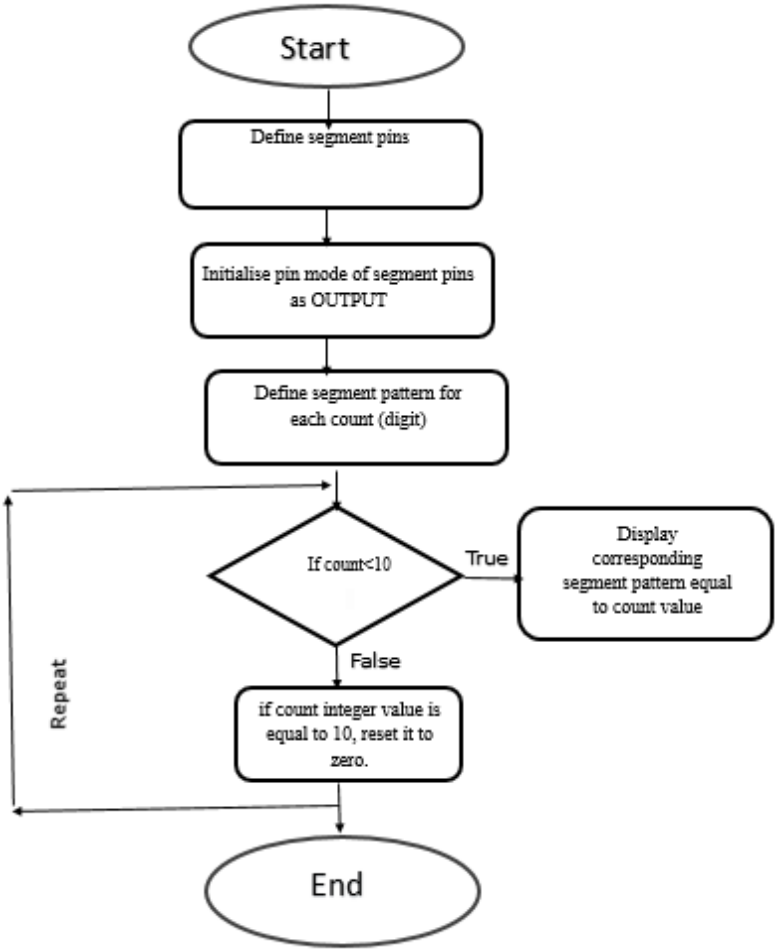
1. Interface seven segment display to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Define segment pins
2. Define the pattern for each number
3. Initialize pin mode of digit pins and set all digits off initially
4. Display each number for 1 second
5. Adjust this delay if needed
6. Repeat loop

Step2: Flowchart



Step3: program

```
#define segA 2// connecting segment A to PIN2

#define segB 3// connecting segment B to PIN3
```

```
#define segC 4// connecting segment C to PIN4

#define segD 5// connecting segment D to PIN5

#define segE 6// connecting segment E to PIN6

#define segF 7// connecting segment F to PIN7

#define segG 8// connecting segment G to PIN8

int COUNT=0; //count integer for 0-9 increment

void setup()
{
          for (int i=2;i<9;i++)
    {
           pinMode(i, OUTPUT); // taking all pins from 2-8 as output
          }
}

void loop()
{

switch (COUNT)

  {

   case 0://when count value is zero show ”0” on disp

       digitalWrite(segA, HIGH);

       digitalWrite(segB, HIGH);

       digitalWrite(segC, HIGH);

       digitalWrite(segD, HIGH);

       digitalWrite(segE, HIGH);

       digitalWrite(segF, HIGH);

       digitalWrite(segG, LOW);
```

```
        break;
    case 1:// when count value is 1 show”1” on disp
        digitalWrite(segA, LOW);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, LOW);
        digitalWrite(segE, LOW);
        digitalWrite(segF, LOW);
        digitalWrite(segG, LOW);
        break;
    case 2:// when count value is 2 show”2” on disp
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, LOW);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, HIGH);
        digitalWrite(segF, LOW);
        digitalWrite(segG, HIGH);
        break;
    case 3:// when count value is 3 show”3” on disp
        digitalWrite(segA, HIGH);
```

```
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, LOW);
        digitalWrite(segF, LOW);
        digitalWrite(segG, HIGH);
        break;
    case 4:// when count value is 4 show”4” on disp
        digitalWrite(segA, LOW);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, LOW);
        digitalWrite(segE, LOW);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
        break;
    case 5:// when count value is 5 show”5” on disp
        digitalWrite(segA, HIGH);
        digitalWrite(segB, LOW);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, LOW);
```

```
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
        break;
    case 6:// when count value is 6 show”6” on disp
        digitalWrite(segA, HIGH);
        digitalWrite(segB, LOW);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, HIGH);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
        break;
    case 7:// when count value is 7 show”7” on disp
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, LOW);
        digitalWrite(segE, LOW);
        digitalWrite(segF, LOW);
        digitalWrite(segG, LOW);
        break;
```

```
    case 8:// when count value is 8 show”8” on disp
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, HIGH);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
        break;
    case 9:// when count value is 9 show”9” on disp
        digitalWrite(segA, HIGH);
        digitalWrite(segB, HIGH);
        digitalWrite(segC, HIGH);
        digitalWrite(segD, HIGH);
        digitalWrite(segE, LOW);
        digitalWrite(segF, HIGH);
        digitalWrite(segG, HIGH);
        break;
        break;
            }
if (COUNT<10)
{
```

```
    COUNT++;
    delay(1000);///increment count integer for every second
}
if (COUNT==10)
{
    COUNT=0;// if count integer value is equal to 10, reset it to zero.
    delay(1000);
}
}
```

XII. Results (Output of the Program)

……
……
……
……

XIII. Conclusion

……
……
……
……

.

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Draw and Explain how to interface two seven segment display

……

……

……..

……..…

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	**Weightage- Process related: 60%**	**Marks-15**
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 24: Display simple message on I2C LCD

I. Practical Significance

To Display simple message on I2C LCD

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 24.1 Interface I2C LCD with Arduino.
LLO 24.2 Write program to display simple message

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

LCD Display:

LCD stands for **Liquid Crystal Display.** LCD is a flat-paneled display**.** It uses liquid crystals combined with polarized to display the content. LCD uses the light modulation property of LCD. LCD is available both in Monochrome and Multicolor. It cannot emit light directly without a backlight. In some LCDs, It displays the content only with the help of a backlight in a dark place.

I2C communication:

I2C or IIC stands for Inter-Integrated Communication. I2C is a serial communication interface to communicate with other I2C devices. I2C uses multi-master / multi slave method. I2C uses 2 lines named SCL and SDA for transmission/reception and another 2 lines for power supply and ground. Each and every I2C device has I2C address to identify. I2C addresses of multiple devices may have the same address. The address is in the format of "0x20"

I2C LCD:

I2C LCD uses I2C communication interface to transfer the information required to display the content. I2C LCD requires only 2 lines (SDA and SCL) for transferring the data. So, the complexity of the circuit is reduced.

VII. Actual Circuit diagram used in laboratory with related equipment rating

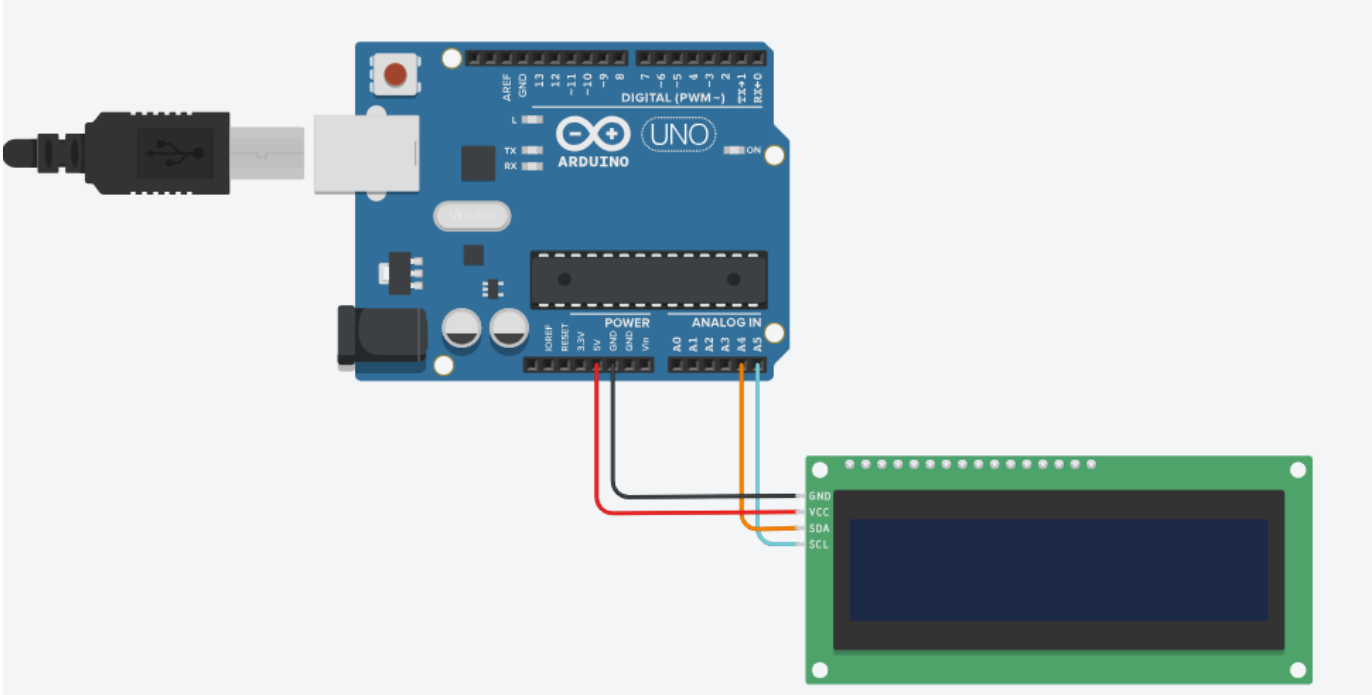


Fig 24.1Ciruit diagram to interface I2C LCD

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qty
1	Arduino IDE software		01
2	Arduino Uno		01
3	I2C LCD		01

IX .Precautions to be followed

Connect as per circuit diagram

X .Procedure

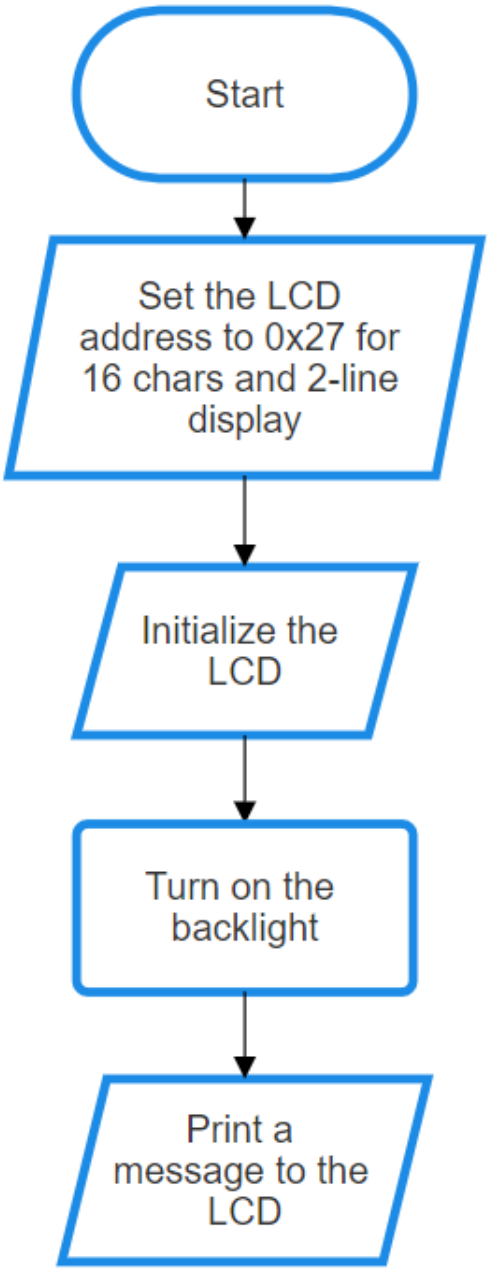
1. Interface I2C LCD to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Set the LCD address to 0x20 for 16 chars and 2-line display
2. Initialize the LCD
3. Turn on the backlight
4. Print a message to the LCD

Step 2: Flowchart



Step 3: Program

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x20 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x20, 16, 2);

void setup() {
  // Initialize the LCD
  lcd.init();
```

```
  // Turn on the backlight
  lcd.backlight();

  // Print a message to the LCD
  lcd.print("Hello, World!");
}

void loop() {
  // No need for anything in the loop for this example
}
```

XII. Results (Output of the Program)

……

……

……

……

XIII. Conclusion

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Write a program to build up simple calculator using I2C LCD

……

……

……

……

……

……

……

XV. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach

2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 25: Display POT value of potentiometer on LCD

I. Practical Significance

To Display POT value of potentiometer on LCD

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO3 - Create IoT applications by interfacing various actuators and embedded boards.

IV. Laboratory Learning Outcome(s)

LLO 25.1 Interface Potentiometer and LCD with Arduino.

LLO 25.2 Write program to display POT reading on LCD.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

c. Follow ethical practices.

VI.Relevant Theoretical Background

LCD Display:

LCD stands for **Liquid Crystal Display.** LCD is a flat-paneled display**.** It uses liquid crystals combined with polarized to display the content. LCD uses the light modulation property of LCD. LCD is available both in Monochrome and Multicolor. It cannot emit light directly without a backlight. In some LCDs, It displays the content only with the help of a backlight in a dark place.

I2C communication:

I2C or IIC stands for Inter-Integrated Communication. I2C is a serial communication interface to communicate with other I2C devices. I2C uses multi-master / multi slave method. I2C uses 2 lines named SCL and SDA for transmission/reception and another 2 lines for power supply and ground. Each and every I2C device has I2C address to identify. I2C addresses of multiple devices may have the same address. The address is in the format of "0x20"

I2C LCD:

I2C LCD uses I2C communication interface to transfer the information required to display the content. I2C LCD requires only 2 lines (SDA and SCL) for transferring the data. So, the complexity of the circuit is reduced.

VII. Actual Circuit diagram used in laboratory with related equipment rating

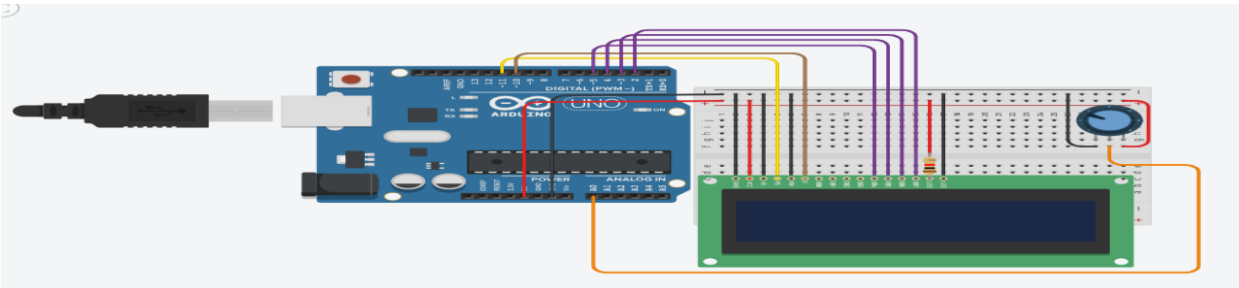


Fig 25.1 Interfacing I2C LCD and potentiometer

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Quantity
1	Arduino IDE software		01
2	Arduino Uno		01
3	I2C LCD		01
4	Potentiometer		01

IX Precautions to be followed

Connect I2C LCD as per circuit diagram

X .Procedure

1. Interface I2C LCD and potentiometer to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

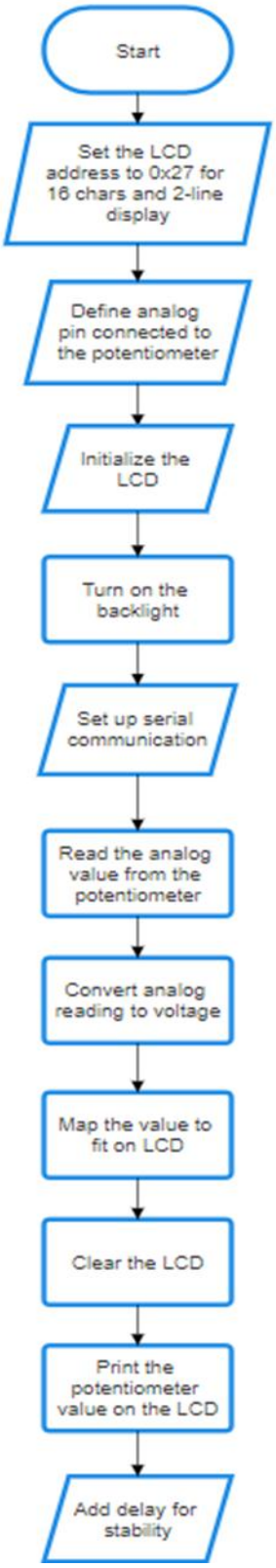
XI. Sample program

Step 1: Algorithm

Set the LCD address to 0x27 for 16 chars and 2-line display

1. Define analog pin connected to the potentiometer
2. Initialize the LCD
3. Turn on the backlight
4. Set up serial communication
5. Read the analog value from the potentiometer
6. Convert analog reading to voltage
7. Map the value to fit on LCD
8. Clear the LCD
9. Print the potentiometer value on the LCD
10. Add delay for stability

Step 2: Flowchart



Step 3: Program

```
#include <LiquidCrystal.h>
int sensorValue = 0;
```

```
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(11, 10, 5, 4, 3, 2);

void setup()
{
  pinMode(A0, INPUT);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.print("Starting System");
    delay(1000);
    lcd.clear();
    lcd.print("System On");
    delay(1500);

lcd.clear();

}

void loop()
{
lcd.setCursor(0,0);


  sensorValue = analogRead(A0);
  Serial.println(sensorValue);

lcd.setCursor(0,1);
lcd.print(sensorValue);
lcd.setCursor(4,1);
lcd.print("Sensor Value");


  delay(500);
  lcd.clear();
}
```

XII. Results (Output of the Program)

……..……

……

……..

……..…

XIII . Conclusion

……..……

……

……..

……..…

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Write applications of LCD

……..……

……

……..

……..…

……..

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)
2. Cornel M Amariei: Arduino Development Cookbook

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 26: Transfer sensor collected data to smartphone using Bluetooth

I. Practical Significance

To Transfer sensor collected data to smartphone using Bluetooth

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO4 - Develop IoT applications using IoT networking devices..

IV. Laboratory Learning Outcome(s)

LLO 26.1 Interface Bluetooth with Arduino/Raspberry Pi.

LLO 26.2 Write a program to send sensor data to smartphone using Bluetooth.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

c. Follow ethical practices.

VI. Relevant Theoretical Background

Bluetooth is one of the most popular wireless communication technologies because of its low power consumption, low cost and a light stack but provides a good range. In this project, data from a temperature sensor is collected by an Arduino and then transmitted to a smartphone via Bluetooth.

VII. Actual Circuit diagram used in laboratory with related equipment rating

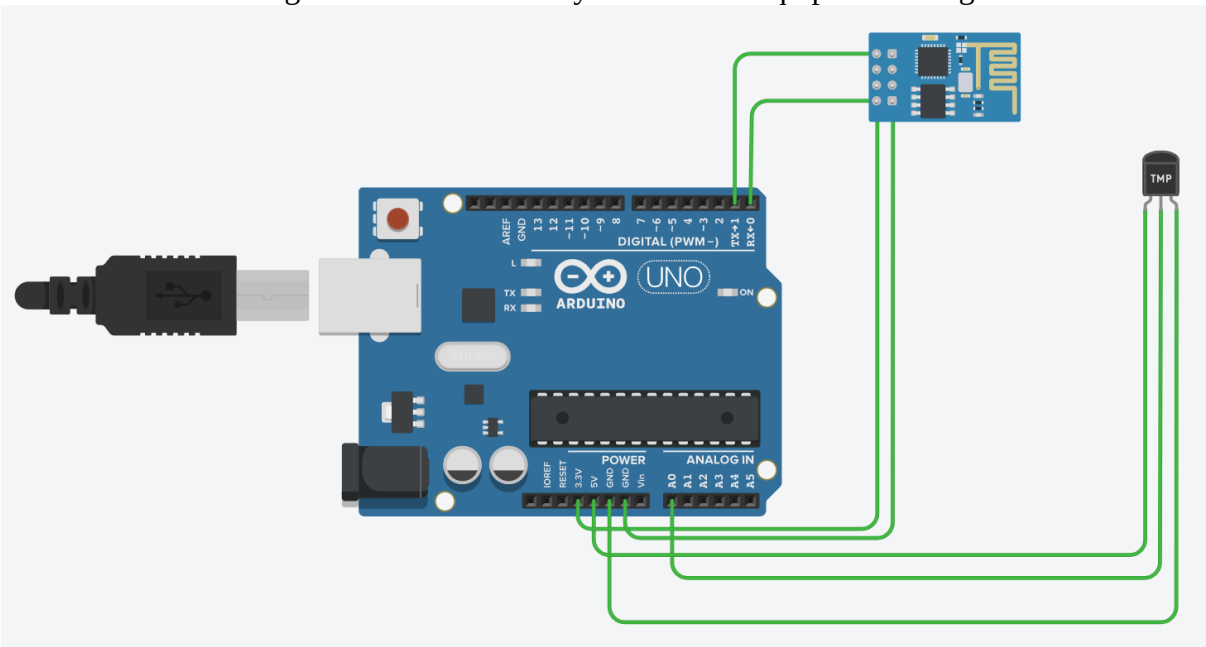


Fig 26.1 Transfer sensor collected data to smartphone using Bluetooth

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qty
1	Arduino IDE software		01
2	Arduino Uno		01
3	Bluetooth Module		01
4	Temperature sensor		01

IX .Precautions to be followed

Connect as per circuit diagram

X .Procedure

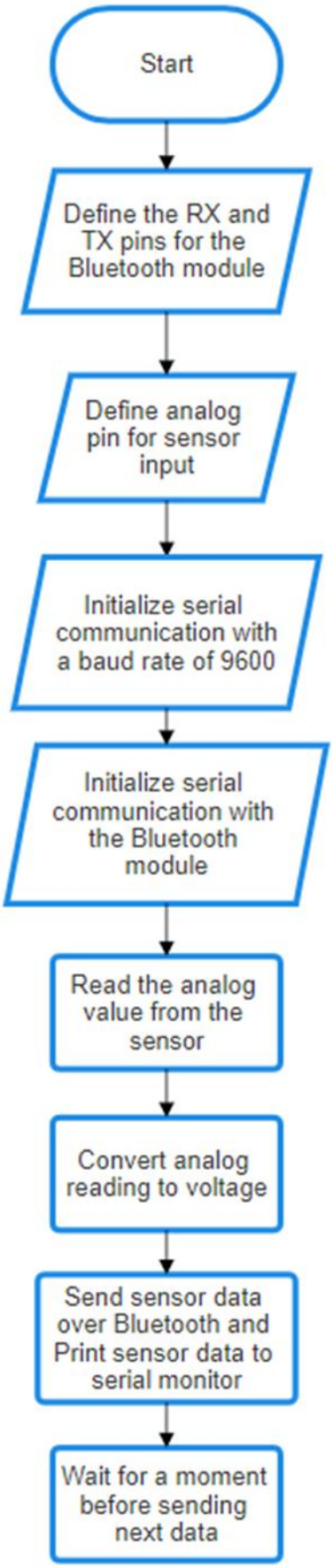
1. Interface Bluetooth module to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

XI. Sample program

Step 1: Algorithm

1. Define the RX and TX pins for the Bluetooth module
2. Create a Software Serial object to communicate with the Bluetooth module
3. Define analog pin for sensor input
4. Initialize serial communication with a baud rate of 9600
5. Initialize serial communication with the Bluetooth module
6. Read the analog value from the sensor
7. Convert analog reading to voltage
8. Send sensor data over Bluetooth
9. Print sensor data to serial monitor
10. Add delay
11. Search Arduino Bluetooth Control on play store and connect with your hardware device
12. Observe data on Application

Step 2: Flowchart



Step 3: Program

```
#include <SoftwareSerial.h>

// Define the RX and TX pins for the Bluetooth module
const int bluetoothTx = 2;  // TX pin of Bluetooth module connected to Arduino's digital pin 2
const int bluetoothRx = 3;  // RX pin of Bluetooth module connected to Arduino's digital pin 3

// Create a SoftwareSerial object to communicate with the Bluetooth module
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

// Analog pin for sensor input
const int sensorPin = A0;

void setup() {
  // Initialize serial communication with a baud rate of 9600
  Serial.begin(9600);

  // Initialize serial communication with the Bluetooth module
  bluetooth.begin(9600);
}

void loop() {
  // Read sensor value
  int sensorValue = analogRead(sensorPin);

  // Convert sensor value to voltage (assuming a 5V reference voltage and 10-bit ADC)
  float voltage = sensorValue * (5.0 / 1023.0);

  // Send sensor data over Bluetooth
  bluetooth.print("Sensor data: ");
  bluetooth.println(voltage);

  // Print sensor data to serial monitor
  Serial.print("Sensor data: ");
  Serial.println(voltage);

  // Wait for a moment before sending next data
  delay(1000);
}
```

XII. Results (Output of the Program)

……

……

……

……

XIII. Conclusion

……

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Explain Bluetooth module

……

……

……

……

……

XV. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 27: Display the message on serial monitor when image is captured

I. Practical Significance

To Display the message on serial monitor when image is captured

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO4 - Develop IoT applications using IoT networking devices.

IV. Laboratory Learning Outcome(s)

LLO 27.1 Connect Camera module with Arduino.

LLO 27.2 Write program to display the message on serial monitor when image is captured.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

b. Maintain tools and equipment.

c. Follow ethical practices.

VI.Relevant Theoretical Background

OV7670 Camera Module is a FIFO camera Module available from different Manufacturers with different pin Configurations. TheOV7670 provides full frame, windowed 8-bit images in a wide range of formats. The image array is capable of operating at up to 30 frames per second (fps) in VGA. The OV7670 includes

Image Sensor Array(of about 656 x 488 pixels)
Timing Generator
Analog Signal Processor
A/D Converters
Test Pattern Generator
Digital Signal Processor(DSP)
Image Scaler
Digital Video Port
LED and Strobe Flash Control Output

VII. Actual Circuit diagram used in laboratory with related equipment rating

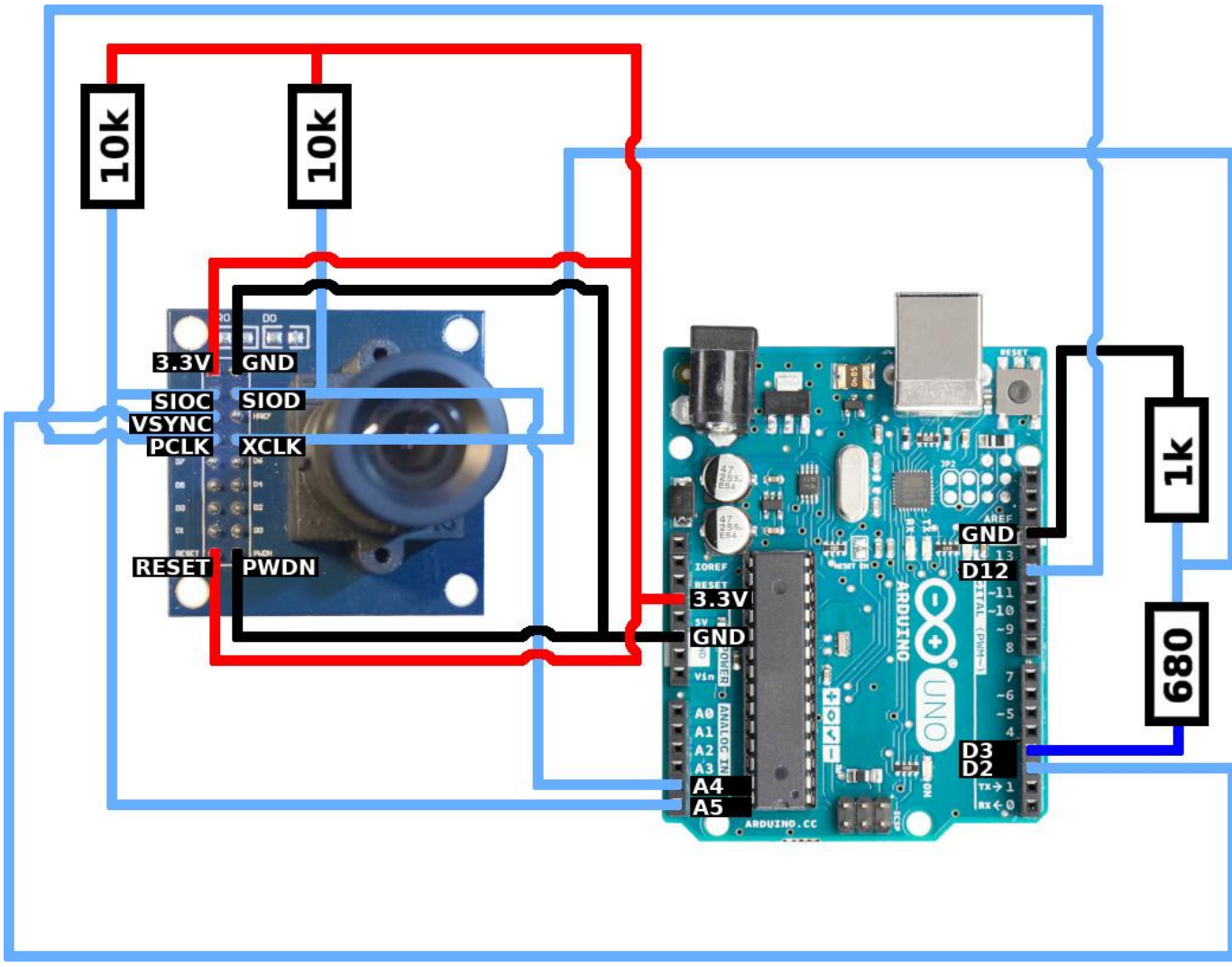


Fig 27.1 Interfacing camera module

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qty
1	Arduino IDE software		01
2	Arduino Uno		01
3	Camera module		01

IX Precautions to be followed

Connect camera module as per circuit diagram

X Procedure

1. Interface to Arduino as per circuit diagram shown in fig
2. Write algorithm for given problem

3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output

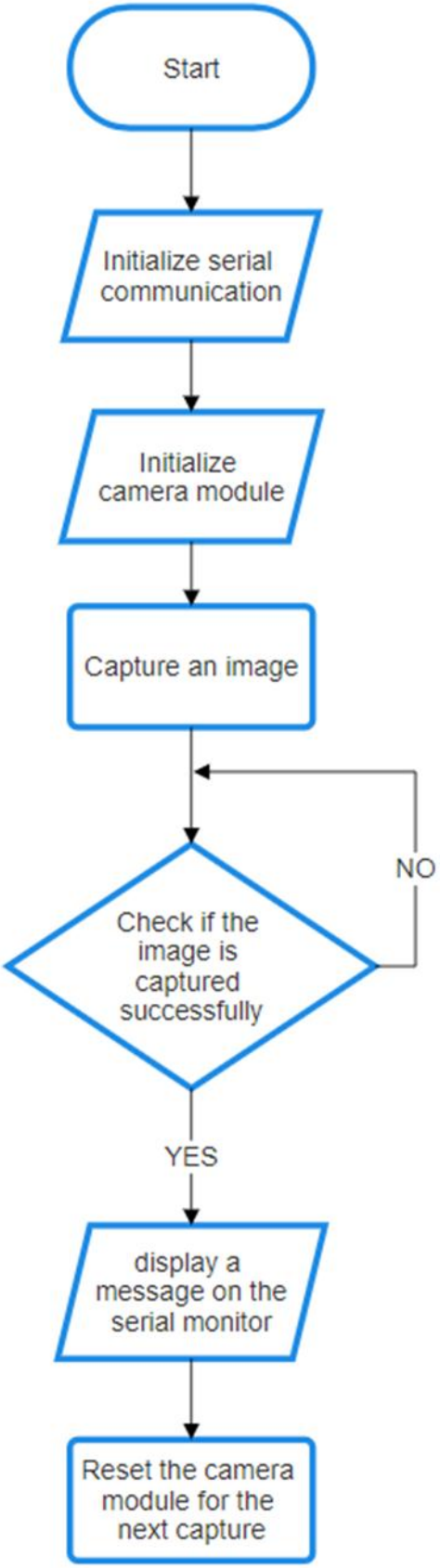
XI. Sample program

Step 1: Algorithm

Initialize serial communication

1. Initialize camera module
2. Capture an image
3. Check if the image is captured successfully
4. If true, display a message on the serial monitor
5. Reset the camera module for the next capture

Step 2: Flowchart


Start
Initialize serial communication
Initialize camera module
Capture an image
Check if the image is captured successfully
NO
YES
display a message on the serial monitor
Reset the camera module for the next capture

Step 3: Program

```
#include <CameraLibrary.h>

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Initialize camera module
  Camera.begin();
}

void loop() {
  // Capture an image
  Camera.capture();

  // Check if the image is captured successfully
  if (Camera.imageReady()) {
    // Display a message on the serial monitor
    Serial.println("Image captured!");

    // Process the captured image if needed
    // Example: Camera.processImage();

    // Reset the camera module for the next capture
    Camera.reset();
  }
}
```

XII. Results (Output of the Program)

……

……

……

……

XIII. Conclusion

……

……

……

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Build any one application using camera module.

……..……

……

……..

……..…

XV. References/Suggestions for further reading

1. Learn Circuits (tinkercad.com)

XVI. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 28: Create Web based IoT application using Node MCU/ESP32/Raspberry Pi to display Temperature on Web Browser

I. Practical Significance

To Create Web based IoT application using Node MCU/Raspberry Pi to display Temperature on Web Browser

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO4 - Develop IoT applications using IoT networking devices.

IV. Laboratory Learning Outcome(s)

LLO 28.1 Connect temperature sensor with embedded board.
LLO 28.2 Write program to display Temperature on Web Browser.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI. Relevant Theoretical Background

Creating a web-based IoT application to display temperature involves integrating various technologies and components. NodeMCU, ESP32 or Raspberry Pi can be used to read temperature data from sensors and send it to a web server. The data is then processed and stored, allowing it to be displayed in real-time on a web browser. This setup requires understanding IoT concepts, communication protocols, web development, and data storage solutions.

VII. Actual Circuit diagram used in laboratory with related equipment rating

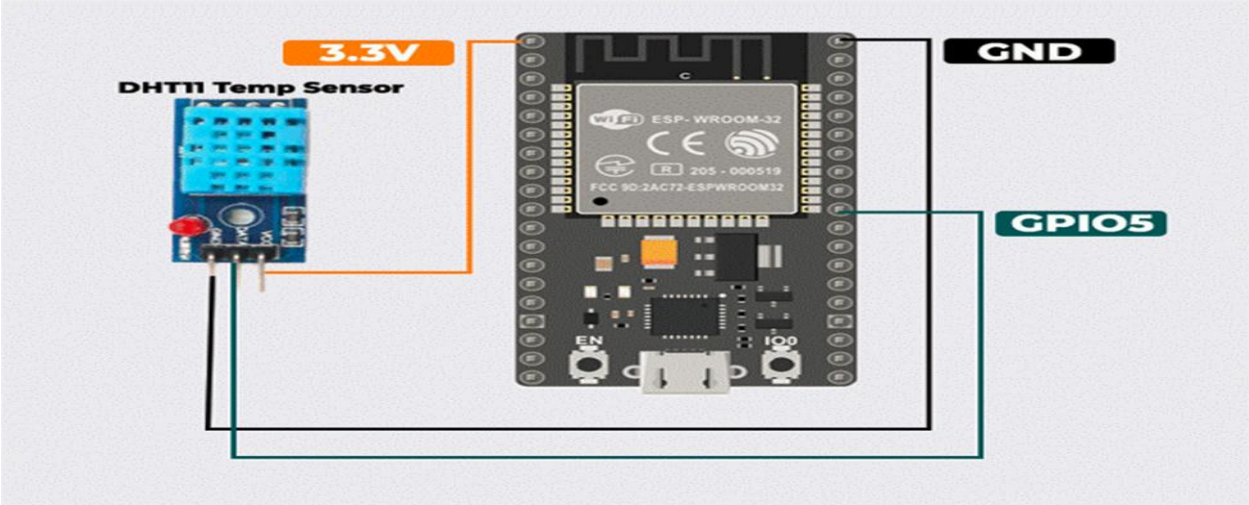


Fig 28.1 Circuit diagram to interface Temperature sensor with NODE MCU

VIII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qty
1	Node MCU		01
2	Temperature Sensor		01

IX .Precautions to be followed

Connect as per circuit diagram

X .Procedure

1. Interface temperature sensor to ESP32 as per circuit diagram shown in fig 28.1
2. Write algorithm for given problem.
3. Draw flowchart for the same.
4. Develop program using Arduino IDE software or any other relevant software tool.
5. Compile program on IDE.
6. Upload the program
7. Observe the output.

Getting Started with ThingSpeak Cloud Platform

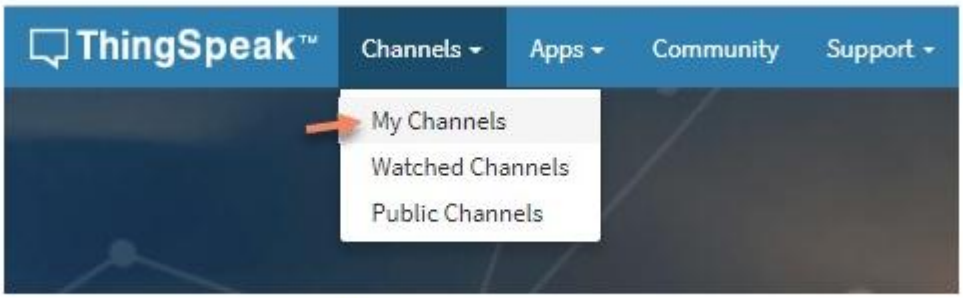
Step 1: Register and Login to ThingSpeak

Visit the ThingSpeak website (www.thingspeak.com) to create an account if you don't have one already.

Use your credentials to log in to your newly created ThingSpeak account.

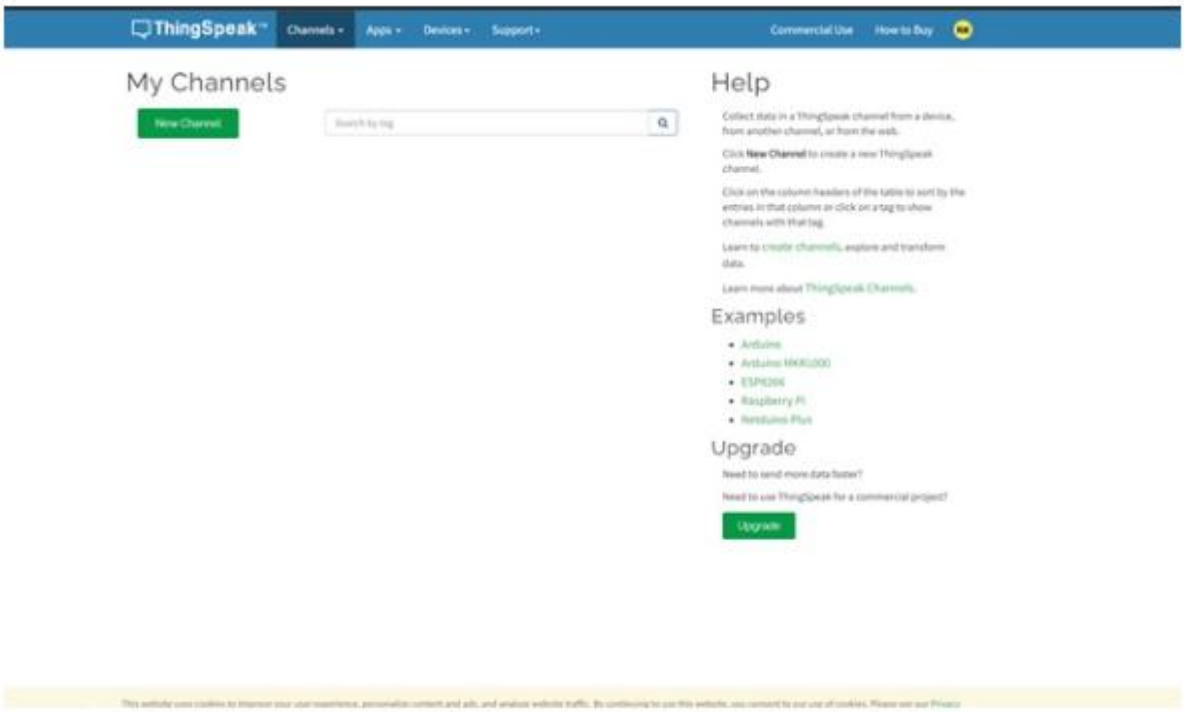
Step 2: Create a New Channel

After logging in, navigate to "Channels" in the top menu, then select "My Channels."



Thingspeak channels

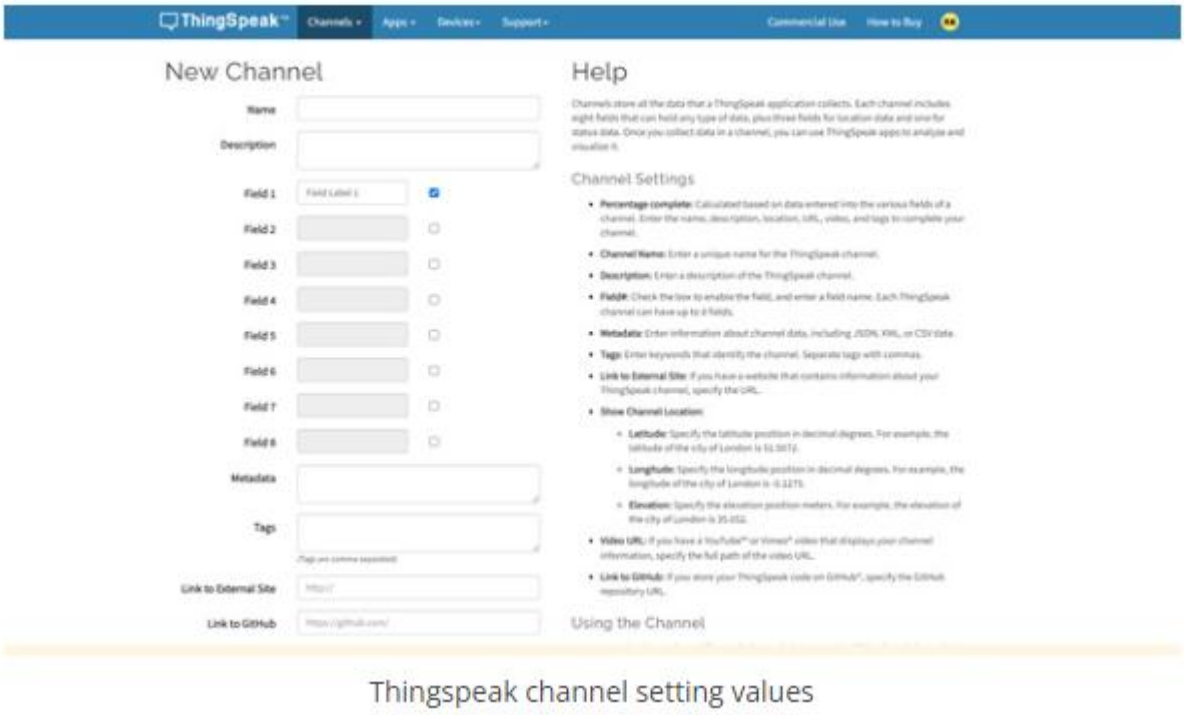
click the "New Channel" button, which initiates the channel creation process.



Thingspeak channel creation process

Provide essential details for your channel, such as its name, description, and the names of the fields representing the collected data.

Field 1: Temperature
Field 2: Humidity

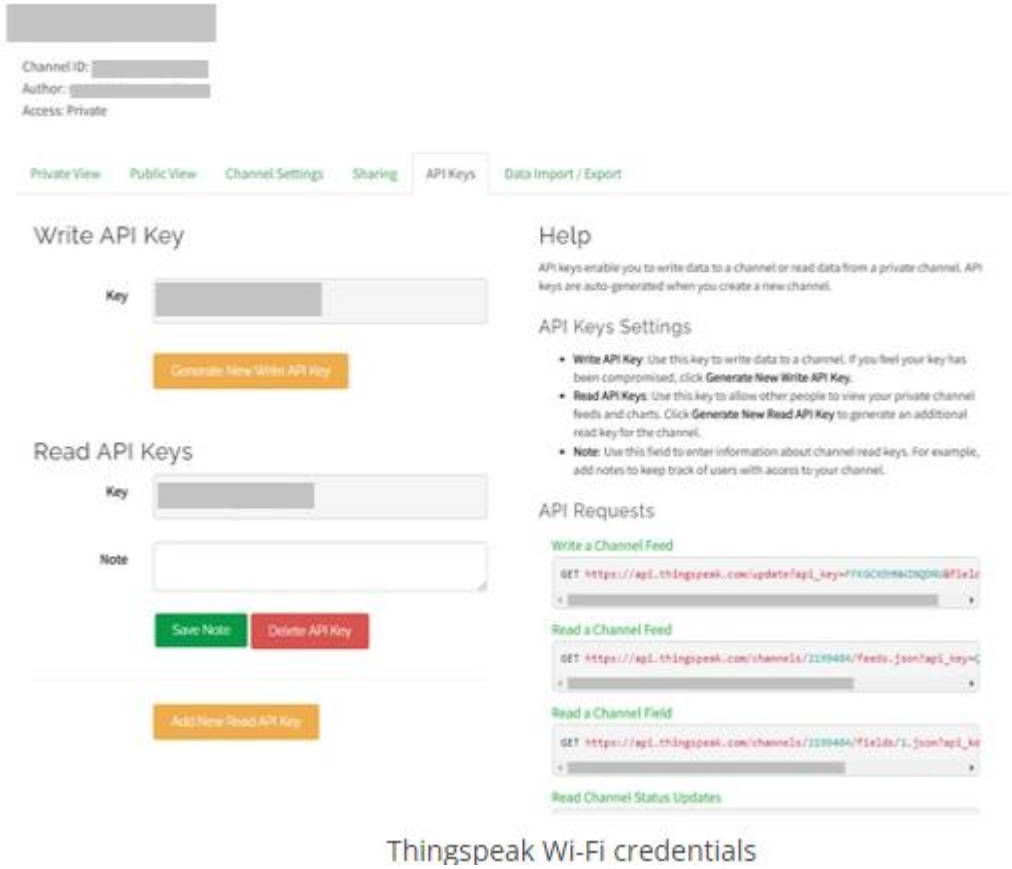


Thingspeak channel setting values

Step 3: Linking ESP32 with ThingSpeak

Set up your ESP32 board within the Arduino IDE. For beginners, tutorials on establishing the development environment are readily available.
Install the relevant libraries for ESP32 and ThingSpeak via the Arduino Library Manager.

Develop your Arduino sketch for the ESP32, including library imports and your Wi-Fi credentials.



Thingspeak Wi-Fi credentials

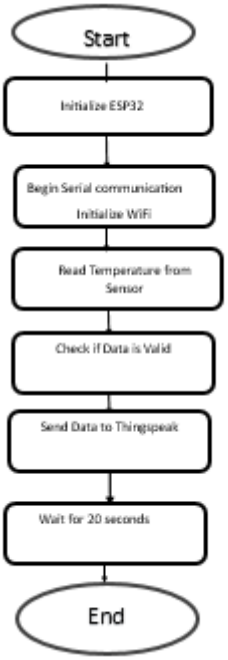
Employ the writeField function to transmit data from the ESP32's sensors to ThingSpeak, updating the defined fields in your channel.

XI. Sample program

Step 1: Algorithm

1. Start
2. Initialize ESP32 -- Begin Serial , Initialize WiFi , Connect to WiFI
3. Initialize Sensor -- Begin Sensor
4. Read Temperature from Sensor
6. Check if Data is Valid
7. Send Data to Thingspeak
8. Wait for 20 seconds
9. Repeat Loop
10. Crete account on Thing speak cloud
11. Copy API key and paste in Arduino code
12. Test data on cloud

Step 2: Flowchart



Step 3: Program

```
#include <DHT.h>
#include <WiFi.h>
#include <ThingSpeak.h>

// Replace with your network credentials
const char* ssid = "your_SSID"; // ex : aiskool
const char* password = "your_PASSWORD";// ex : aiskool

// ThingSpeak Settings
unsigned long myChannelNumber = YOUR_CHANNEL_NUMBER;
const char* myWriteAPIKey = "YOUR_WRITE_API_KEY";

// DHT Sensor Settings
#define DHTPIN 2      // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22 // DHT 22 (AM2302)

// Initialize DHT sensor
DHT dht(DHTPIN, DHTTYPE);

WiFiClient client;

void setup() {
  // Initialize serial
  Serial.begin(115200);
  delay(10);

  // Initialize DHT sensor
  dht.begin();

  // Connect to WiFi
  Serial.println("Connecting to WiFi");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting...");
  }
  Serial.println("Connected to WiFi");

  // Initialize ThingSpeak
  ThingSpeak.begin(client);
}

void loop() {
  // Read temperature and humidity
  float humidity = dht.readHumidity();
```

```
  float temperature = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Print values to serial monitor
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" *C");

  // Write to ThingSpeak
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);

  // Execute the write operation
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  // Wait 20 seconds to update the channel again
  delay(20000);
}
```

XII. Results (Output of the Program)

……..……

……

……..

……..…

XIII. Conclusion

……..……

……

……..

……..…

XIV. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO

1. Develop an application to interface ultrasonic sensor with nodeMCU to display distance on web browser

……..……

……

……..

……..…

……

XV. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. A Hands-On Approach Textbook Series | Internet of Things (hands-on-books-series.com)

XVI .Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 29: * Setup Raspberry Pi as a Single board computer with following accessories: display, cable to connect Raspberry Pi to display a keyboard a mouse SD card

I. Practical Significance

To Setup Raspberry Pi as a Single board computer

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

III. Course Level Learning Outcome(s)

CO5 - Develop database based IoT application by integrating sensors with single board computer.

IV. Laboratory Learning Outcome(s)

LLO 29.1 Install appropriate OS for embedded board.

LLO 29.2 Connect various accessories to Raspberry Pi.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices

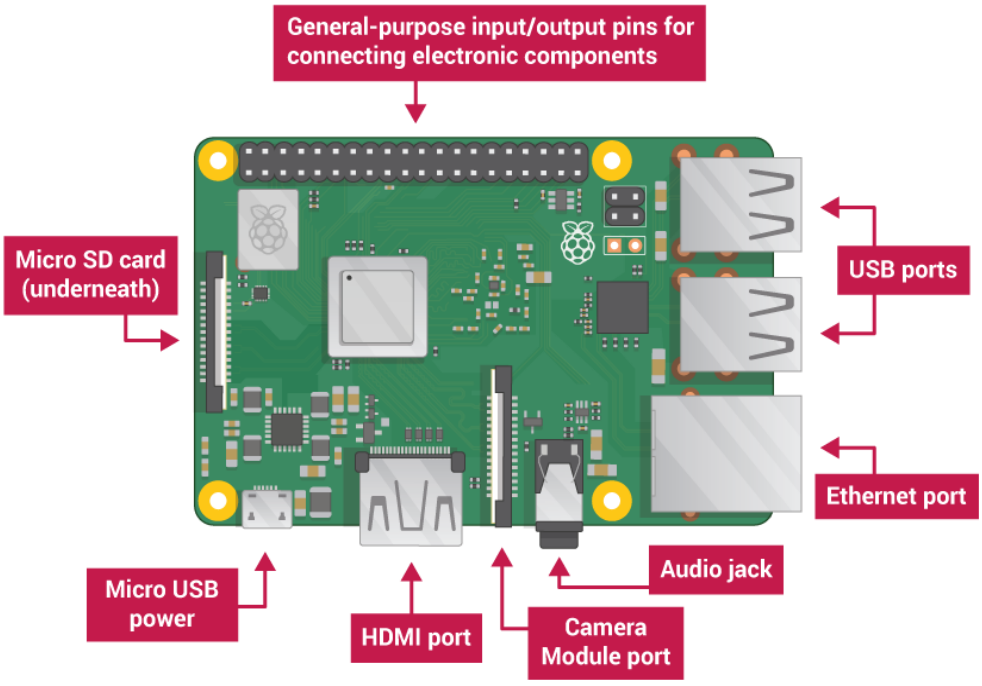
b. Maintain tools and equipment.

c. Follow ethical practices.

VI. Relevant Theoretical Background

The Raspberry Pi is a mini computer that was specifically created to make tech learning easier. It has a lot of components for computer-based projects, like USB ports, an Ethernet port, an SD card slot, Wi-Fi antenna ports, and more.

VII. Actual Circuit diagram used in laboratory with related equipment rating



1. **USB ports** - these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.

2. **SD card slot** - you can slot the SD card in here. This is where the operating system software and your files are stored.

3. **Ethernet port** - this is used to connect the Raspberry Pi to a network with a cable. The Raspberry Pi can also connect to a network via wireless LAN.

4. **Audio jack** - you can connect headphones or speakers here.

5. **HDMI port**- this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.

6. **Micro USB power connector** - this is where you connect a power supply. You should always do this last, after you have connected all your other components.

7. **GPIO ports** -these allow you to connect electronic components such as LEDs and buttons to the Raspberry Pi.

VIII. Required Resources

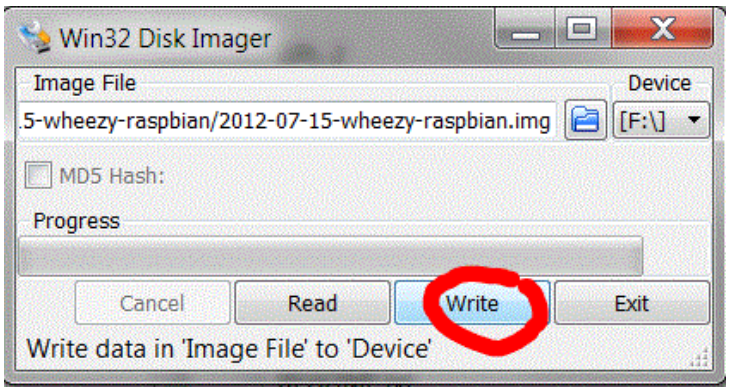
Sr. No.	Name of the Resources	Specifications	Quantity
1	Raspberry Pi		1
2	Power Supply		1
3	USB Keyboard		1
4	USB Mouse		1
5	Micro SD card		1
6	Micro SD USB card reader		1
7	A Monitor that supports HDMI		1
8	An HDMI cable		1
9	An Ethernet cable		1

IX .Precautions to be followed

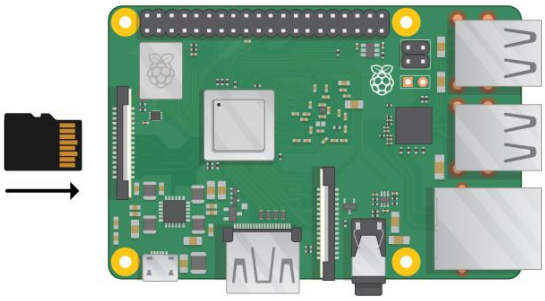
Handle all components carefully.

X .Procedure

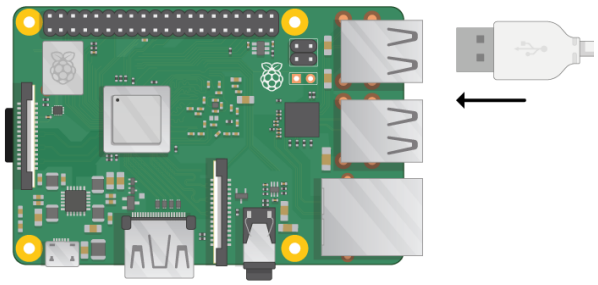
1. Insert the SD card into your SD card reader..
2. Run the Win32DiskImager utility from desktop or menu.
3. Select the image file and click on Write Button



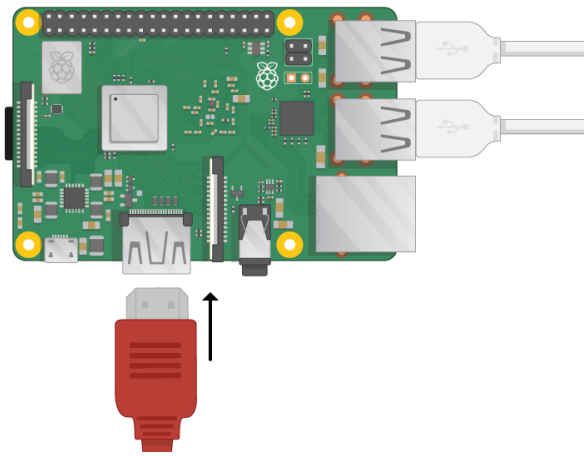
4. Remove SD Card & Insert in Raspberry Pi



5. Connect the mouse & Keyboard to USB port of Raspberry Pi.

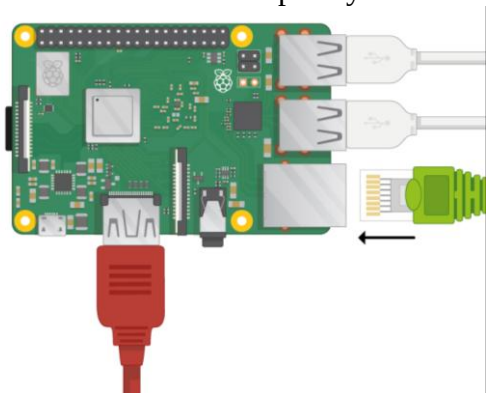


6. Connect Raspberry Pi to HDMI port directly or use HDMI to VGA connector.

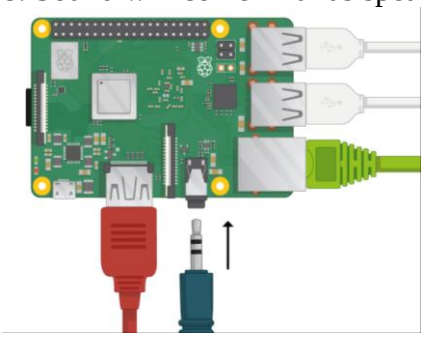


7. Connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the

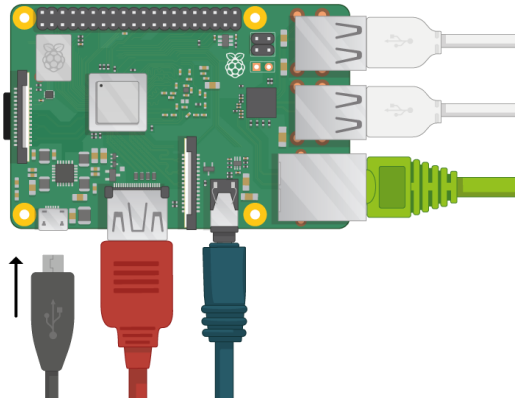
Raspberry Pi.



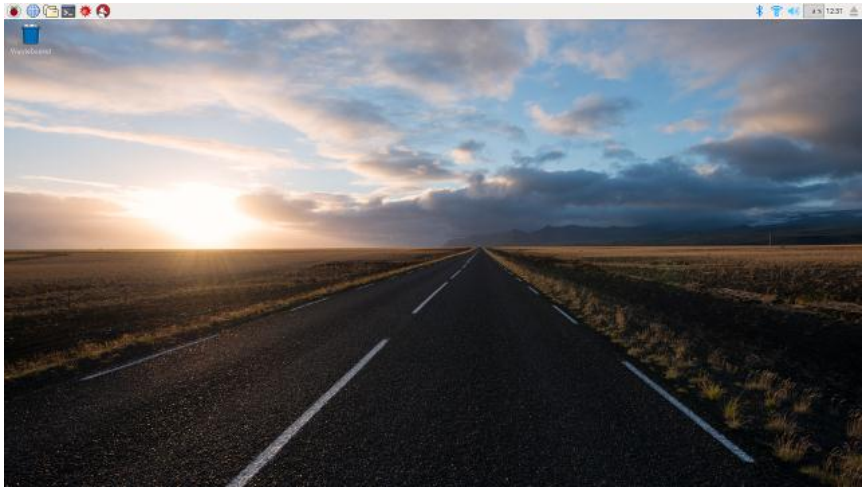
8. Sound will come if it has speakers or connect headphones or speakers to Audio jack if necessary



9. Plug the power supply into a socket and connect it to the micro USB power port

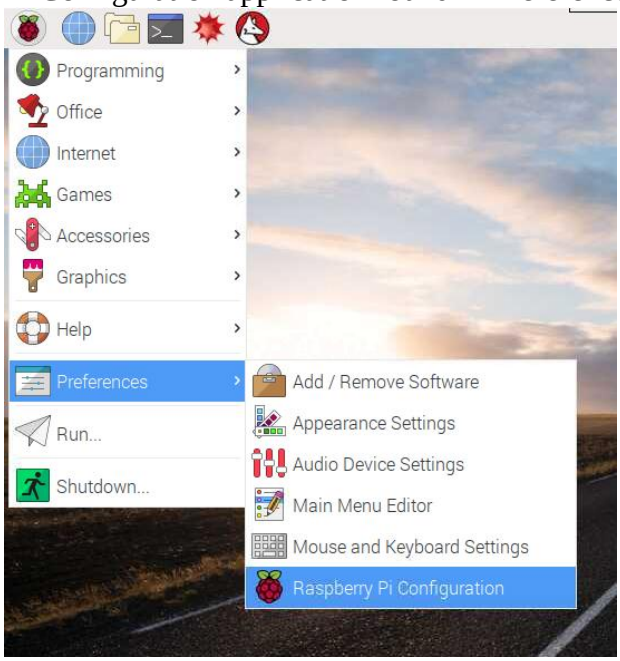


10. Red light will indicate Power On and Green Will Indicate the booting of the Raspberry Pi. The Pi will boot up into a graphical desktop.



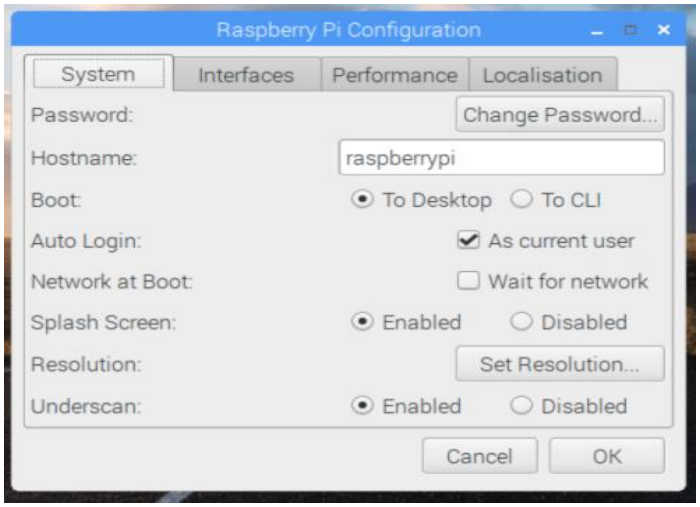
11. Configuring your Pi

1. You can control most of your Raspberry Pi's settings, such as the password, through the Raspberry Pi Configuration application found in Preferences on the menu.



II. System

In this tab you can change basic system settings of your Pi.



Password - set the password of the pi user (it is a good idea to change the password from the factory default 'raspberry')

Boot - select to show the Desktop or CLI (command line interface) when your Raspberry Pi starts

Auto Login- enabling this option will make the Raspberry Pi automatically log in whenever it starts

Network at Boot-selecting this option will cause your Raspberry Pi to wait until a network connection is available before starting

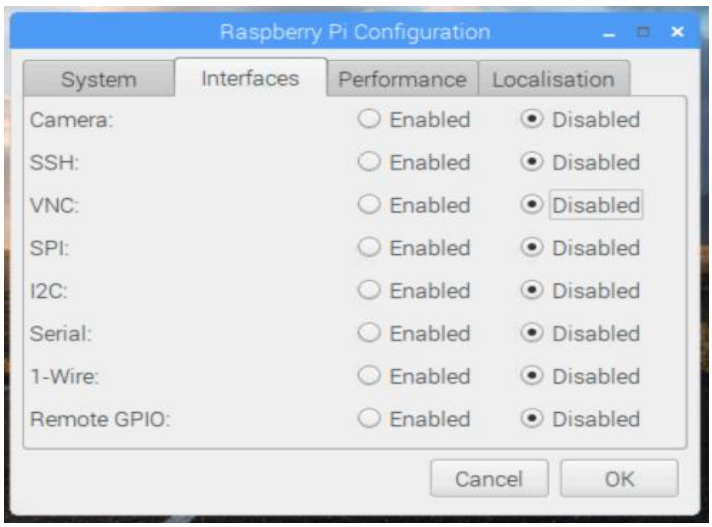
Splash Screen -choose whether or not to show the splash (startup) screen when your Raspberry Pi boots

Resolution - you can set the screen resolution here

Underscan - choose whether your Pi should show black bars at the top and bottom of the screen when it can't match the screen resolution

III. Interfaces

You can link devices and components to the Raspberry Pi using a lot of different types of connections. The Interfaces tab is where you turn these different connections on or off, so that the Pi recognizes that you've linked something to it via a particular type of connection.



Camera - enable the Raspberry Pi Camera Module

SSH - allow remote access to your Raspberry Pi from another computer using SSH

VNC - allow remote access to the Raspberry Pi Desktop from another computer using VNC

SPI -enable the SPI GPIO pins

I2C - enable the I2C GPIO pins

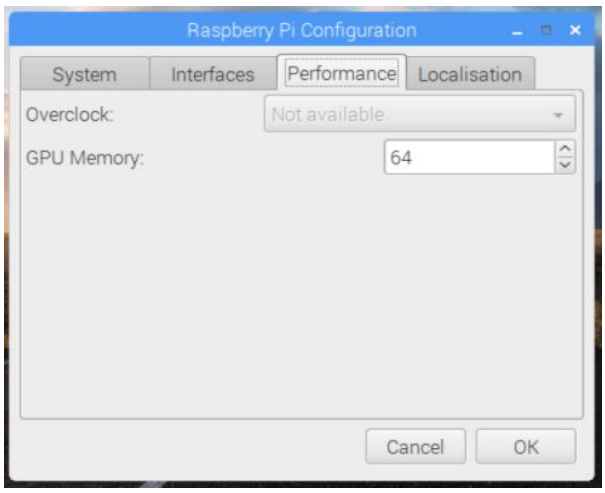
Serial -enable the Serial (Rx, Tx) GPIO pins

1-Wire - enable the 1-Wire GPIO pin

Remote GPIO - allow access your Raspberry Pi's GPIO pins from another computer

IV. Performance

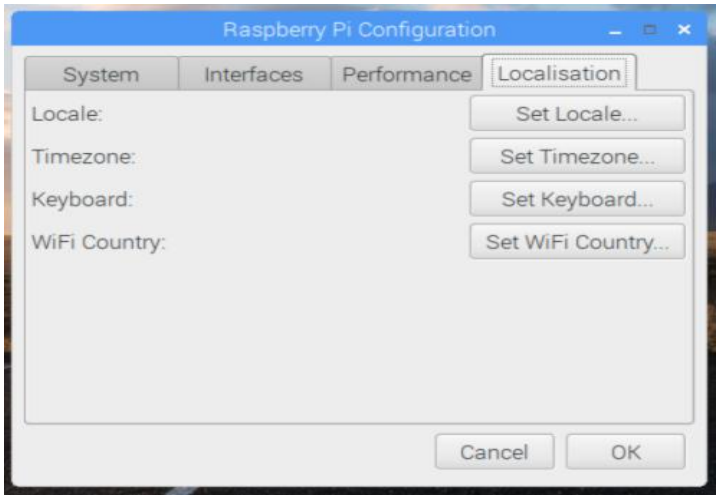
If you need to do so for a particular project you want to work on, you can change the performance settings of your Pi in this tab.



Overclock - change the CPU speed and voltage to increase performance

GPU Memory - change the allocation of memory given to the GPU

V. Localisation



This tab allows you to change your Raspberry Pi settings to be specific to a country or location.

Locale -set the language, country, and character set used by your Raspberry Pi

Timezone- set the time zone

Keyboard - change your keyboard layout

Wi-Fi Country - set the Wi-Fi country code

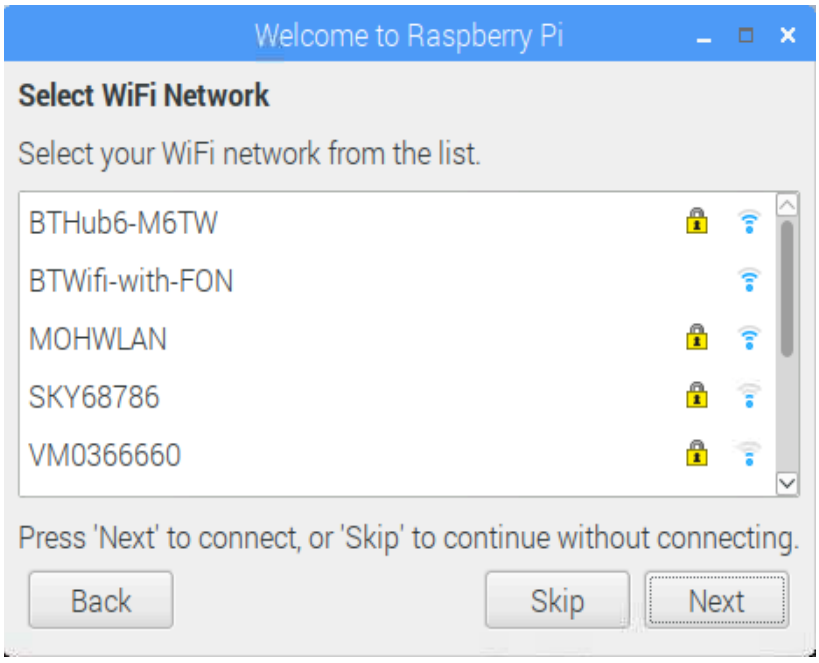
14. After starting with Raspberry Pi for the first time, the Welcome to Raspberry Pi application will pop up and it will guide through the initial setup.

15. Click Next to start the setup.

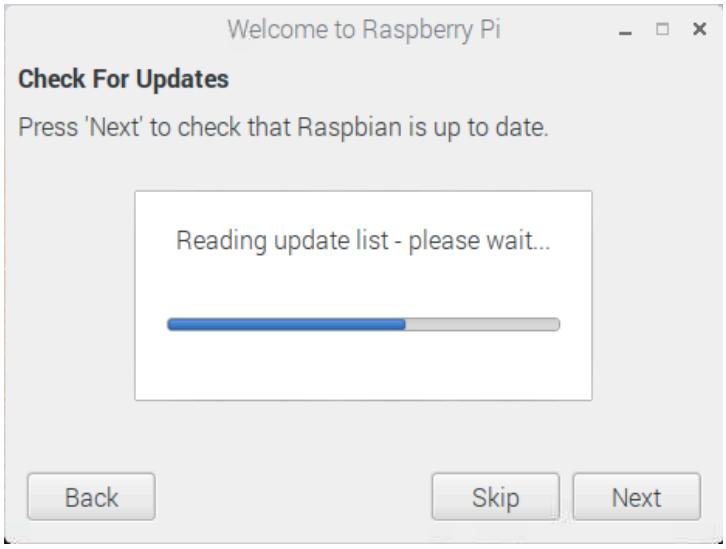
16. Set respective Country, Language, and Time zone, then click Next again.

17. Enter a new password for Raspberry Pi and click next

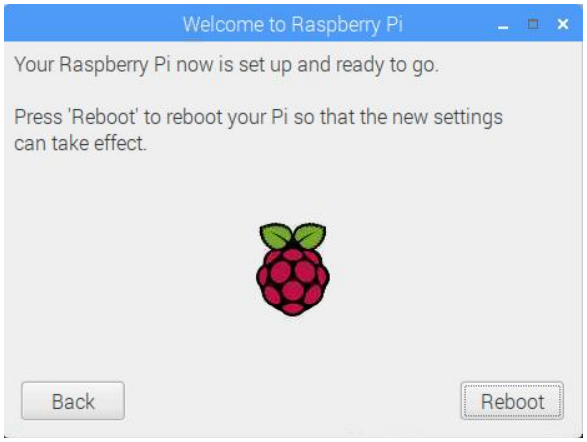
18. Connect to your Wi-Fi network by selecting its name, entering the password, and clicking next



19. Click next let the wizard check for updates to Raspbian and install.



20. Click done or Reboot to finish the setup.



XI. Results

……

……

……

……

XII. Conclusion

……

……

……

……

XIII. References/Suggestions for further reading

1. Arshdeep Bahga, Vijay Madisetti: Internet of Things: A Hands-On Approach
2. Simen Monk Raspberry Pi Cookbook Publisher(s): O'Reilly Media, Inc. ISBN: 9781098130923

XIV. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	

Practical No. 30: Install MariaDB database in Raspberry Pi and execute basic SQL queries

I. Practical Significance

To Install MariaDB database in Raspberry Pi and execute basic SQL queries

II. Industry / Employer Expected Outcome(s)

Develop and implement creative solutions for real time problems that can enhance efficiency, safety and convenience across various domains

.

III. Course Level Learning Outcome(s)

CO5 -Develop database based IoT application by integrating sensors with single board computer.

IV. Laboratory Learning Outcome(s)

LLO 30.1 Install database on single board computer.
LLO 30.2 Perform various queries for displaying desired result.

V. Relevant Affective Domain related Outcomes

a. Follow safe practices
b. Maintain tools and equipment.
c. Follow ethical practices.

VI.Relevant Theoretical Background

In this experiment we will see how to install MariaDB on Raspberry Pi..MariaDB is a database management system derived from MySQL with a GPL (General Public License). It is developed by Michael Widenius, one of the founders of MySQL, the MariaDB foundation and the open source software development community.

VII. Required Resources

Sr. No.	Name of the Resources	Specifications	Qty
1	Raspberry PI		01
2	microSD card		01

VIII. Procedure

1. Install MariaDB

Installing MariaDB on Raspberry Pi is just as simple as with MySQL. In fact, it is practically identical. First, we update packages with following commands on command prompt

1. **sudo apt update**

Next, we install MariaDB by running the following command,

2. **sudo apt install mariadb-server**

As we can see, exactly the same as in the MySQL tutorial but changing ‘mysql-server’ to ‘mariadb-server’.

Just like in MySQL, the “factory” configuration is insecure because we should not leave the default parameters. Therefore, the immediate next step is to run the script.

3. sudo mysql_secure_installation

We answer the questions about users and passwords, and now the installation is ready.

Finally, we check that the service is running correctly with the following command,

4. systemctl status mariadb.service

And that’s how easy it was to install MariaDB. You can use MySQL/MariaDB interchangeably

IX. Conclusion

……

……

……

……

X. Practical related questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified

1. Explain how to install MySQL on raspberry pi

……

……

……

……

XI. References/Suggestions for further reading

1.https://www.luisllamas.es/en/how-to-install-mariadb-on-raspberry-pi/

XII. Assessment Scheme (25 Marks)

S. No.	Weightage- Process related: 60%	Marks-15
1.	Coding and Debugging ability	
2.	Making connections of hardware	
3.	Follow ethical practices.	
	Weightage- Product related: 40%	**Marks-10**
4.	Relevance of output of the problem definition	
5.	Timely Submission of report	
	Total 25	
	Dated Signature of Course Teacher	