Go, Change the world®

RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi

Academic Year 2023-24

IDEA Lab

(Idea Development, Evaluation and Application Lab)

(ME111AL/ME121AL)

BE PROGRAMS

(Common to All Branches)

Name of the Student _____

USN_____Semester_____

DEPARTMENT OF MECHANICAL ENGINEERING

Dr. M Krishna
&
Dr. Roopa T S

Dr. Kendaganna Swamy S
&
Dr. Rajasree P M

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to VTU, Belagavi)

R.V. Vidyaniketan Post, Mysore Road, Bengaluru – 560 059

IDEA Lab (ME111AL/ME121AL)
BE Programs
Common to All Branches

Certificate

This is to certify that Mr./Ms..... USN .....of …… Semester BE program ……of RV College of Engineering has attained predefined practical outcomes (PROs) satisfactorily in the course Idea Lab for the academic year ……. as prescribed in the curriculum

Faculty Incharge Head of the Department

VISION

Quality education in Design, Materials, Thermal and Manufacturing with emphasis on research, sustainable technologies, and entrepreneurship for societal symbiosis.

MISSION

1. Imparting knowledge in basic and applied areas of Mechanical Engineering.
2. Providing state-of-the-art laboratories and infrastructure for academics and research in the areas of design, materials, thermal engineering, and manufacturing.
3. Facilitating faculty development through continuous improvement programs.
4. Promoting research, education and training in materials, design, manufacturing, Thermal Engineering, and other multidisciplinary areas.
5. Strengthening collaboration with industries, research organizations and institutes for internship, joint research, and consultancy
6. Imbibing social and ethical values in students, staff, and faculty through personality development programs.

Course Outcomes: After going through this course, the student will be able to:	
CO1:	Hands-on-experience to interface various Digital and Analog Sensors with Arduino
CO2:	Assembling Mechanical and Electronic Components to analyze the functioning of Actuators
CO3:	Exploring Knowledge about Connecting Sensors and devises to various IoT platforms
CO4:	Improved coding experience

Practical- Course Outcome matrix

SI. No.	Practical Outcome	CO1	CO2	CO3	CO4
Introductory Experiments					
1.	a) LED On/Off Multiple LED On/Off	√		√	√
2.	LED with switch Interface	√		√	√
YELLOW KIT - PART A					
1.	Ultrasonic Sensor with LED and buzzer Interface ** (DC pump)	√	√	√	√
2.	Temperature & Humidity Sensor with LCD interfacing ** (DC Motor)	√	√	√	√
3.	Flame Sensor interfacing with buzzer and LED ** (DC motor/Pump)	√	√	√	√
4.	Float Sensor interfacing with buzzer and LED ** (DC Pump)	√	√	√	√
5.	Interfacing Heartbeat Sensor with Arduino UNO	√	√	√	√
RED KIT- PART B					
6.	Touch Sensor interfacing with LED and solenoid lock *	√	√	√	√
7.	Interfacing Soil Moisture Sensor with Arduino UNO ** (DC Pump)	√		√	√
8.	Interfacing LDR sensor with Arduino UNO to switch on/off LED or 230V bulb using a 5V Relay	√		√	√
9.	Smoke Sensor interfacing with LED and buzzer	√		√	√
10.	Interfacing IR sensor with Arduino and servo motor	√	√	√	√
Demonstration Experiments					
1.	Interfacing Magnetic Door Sensor with LED and buzzer	√		√	√
2.	Robot Demonstration		√		
3.	Experiential Learning	√	√	√	√
4.	CIE Lab	√	√	√	√
5.	Exhibition	√	√	√	√
** Part of EL component					

Guidelines to Teachers

1. Teacher needs to ensure that a dated logbook for the whole semester, apart from the laboratory manual is maintained by every student which she/he has to submit for assessment to the teacher in the next practical session.
2. There will be two sheets of blank pages after every practical for the student to report other matters (if any), which is not mentioned in the printed practicals.
3. For difficult practical if required, teacher could provide the demonstration of the practical emphasizing of the skills which the student should achieve.
4. Teachers should give opportunity to students for hands-on after the demonstration.
5. Assess the skill achievement of the students and COs of each unit.
6. One or two questions are to be added in each practical for different batches. For this teacher can maintain various practical related question bank.
7. If some repetitive information like data sheet, use of software tools etc. has to be provided for effective attainment of practical outcomes, they can be incorporated in Appendix.
8. For effective implementation and attainment of practical outcomes, teachers ought to ensure that at the beginning of each practical, students must read through the complete write-up of that practical sheet.
9. During practical, ensure that each student gets chance and takes active part in taking observations/ readings and performing practical.
10. Teacher ought to assess the performance of students continuously according to the guidelines.

Instructions for Students

1. For incidental writing on the day of each practical session every student should maintain a dated logbook for the whole semester, apart from this laboratory manual which she/he has to submit for assessment to the teacher in the next practical session.
2. For effective implementation and attainment of practical outcomes, in the beginning itself of each practical, students need to read through the complete write-up including the practical related questions and assessment scheme of that practical sheet.
3. Students should not hesitate to ask about any difficulties they face during the conduct of practical.

Expt. No.	Rubrics			Total Marks for 10
	Clear effective communication and quality of report presentation (03)	Demonstration of process and understanding of the IoT technique (05)	Effective achievement of report Goal (02)	
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				
Total Marks of Lab Experiments (50)				
Lab Test (50)				
Experiential Learning (10)				
Total Marks (50)				

Conduction of laboratory exercises, Lab report & observation & analysis (50 Marks), Lab Test (50 Marks) both will be added and will be reduced to 40 marks & Innovative Experiment/Concept Design & Implementation (10 Marks) adding up to 50 Marks.

50

Signature of the Student **Signature of the Faculty**

1. Pre-requisite: Arduino

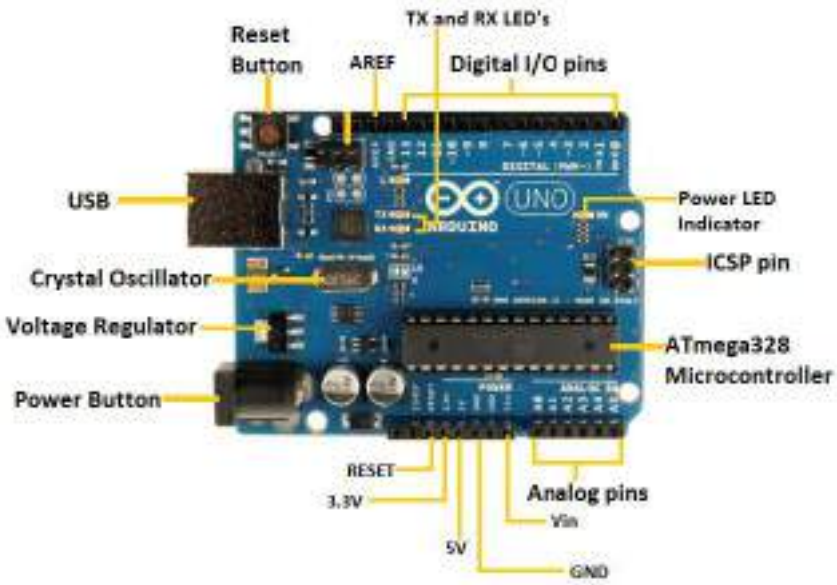

Fig. 1: Arduino Uno Board

1.1 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

1.2 Power USB

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection. Arduino boards can be powered directly from the AC mains power supply by connecting it to the Jack of Power button.

1.3 Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

1.4 Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz

1.5 Arduino Reset

It can reset Arduino board, i.e., start program from the beginning. It can be reset the UNO board in two ways. First, by using the reset button on the board. Second, connect an external reset button to the Arduino pin labelled RESET.

1.6 AREF

AREF stands for Analog Reference. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

1.7 TX and RX LEDs

Board can be labeled in two ways: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led. The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

1.8 Digital I/O

The Arduino UNO board has 14 digital I/O pins (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.

1.9 Power LED indicator

This LED should light up when plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

1.10 ICSP pin

Mostly, ICSP is a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, V_{CC}, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

1.11 Analog pins

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

1.12 Pins (3.3, 5, GND, Vin)

- ✓ 3.3V (6) − Supply 3.3 output volt
- ✓ 5V (7) − Supply 5 output volt
- ✓ Most of the components used with Arduino board works fine with 3.3 volt and 5 volts.
- ✓ GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- ✓ Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

1.13 Main microcontroller

Each Arduino board has its own microcontroller. You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE.

It offers some advantages over other systems:

- ✓ Inexpensive
- ✓ Cross-platform
- ✓ Simple, clear programming environment
- ✓ Open source and extensible software
- ✓ Open source and extensible hardware

1.14Arduino UNO board is used in the following applications.

- ✓ Weighing Machines.
- ✓ Traffic Light Count Down Timer.
- ✓ Parking Lot Counter.
- ✓ Embedded systems.
- ✓ Home Automation.
- ✓ Industrial Automation.
- ✓ Medical Instrument.
- ✓ Emergency Light for Railways.

1.15 Arduino IDE

Set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

 # Step 1 − In case you use Arduino UNO, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in Fig 2.

Fig. 2: Arduino-USB cable

 # Step 2 − Download Arduino IDE Software (Fig. 3). You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Fig. 3: Download Arduino IDE Software

 # Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

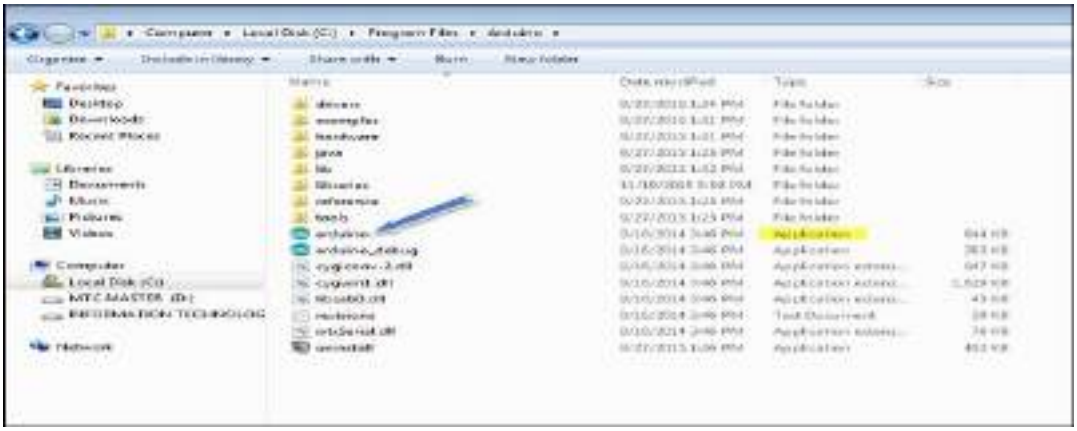

Fig. 4: Launch Arduino IDE

 # Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → New.

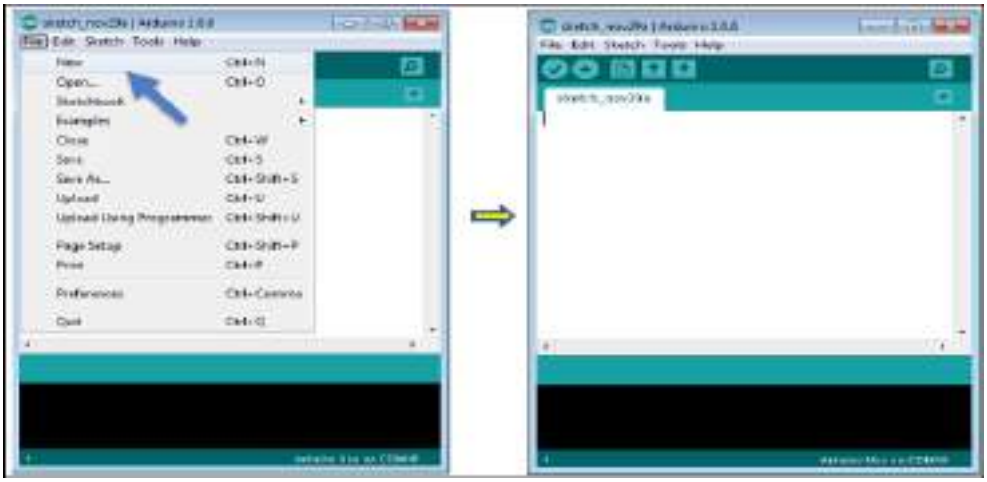

Fig. 5: Open your first project.

To open an existing project example, select File → Example → Basics → Blink.

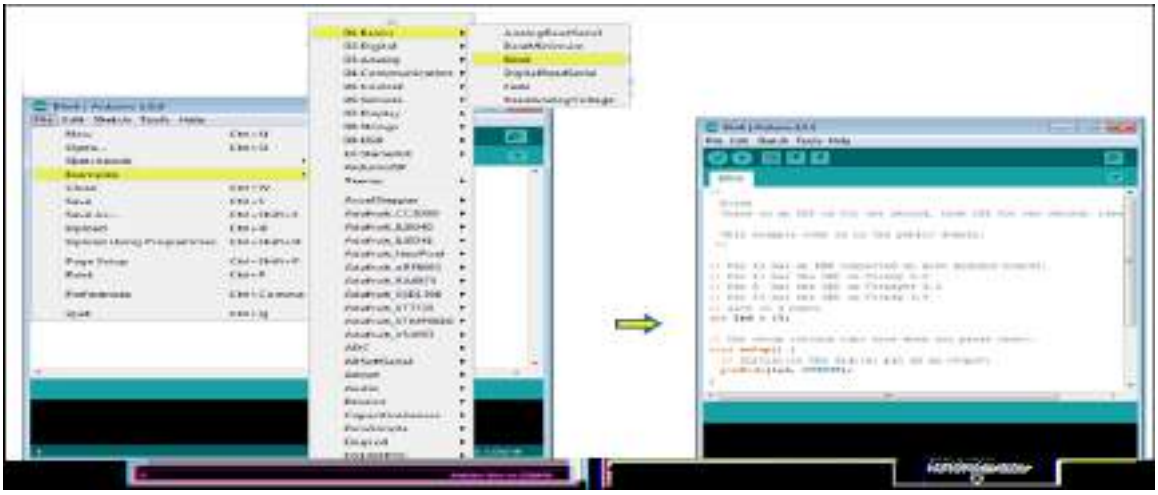

Fig. 6: To open an existing project

We are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

 # Step 6 − Select your Arduino board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

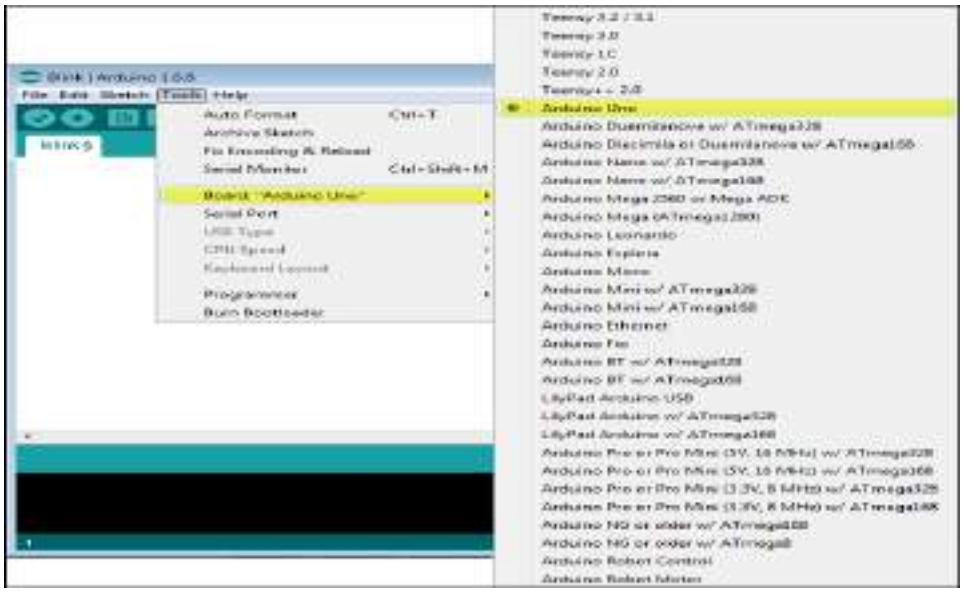

Fig. 7: Select your Arduino board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

 # Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

Fig. 8: Select your serial port.

 # Step 8 − Upload the program to your board. Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

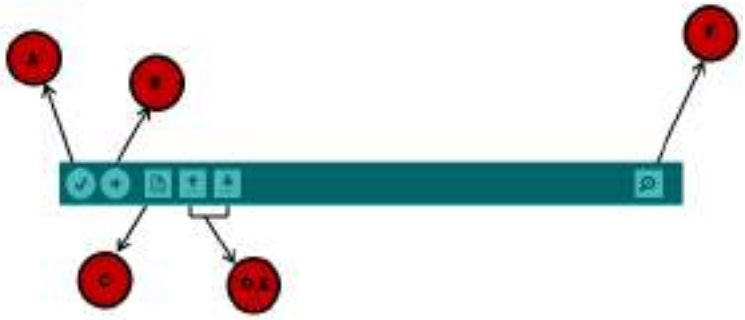

Fig. 9: Each Icon information on Arduino IDE

A − Used to check if there is any compilation error.
B − Used to upload a program to the Arduino board.
C − Shortcut used to create a new sketch.
D − Used to directly open one of the example sketches.
E − Used to save your sketch.
F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

1.16 Libraries

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the Liquid Crystal I2C library makes it easy to talk to character LCD displays. There are thousands of libraries available for download directly through the Arduino IDE, and you can find all of them listed at the Arduino Library Reference.

1.17 Installing Libraries

This library should be already downloaded to the Arduino IDE. Go to the Sketch menu >> Include Library >> Library Manager. In the top right text box, type in LiquidCrystal I2C then look for the LiquidCrystal I2C Library. Once you find the library click on it, select the latest version and hit install. Wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, an *Installed* tag should appear.

Fig. 10: Installing the libraries.

Downloading may take time depending on your connection speed. Once it has finished, an *Installed* tag should appear.

Fig. 11: After download, an Installed tag should appear.

2. Prerequisite: Sensors and Actuators

We live in a World of Sensors. You can find different types of Sensors in our homes, offices, cars etc. working to make our lives easier by turning on the lights by detecting our presence, adjusting the room temperature, detect smoke or fire, make us delicious coffee, open garage doors as soon as our car is near the door and many other tasks. All these and many other automation tasks are possible because of Sensors.

2.1 Sensors

A sensor is a device that measures physical input from its environment and converts it into data (electrical quantity) that can be interpreted by either a human or a machine.

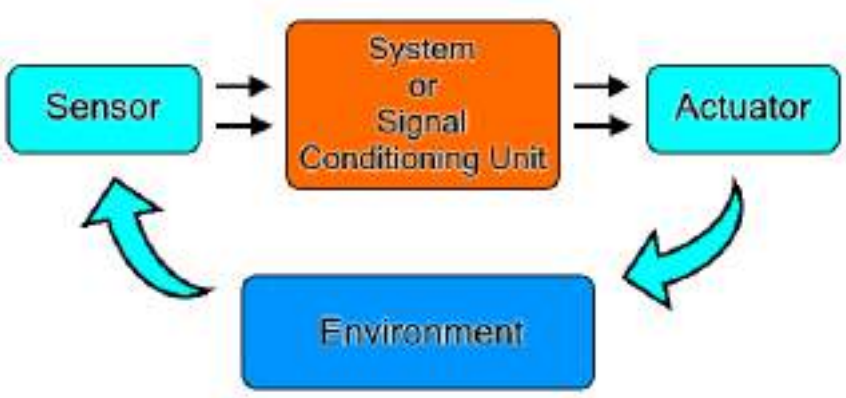

Fig. 12: Sensors and actuators block

2.2 Actuators:

An actuator is a part of a device or machine that helps it to achieve physical movements by converting energy, often electrical, air, or hydraulic, into mechanical force. Actuators are present in almost every machine around us, from simple electronic access control systems, the vibrator on your mobile phone and household appliances to vehicles, industrial devices, and robots. Common examples of actuators include electric motors, stepper motors, jackscrews, electric muscular stimulators in robots, etc.

2.3 Classification of Sensors

- ✓ In the first classification of the sensors, they are divided in to Active and Passive. Active Sensors are those which require an external excitation signal or a power signal. Passive Sensors, on the other hand, do not require any external power signal and directly generates output response.
- ✓ The other type of classification is based on the means of detection used in the sensor. Some of the means of detection are Electric, Biological, Chemical, and Radioactive etc.
- ✓ The next classification is based on conversion phenomenon i.e., the input and the output. Some of the common conversion phenomena are Photoelectric, Thermoelectric, Electrochemical, Electromagnetic, Thermo-optic, etc.
- ✓ The final classification of the sensors are Analog and Digital Sensors. Analog Sensors produce an analog output i.e., a continuous output signal (usually voltage but sometimes other quantities like Resistance etc.) with respect to the quantity being measured. Digital Sensors, in contrast to Analog Sensors, work with discrete or digital data. The data in digital sensors, which is used for conversion and transmission, is digital in nature.

2.4 Different Types of Sensors

The following is a list of different types of sensors that are commonly used in various applications. All these sensors are used for measuring one of the physical properties like Temperature, Resistance, Capacitance, Conduction, Heat Transfer etc.

1. Temperature Sensor
2. Proximity Sensor
3. Accelerometer
4. IR Sensor (Infrared Sensor)
5. Pressure Sensor
6. Light Sensor
7. Ultrasonic Sensor
8. Smoke, Gas and Alcohol Sensor
9. Touch Sensor
10. Humidity Sensor
11. Solar cell sensor
12. Metal detector sensor
13. Soil Moisture Sensor
14. Heartbeat Sensor
15. Microphone (Sound Sensor)
16. Tilt Sensor
17. Water Flow and Level Sensor
18. PIR (passive infrared) Sensor
19. Rain Sensor
20. Oil moisture Sensor
21. Real time clock Sensor
22. Vibration Sensor

Fig. 13: Various sensors.

Introductory Experiment No: 1 (a)

LED ON/OFF using Arduino.

AIM: To switch on the LED using Arduino

A Light Emitting Diode (LED) is a semiconductor device, which can emit light when an electric current passes through it. To do this, holes from p-type semiconductors recombine with electrons from n-type semiconductors to produce light. LEDs are comprised of compound semiconductor materials, which are made up of elements from group III and group V of the periodic table (these are known as III-V materials). Examples of III-V materials commonly used to make LEDs are gallium arsenide (GaAs) and gallium phosphide (GaP). LED, however, is noted for its long lifespan, energy efficiency, and low maintenance costs. An LED light can operate up to 50,000 to 100,000 operating hours or roughly five to 10 years. It is more energy-efficient than incandescent as it wastes less energy by producing a low amount of heat.

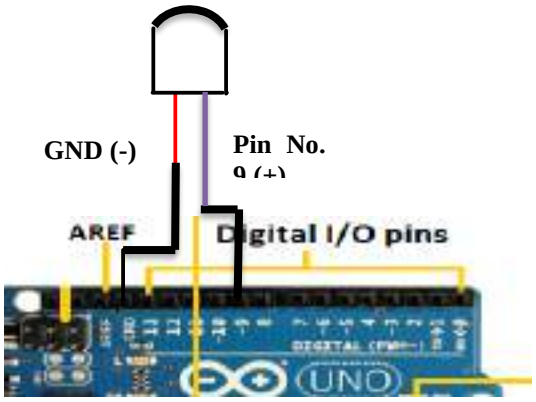

Fig. 1. Circuit Diagram**.**

Particulars
Arduino UNO:1
LED Pin-1
USB cable for uploading the code-1
Jumper wires -2

Program Code:

```
void setup () {
pinMode (9, OUTPUT); // define 9 as an output.
}
void loop () {
  digitalWrite (9, HIGH); //Turn On the LED
  delay (1000);
  digitalWrite (9, LOW); //Turn Off the LED
  delay (1000);
}
```

Procedure

- Verify the Program for errors.
- Connect Arduino to PC Via USB Cable
- Open Arduino IDE, Select the Arduino UNO board and Port.
- Write the Code using Arduino IDE Software.
- Initialize the Green LED Pin to Arduino Data Pin 9.
- Make the Connections as shown in the Circuit Diagram.
- Now upload the code to the Arduino.

Applications:

- Picture phones and digital watches.
- Camera flashes and automotive heat lamps.
- Aviation lighting.
- Digital computers and calculators.
- Traffic signals and Burglar alarms systems.
- Microprocessors and multiplexers.
- Picture phones and digital watches.
- Camera flashes and automotive heat lamps.
- Aviation lighting.
- Digital computers and calculators.
- Traffic signals and Burglar alarms systems.
- Microprocessors and multiplexers.

Results:

The Code to turn ON/OFF the LED has been written and executed successfully using Arduino UNO.

Date: **Signature of Faculty**

Introductory Experiment No: 1(b)

Multiple LEDs ON/Off using Arduino.

AIM: To switch on the multiple LEDs using Arduino

LEDs are small, powerful lights that are used in many different applications. To start, we will work on blinking multiple LEDs. It is as simple as turning a light on and off. Establishing this important baseline will give you a solid foundation as we work towards experiments that are more complex. To find out the polarity of an LED, look at it closely. The shorter of the two legs, towards the flat edge of the bulb indicates the negative terminal.

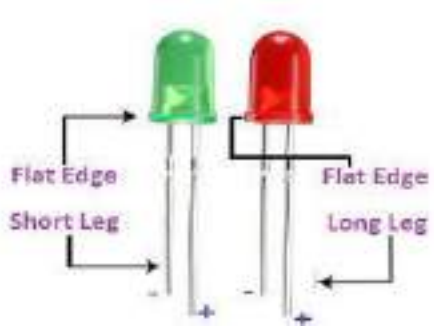

Fig. 2: LED showing the terminals.

Program Code:

```
void setup() {
  // put your setup code here, to run once:
pinMode (10,OUTPUT);
pinMode (11,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
digitalWrite (10, HIGH); // TURN ON Green LED
delay (1000); // delay of ms
digitalWrite (10, LOW); // TURN OFF Green LED
delay (500);
digitalWrite (11, HIGH); // TURN ON Red LED
delay (1000);
digitalWrite (11, LOW);  // TURN OFF Red LED
delay (500);
}
```

Particulars
Arduino UNO:1
LED Pin-1
USB cable for uploading the code-1
Breadboard -1

Procedure

- Open Arduino IDE, Select the Arduino UNO board and Port.
- Write the Code using Arduino IDE Software.
- Initialize the Green LED Pin and Red LED Pin to Arduino Data Pin 10 and 11 respectively.
- Make the Connections as shown in the Circuit Diagram.
- Now upload the code to the Arduino

Applications:

- Picture phones and digital watches.
- Camera flashes and automotive heat lamps.
- Aviation lighting.
- Digital computers and calculators.
- Traffic signals and Burglar alarms systems.
- Microprocessors and multiplexers.

Circuit Diagram:

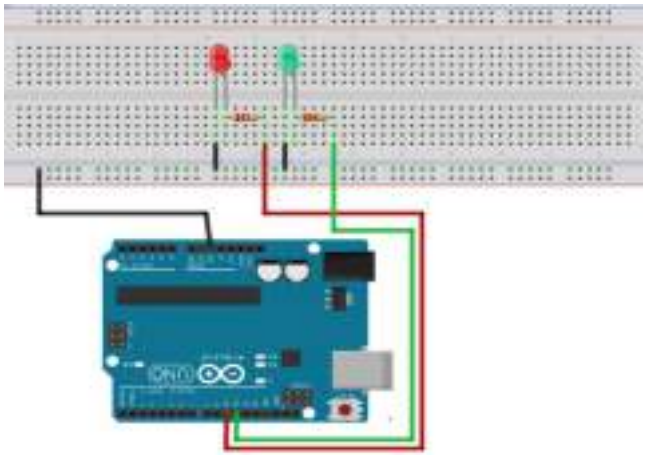

Fig. 3 Circuit diagram to switch the multiple LEDs

Result:

The Code to turn On/OFF the Green and Red LEDs alternatively has been written and executed successfully using Arduino UNO.

Date: **Signature of Faculty**

Introductory Experiment No: 2

LED with switch Interface

Aim: Interfacing Push Button to LED with Arduino UNO for Switch Operation of LED Light.

Theory: A LED is a two-terminal semiconductor light source. This simple P-N junction diode emits light when it gets activated and usually requires a small voltage for its operation. When voltage is applied, the electrons will recombine with the holes and will release energy in the form of light. A Push Button works like a switch, where when there's a pressure on the surface of a button, it opens up an electrical circuit and allows currents to flow through it.

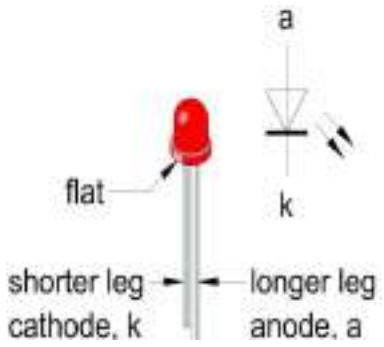

Fig. 4: Push-Button LED Interface

- Vcc – 5V on the Arduino Board
- Signal – Pinout on Arduino board (whichever has been defined)
- LED Gnd – Gnd on the Arduino board
- LED +ve – Pinout on Arduino board (whichever has been defined)

Here we are using a push button that servers as a toggle switch. You can read the working of a push button with Arduino UNO for a better understanding of the project.

When you press the push button the corresponding LED will turn on. For turning the LED off you have to press the button again. In the code, we use a status variable that initially holds a false value, and when you press the button, its value changes.

A push-button completes the circuit when you press it and breaks the circuit when released. The Arduino keeps tracking the state of the LED and then invert its state when you push the button. The complete project looks like this.

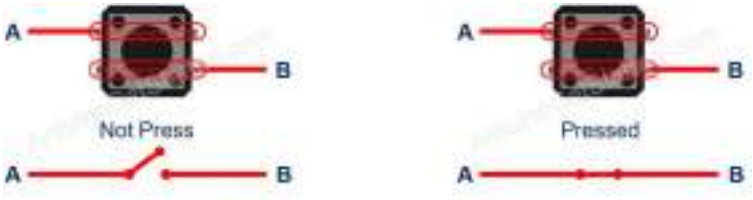

Fig. 5: Internal structure of push button

The Key Applications of Switch-LED Interface Technology

As mentioned in introduction, a Switch-LED Interface is used for a wide array of applications, with those being:

1. Push buttons are switches that are either concealed inside machinery or plugged in. In layman's terms, they can be seen and used. The push button switches consist mainly of two types – momentary and non-momentary. These switches are ordinarily used in calculators, push-button telephones, kitchen appliances, magnetic locks, and several other mechanical and electronic devices used across homes or industries. Smart phones, automotive, industrial applications

2. The extensive control yielded by buttons mandates a scope of operation. The future situation may not support manual operations as of now, as permitted by push buttons. Therefore, the workforce shall be well-versed with automation training as well.

Program Code:

```
int ledPin = 12;  // the number of the LED pin
int switchPin = 2;     //the number of the pushbutton pin
int val =0;
void setup() {
  // put your setup code here, to run once:
pinMode(ledPin, OUTPUT);      // initialize the LED pin as an output:
pinMode(switchPin, INPUT);     // initialize the pushbutton pin as an input
}
void loop() {
  // put your main code here, to run repeatedly:
val = digitalRead(switchPin);
if (val== HIGH)
{
  digitalWrite (ledPin, HIGH);  // turn LED on:
}
else
{
  digitalWrite(ledPin, LOW);    // turn LED off:
  }
}
```

Procedure

- Open Arduino IDE, Select the Arduino UNO board and Port.
- Write the Code using Arduino IDE Software.
- Initialize the LED Pin and Switch Pin to Arduino Data Pin 12 and 2 respectively.
- Make the Connections as shown in the Circuit Diagram and Calculate distance.

Particulars
Arduino UNO:1
Push Button-1
LED Pin-1
USB cable for uploading the code-1
Resistor (220 ohm)
Breadboard -1

Applications:
- Solenoid lock opening and closing
- Hand sanitizer
- Automated tap control

Circuit Diagram

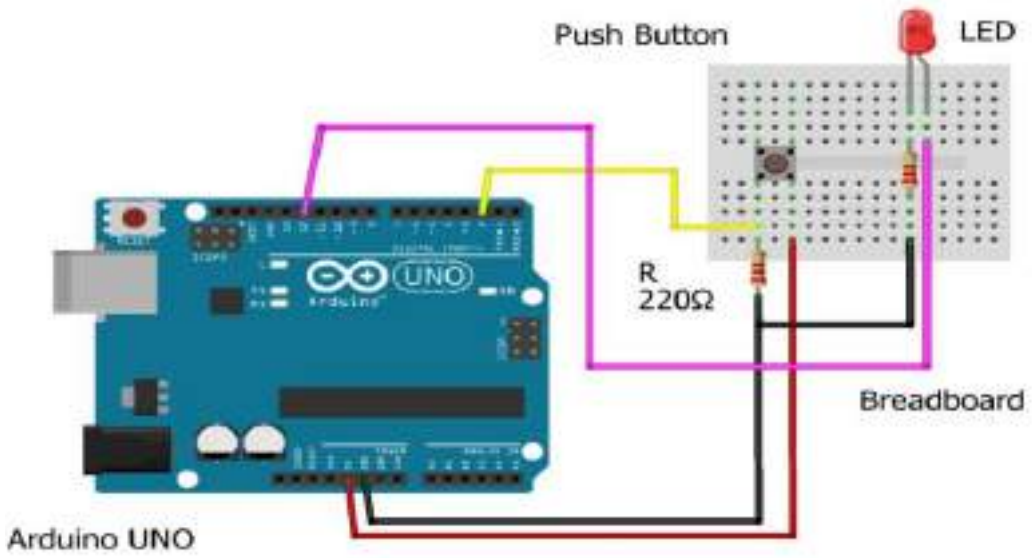

Fig. 6: Interfacing Push Button to LED with Arduino UNO

Result:

The Program to turn ON/OFF the LED using Push button has been written and executed successfully using Arduino UNO.

Date: **Signature of Faculty**

Experiment No: 1

Ultrasonic Sensor-LED Interface

Aim: Interfacing LED with Ultrasonic Sensor using Arduino Board to indicate the obstacle

Theory: Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules include ultrasonic transmitters, receiver, and control circuit. In this experiment the ultrasonic sensor (HC-SR04) in Arduino where we will turn on a led with the variation of distance and print the distance from an object to the serial monitor.

Circuit Diagram:

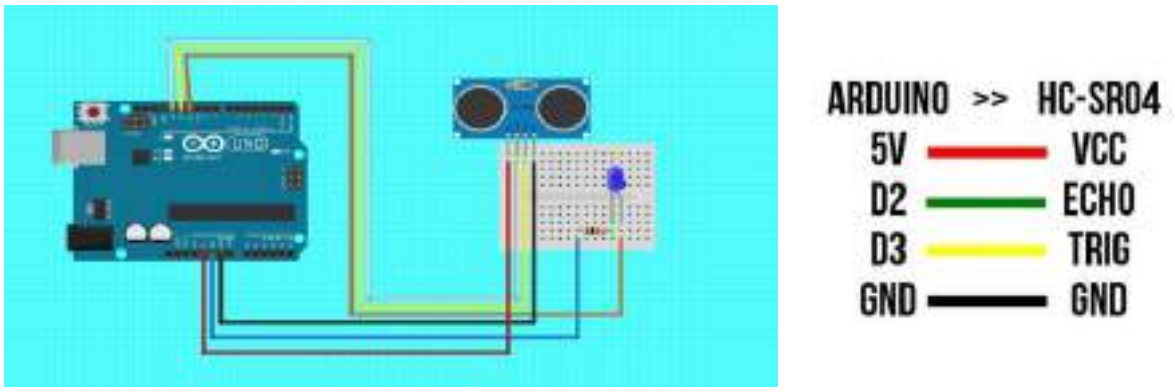

Fig.7: Interfacing LED with Ultrasonic Sensor using Arduino Board

Procedure

- Open Arduino IDE, Select the Arduino UNO board and Port.
- Write the Code using Arduino IDE Software.
- Define the Trigger Pin and Echo Pin to Arduino Data Pin 3 and 2 respectively.
- Set LED Pin
- Define the distance calculating formula in Void loop sketch.
- Now upload the code to the Arduino
- Make the Connections as shown in the Circuit Diagram.
- Move your hand in front of Sensor and see the LED's State.

Particulars

Arduino UNO:1
Ultrasonic Sensor -1
LED Pin-1
USB cable for uploading the code-1
Breadboard -1

Applications:

- Automatic Hand wash / Sanitizer Dispenser
- Water Level Indicating System
- Intruder Alarm

Program Code:

```
int trigPin = 9;
int echoPin = 8;
int led_Pin = 3;
int distance_threshold = 40;
float duration_us, distance_cm;

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led_Pin, OUTPUT);
}

void loop() {
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

duration_us = pulseIn(echoPin, HIGH);

distance_cm = 0.017 * duration_us;
if(distance_cm < distance_threshold)
digitalWrite(led_Pin, HIGH);
else digitalWrite(led_Pin, LOW);

Serial.print("distance: ");
Serial.print (distance_cm);
Serial.println(" cm");
delay(500);
}
```

Result:

Date: **Signature of Faculty**

Experiment No -2

Temperature and Humidity Sensor with LCD interfacing

AIM: Interfacing Humidity sensor with Arduino UNO to detect humidity and temperature and to display it on an LCD

Theory: Humidity sensors are electronic devices that measure and report the moisture and air temperature of the surrounding environment where they are deployed e.g., in air, soil, or confined spaces. Humidity measurements indicate the concentration of water vapour presented in the air.

In this experiment, we are going to interface a DHT11 temperature and humidity sensor and display the data on a 16x2 LCD. The project displays temperature in Celsius and humidity with an LCD screen. The DHT11 sensor can be used for a weather station, weather balloon, drone, or greenhouse.

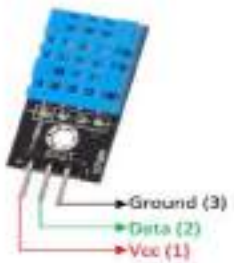

Fig 8: Pin diagram of Humidity Sensor

Working Principle of DHT11 humidity Sensor

The DHT11 sensor has a thermal resistance. Practically, it's a resistance that changes with temperature. Hence, the term Negative resistance coefficient (NTC) is associated with this. NTC means that resistance decreases with the increment of the temperature. Also, the sensor includes the small PCB inside which estimates the analog signal, converts the analog signals into digital, and gives the digital output.

Pin number	Pin Name	Description
1	Vcc	The power supply pin requires voltage from 3.5V to 5.5V
2	Data	Output pin for both humidity and temperature through serial Data
3	NC	No connection Pin. This pin is not for connecting.
4	Ground	Connection of sensor to the system's ground

Pinout of Humidity Sensor

Features of humidity Sensor

- The sensor ensures high reliability.
- Full range temperature compensated.
- Relative humidity and temperature measurement
- Calibrated digital signal.
- Outstanding long-term stability
- Extra components not needed.
- Long transmission distance
- Low power consumption
- 4 pins packaged and fully interchangeable.
- Fast response
- Anti-interference ability
- Cost Effective

Specifications of humidity Sensor

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16 bits
- Accuracy: ±1°C and ±1%

LCD display

LCD (Liquid Crystal Display) is a type of flat panel display which uses liquid crystals in its primary form of operation. LEDs have a large and varying set of use cases for consumers and businesses, as they can be commonly found in smartphones, televisions, computer monitors and instrument panels.

Fig 9: Pin diagram of Liquid Crystal Display

Circuit diagram

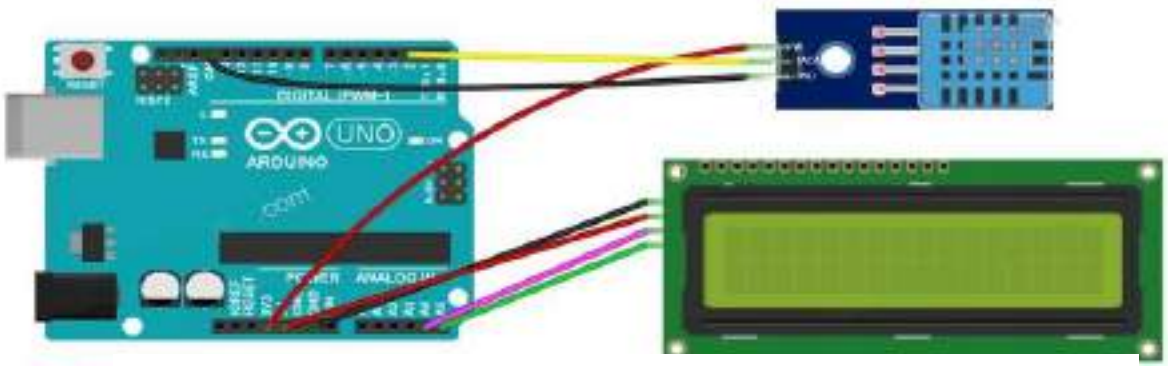

Fig.10: Interfacing LCD with Humidity Sensor using Arduino Board

Particulars

Arduino UNO:1
Humidity Sensor - 1
DC 5 V Supply -1
LCD -1
Buzzer -1
USB cable for uploading the code-1
Bread Board -1

Applications:

- IoT Environmental Monitoring
- Weather Station
- Climate Control
- Home Automation Systems

Procedure

- Connect the 5-volts pin of the Arduino to the V_{CC} pin of the humidity sensor.
- Connect the 5-volts pin of the Arduino to the V_{CC} pin of the LCD Display.
- Join the GND pin of the Arduino to the GND pin of the humidity sensor and lcd display.
- Connect the data signal pin of humidity sensor to pin 8 of Arduino.
- Connect SDA of LCD display to A4 pin Arduino
- Connect SCL of LCD display to A5 pin Arduino.
- Now upload the code to the Arduino.

Program Code

```
#include"DHT.h"
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN,DHTTYPE);

void setup(){

}
void loop(){
  float humi=dht.readHumidity(0);
  float tempC=dht.readTemperature();
  Serial.print("humidity: ");
  Serial.print(humi);
  Serial.print("%");
  Serial.print("     |    ");
  Serial.print("temperature");
  Serial.print(tempC);
  Serial.print("    .C   ");

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("humi: ");
  lcd.print(humi);
  lcd.print("%");
  Serial.begin(9600);
  dht.begin();
  lcd.setCursor(0,1);
  lcd.print("Temp: ");
  lcd.print(tempC);
  lcd.print("    .C    ");
  delay (500);
}
```

Result:

Date: **Signature of Faculty**

Experiment No. 3

Flame Sensor

Aim: Interfacing Flame sensor with Arduino UNO to detect the flame or fires by sensing the other than normal light and temperature

Theory: 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. Basically, flame detector senses the alteration in normal light, typically as an indicator of fire. Commercial flame detectors issue a signal to a fire alarm control panel as part of a fire alarm system. Household flame detectors, also known as fire alarms, generally issue an audible or visual alarm from the detector itself or several detectors if there are multiple devices interlinked. Flame sensor mainly detects the wavelength within the range of 760 nm -1100 nm from the light source. This sensor can be easily damaged by high temperature. Therefore, it is placed at a certain distance from the flame. The flame detection can be done from a 100cm distance, and the detection angle will be 60°. The output of flame sensor is an analog signal or digital signal. These sensors are used in firefighting robots like as a flame alarm. The response of flame sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.

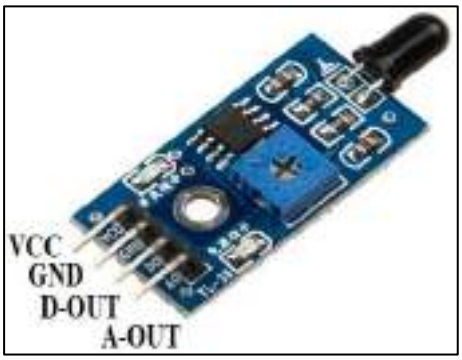

Fig. 11: Pin diagram of Flame

Flame Sensor Module

The pin configuration of this sensor is shown in the Figure.

Pin1 (V_{CC} pin): Voltage supply rages from 3.3V to 5.3V
Pin2 (GND): This is a ground pin.
Pin3 (A_{OUT}): This is an analog output pin.
Pin4 (D_{OUT}): This is a digital output pin.

Different types of Flame Sensors: IR single frequency, IR multi-spectrum, UV flame detectors, UV/ IR flame detectors

The working principle of flame sensor is based on the detection of electromagnetic radiation. The sensor uses the infrared flame flash method, which allows the sensor to receive the electromagnetic radiation through a coating of oil, dust, water vapour, ice, etc.

Features & Specifications

The features of this sensor include the following.

- Photosensitivity is high.
- Response time is fast.
- Simple to use.
- Sensitivity is adjustable.
- The detection angle is 600.
- It is responsive to the flame range.
- Accuracy can be adjustable.

- Operating voltage of this sensor is 3.3V to 5V.
- Analog voltage o/ps and digital switch o/ps
- The PCB size is 3cm X 1.6cm
- Power indicator & digital switch o/p indicator
- If the flame intensity is lighter within 0.8m then the flame test can be activated, if the flame intensity is high, then the detection of distance will be improved.

Pinout of Flame Sensor

Pin Name	Pin Number	Description
V_{CC}	1	Power pin requires an operating voltage of 5V
GND	2	Ground Pin
D_O	3	Digital output pin to get the digital output from the sensor
A_O	4	Analog output pin to indicate the intensity of fire or flame

Buzzer

The vibrating disk in a magnetic buzzer is attracted to the pole by the magnetic field. When an oscillating signal is moved through the coil, it produces a fluctuating magnetic field which vibrates the disk at a frequency equal to that of the drive signal.

Fig. 12: Buzzer

Circuit Diagram:

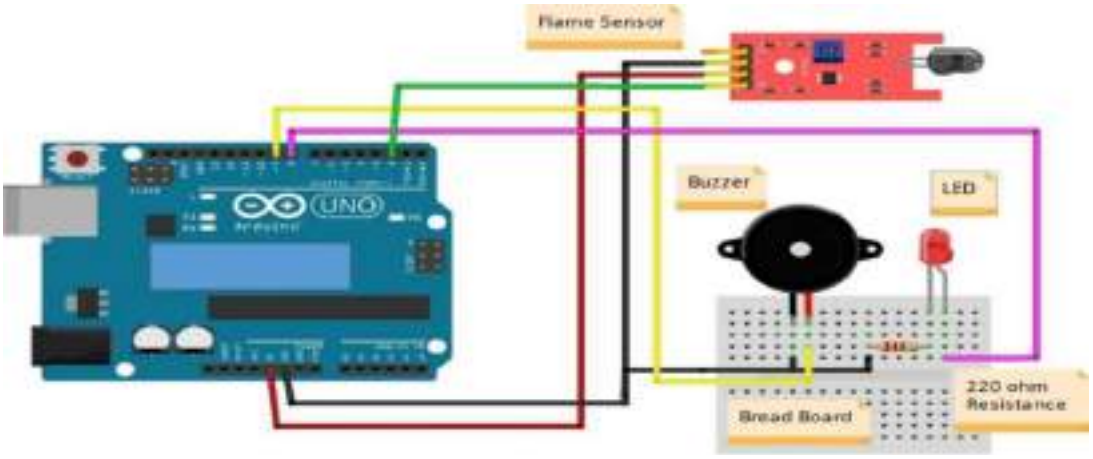

Fig. 13: Circuit diagram for interfacing Flame sensor with Arduino UNO

Procedure

- Connect the 5-volts pin of the Arduino to the V_{CC} pin of the smoke sensor.
- Join the GND pin of the Arduino to the GND pin of the flame sensor.
- Connect the RED LED Positive anode to pin 12 of Arduino and negative cathode to GND.
- Connect the positive wire of the buzzer to the Arduino Pin 10 and other side of wire to the GND.
- Now upload the code to the Arduino.

Applications:

- Low-Cost Fire Detection and alarm System
- SMS based Fire detection system using light variation and Temperature sensor.
- Arduino and IOT based home security system.
- Fire alarm, firefighting robot.
- Detection of flame/fire at Hydrogen stations, industrial heating, gas-powered cooking devices, domestic heating systems, industrial gas turbine.

Particulars

Arduino UNO:1
Flame Sensor - 1
DC 5 V Supply -1
LED -1
Buzzer -1
USB cable for uploading the code-1
Bread Board -1

Program Code

```
int Buzzer =13;
int FlamePin = 2;
int Flame = HIGH;
void setup() {
  pinMode (Buzzer, OUTPUT);
  pinMode(FlamePin, INPUT);
  Serial.begin(9600);
}

void loop() {
  Flame = digitalRead(FlamePin);
  if (Flame == HIGH)
    {
     Serial.println("HIGH FLAME");
    digitalWrite(Buzzer, HIGH);
  }
  else
{
Serial.println("NO FLAME");
digitalWrite(Buzzer, LOW);
}
}
```

Result:

Date: **Signature of Faculty**

Experiment No -4

Water level sensor

Aim: Interfacing Water Level sensor with Arduino UNO to detect the level of liquid

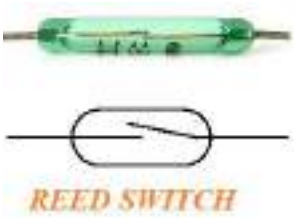

Fig. 14a: Reed Switch

Theory: A float sensor or float switch is used to detect the level of liquid within a tank. It is also called as magnetic float sensor or float switch as its working is same as of switch. Magnetic float sensor is an electromagnetic ON/OFF switch. It senses the level of water present in the tank by making switching connection.

To make a circuit closed/complete circuit float sensor use a magnetic reed switch.

Reed switch: It is an electrical switch that can operate/work by applying a magnetic field. It consists of ferromagnetic flexible metal reeds contact isolated in a hermetically sealed glass envelope. The metal contact is normally open. But as the magnetic field is applied or a magnet is placed near one metal end of the switch it attracts the other end of metal, which makes a path for current to pass through. When the magnet moves far away from the switch, the contacts demagnetize and will separate (breaking the circuit).

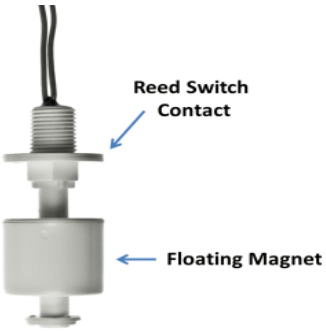

Fig. 14b: Connections for Float Sensor

The float sensor works on the same concept as the reed switch. In the reed switch, a magnet is used. A float sensor is designed in such a manner that the magnet is already present. When the sensor is placed in an empty tank the switch position is at the bottom but as water starts filling up switch position rises up which generates a magnetic field for the reed switch. When the water level goes down sensor breaks the circuit and the series led connected will turn off, when the level goes up sensor makes the circuit, and the series led connected will turn on.

Circuit Diagram

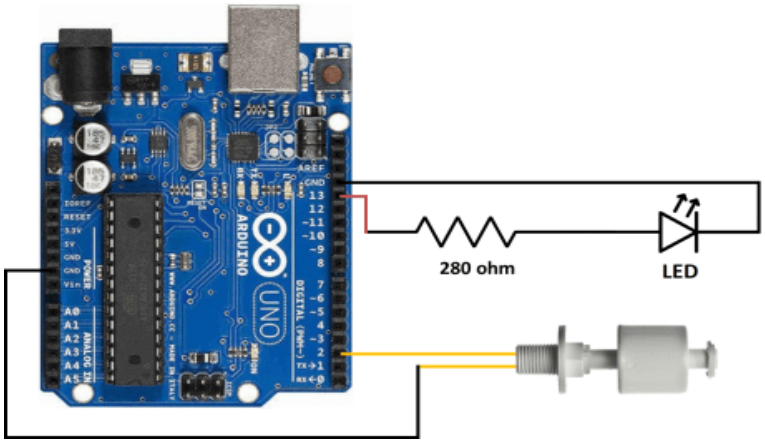

Fig 15: Interfacing Water Level sensor with Arduino UNO

Particulars

Arduino UNO:1
Magnetic float sensor
Led pin-1
Resistor 280 Ω
USB cable for uploading the code-1

Procedure

- The V_{CC} terminal of float sensor goes to digital pin on the Arduino board as shown in figure.
- GND terminal to ground of Arduino board.
- Led pin positive is connected to the pin Arduino board as shown in figure.
- Resistor of value 280 ohms is to the positive of the LED pin and negative of the Led Pin is connected to the ground of the Arduino Board.
- Now upload the Program.

Program Code:

```
int FloatSensor=2;
int led=13;
int buttonState=1;
void setup(){
  Serial.begin(9600);
  pinMode(FloatSensor,INPUT_PULLUP);
  pinMode(led,OUTPUT);

}
void loop(){
  buttonState=digitalRead(FloatSensor);
  if(buttonState==HIGH){
    digitalWrite(led,LOW);
    Serial.println("WATERLEVEL - HIGH");
  }
 else{
  digitalWrite(led,HIGH);
  Serial.println("WATERLEVEL -LOW");
 }
 delay(1000);
}
```

Applications:

- Pump or Valve Control
- High or Low-Level Alarm
- Heater Protection

Result:

Date: **Signature of Faculty**

Experiment No -5

Heartbeat Sensor

Aim: **Interfacing Heartbeat sensor with Arduino UNO to detect the pulse count of human beings**

Theory: A heartbeat sensor, also known as a heart rate sensor or heart rate monitor, is a device that measures the electrical signals generated by the heart and provides information about a person's heart rate or pulse. It is commonly used in medical settings for monitoring patients' heart health, as well as in fitness and sports applications for tracking and optimizing exercise performance. The Pulse Sensor is a plug-and-play heart-rate sensor for Arduino. The essence is an integrated optical amplifying circuit and noise eliminating circuit sensor. The front of the sensor comes with the heart logo. This is where you place your finger. On the front side, you will see a small round hole, from where the green LED shines. Just below the LED is a small ambient light photosensor **APDS9008** which adjust the brightness in different light conditions.

Fig. 16: Heartbeat Sensor with pin diagram

On the back of the module, you will find MCP6001 Op-Amp IC, a few resistors, and capacitors. This makes up the R/C filter network. There is also a reverse protection diode to prevent damage if you connect the power leads reverse.

Circuit Diagram:

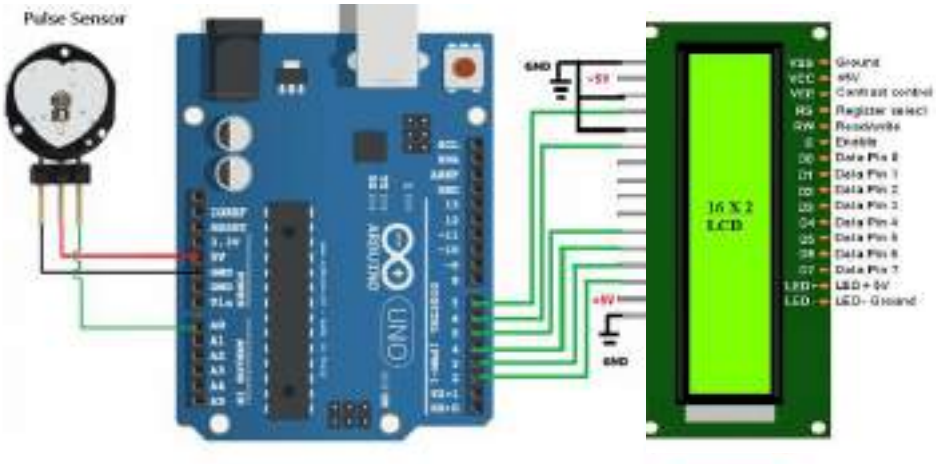

Fig. 17: Circuit diagram interfacing Heartbeat sensor with Arduino UNO

Procedure

- Connect the V_{CC} pin of the Sensor to Arduino 5V Pin.
- Connect the GND pin of sensor to GND.
- Connect the Analog output pin of the sensor to the A0 pin of the Arduino.
- Connect the LCD display to the Arduino according to the connections shown in the above diagram.
- Refer to the pin diagram of the LCD 16X2 given below.
- Now upload the Program.

Particulars

- Arduino UNO:1
- Breadboard
- Led pin-1
- 5V Battery
- USB cable for uploading the code-1

Applications:

- Sleep Tracking
- Monitoring of anxiety
- Alarm system
- Remote patient monitoring
- Health bands
- Advanced gaming consoles

Program Code:

```
#include <LiquidCrystal.h>
#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.
const int PulseWire = 0; // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
const int LED13 = 13; // The on-board Arduino LED, close to PIN 13.
int Threshold = 550; // Determine which Signal to "count as a beat" and which to ignore.
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
PulseSensorPlayground pulseSensor; // Creates an instance of the PulseSensorPlayground object called "pulseSenso
void setup() {
 Serial.begin(9600); // For Serial Monitor
lcd.begin(20,4);
// Configure the PulseSensor object, by assigning our variables to it.
pulseSensor.analogInput(PulseWire);
pulseSensor.blinkOnPulse(LED13); //auto-magically blink Arduino's LED with heartbeat.
pulseSensor.setThreshold(Threshold);
 // Double-check the "pulseSensor" object was created and "began" seeing a signal.
if (pulseSensor.begin())
{
Serial.println("We created a pulseSensor Object !");
lcd.setCursor(0,0);
lcd.print(" Heart Rate Monitor");
 }
}
 void loop() {
 int myBPM = pulseSensor.getBeatsPerMinute(); // Calls function on our pulseSensor object that returns BPM as an
// "myBPM" hold this BPM value now.
if (pulseSensor.sawStartOfBeat()) {
Serial.println("♥ A HeartBeat Happened ! ");
 Serial.print("BPM: ");
 Serial.print("BPM: ");
Serial.println(myBPM);
lcd.setCursor(0,2);
lcd.print("HeartBeat Happened !");
lcd.setCursor(5,3);
lcd.print("BPM: ");
lcd.print(myBPM);
}
delay(20);  }
```

Result:

Date: **Signature of Faculty**

Experiment No -6

Touch Sensor

Aim: Interfacing Touch sensor with Arduino UNO for Sensor Operation with LED Light and Solenoid lock

Theory: A touch sensor is an electronic sensor used in detecting and recording physical touch. Also known as tactile sensors, it's a small, simple, low-cost sensor made to replace old mechanical switches seen in the past. A touch sensor works like a switch, if there's contact, touch, or pressure on the surface of a touch sensor, it opens up an electrical circuit and allows current to flow through it. Touch sensors are broadly classified into two main types:

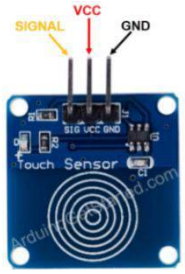

Fig. 18: Pin diagram of touch sensor

- ✓ Capacitive touch sensor– A Capacitive touch sensor measures touch based on electrical disturbance from a change in capacitance. It consists of an electrode film on top of the glass panel that's conductively coated with a printed circuit pattern around the outer viewing area.
- ✓ Resistive touch sensor– Resistive touch sensor measures touch by responding to the pressure applied to their surface. It consists of two conductive layers and a non-conductive separator. Unlike the capacitive touch sensors, it's not multi-touch compatible.

Vcc - 5V on the Arduino Board
Sig – pinout on Arduino board (whichever has been defined)
Gnd – Gnd on the Arduino board

Touch sensors work similar to a switch. When they are subjected to touch, pressure or force they get activated and act as a closed switch. When the pressure or contact is removed, they act as an open switch. The capacitive touch sensor contains two parallel conductors with an insulator between them. These conductors' plates act as a capacitor with a capacitance value C_0.

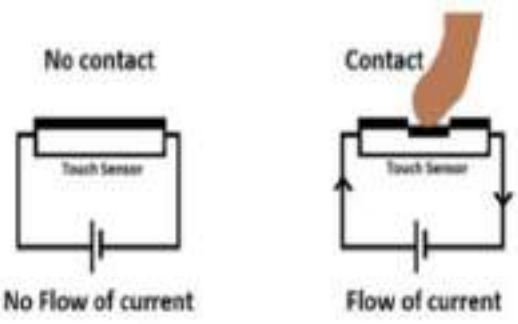

Fig. 19: Operation of touch sensor

When these conductor plates come in contact with our fingers, our finger acts as conductive object. Due to this, there will be an uncertain increase in the capacitance. A capacitance measuring circuit continuously measures the capacitance C_0 of the sensor. When this circuit detects a change in capacitance it generates a signal. The resistive touch sensors calculate the pressure applied on the surface to sense the touch. These sensors contain two conductive films coated with indium tin oxide, which is a good conductor of electricity, separated by a very small distance.

Across the surface of the films, a constant voltage is applied. When pressure is applied to the top film, it touches the bottom film. This generates a voltage drop which is detected by a controller circuit and signal is generated thereby detecting the touch.

Circuit Diagram

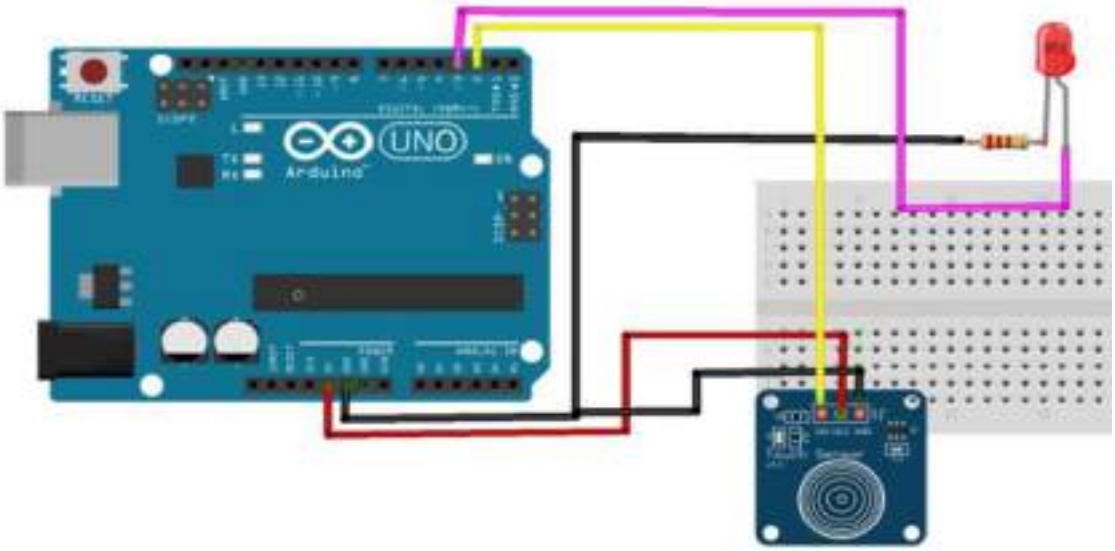

Fig. 20: Circuit diagram of interfacing Touch sensor with Arduino UNO

Procedure

- Connect the 5-volts pin of the Arduino to the V_{CC} pin of the Touch Sensor device.
- Join the GND pin of the Arduino to the GND pin of the Touch Sensor device.
- Attach the Touch Sensor device SIG pin with the digital- pin of the Arduino as shown in Circuit.
- The positive wire of the LED Pin has to be connected to the digital- pin of the Arduino and the negative wire to the GND pin of the Arduino Board.
- Next the code has to be uploaded to the Arduino.

Particulars

Arduino UNO:1
Touch Sensor - 1
LED Pin -1
USB cable for uploading the code-1
Bread Board -1

Program Code

```
const int TOUCH_SENSOR_PIN = 2; // the Arduino's input pin that connects to the sensor's SIGNAL pin
const int LED_PIN = 3;
void setup() {
Serial.begin(9600);
pinMode(TOUCH_SENSOR_PIN,INPUT); // initialize the Arduino's pin as an input
pinMode(LED_PIN,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
int touchState = digitalRead(TOUCH_SENSOR_PIN);
if (touchState == HIGH)
{
  Serial.println(" The Sensor being touched");
 digitalWrite(LED_PIN, HIGH);
 delay(1000);
}
else
{
  if (touchState == LOW)
{
  Serial.println(" The Sensor is untouched");
 digitalWrite(LED_PIN, LOW);
 delay(1000);
}
}
}
```

Applications:

- Touch sensor in robotics; a touch sensor is commonly used in robots, enabling basic movement and the ability to detect a touch in its surroundings (Ex. When the robot runs into something, the touch sensor can have it stop moving)
- Smartphones, automotive, industrial applications like hand sanitizer, Solenoid lock opening and closing.
- Touch sensor faucet in kitchens; allowing for control of running water without having to physically turn the knob.
- Most other applications that require pressure/distance measurement

Result:

Date: **Signature of Faculty**

Experiment No -7

Moisture Sensor

Aim: Interfacing Soil moisture sensor with Arduino UNO to detect moisture in the soil

Theory: A soil moisture-sensor is one kind of detector which measures the volumetric content of water inside the soil and gives the moisture level as output. The sensor is equipped with both analog and digital output, so it can be used in both analog and digital mode. The soil moisture sensor consists of two probes which are used to measure the volumetric content of water. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value. When there is more water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there will be less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower.

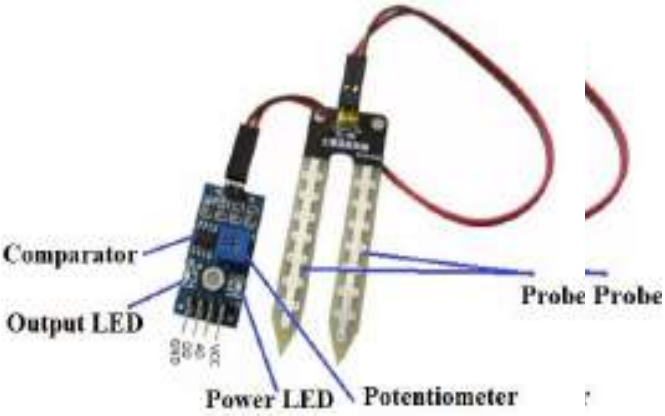

Fig. 21: Pin diagram of Moisture Sensor

Soil moisture Sensor Module

The pin configuration of this sensor is shown in the Figure.

Pin1 (V_{CC} pin): Voltage supply rages from 3.3V to 5.3V
Pin2 (AOUT): This is an analog output pin.
Pin3 (DOUT): This is a digital output pin.
Pin4 (GND): This is a ground pin.

Pin out of Soil Sensor

Pin Name	Pin Number	Description
V_{CC}	1	Power pin requires an operating voltage of 5V
D_O	2	Digital output pin to get the digital output from the sensor
A_O	3	Analog output pin to indicate the level of moisture in soil
GND	4	Ground Pin

Interfacing soil moisture sensor and Arduino – Analog Mode

To connect the sensor in the analog mode, we will need to use the analog output of the sensor. When taking the analog output from the soil moisture sensor, the sensor gives us the value from 10-500. The moisture is measured in percentage, so we will map these values from 0 - 100 and then we will show these values on the serial monitor. You can further set different ranges of the moisture values and turn on or off the water pump according to it. The features of this sensor include the following.

Interfacing soil moisture sensor and Arduino – Digital Mode

To connect the soil moisture sensor in the digital mode, we will connect the digital output of the sensor to the digital pin of the Arduino. The Sensor module contains a potentiometer with it, which is used to set the threshold value. This threshold value is then compared with the sensor output value using the comparator which is placed on the sensor module through the digital pin. When the sensor value will be greater than the threshold value, then the digital pin will give us 5V and the LED on the sensor will light up and when the sensor value will be less than this threshold value, then the digital pin will give us 0V and the LED light will go down.

Circuit Diagram

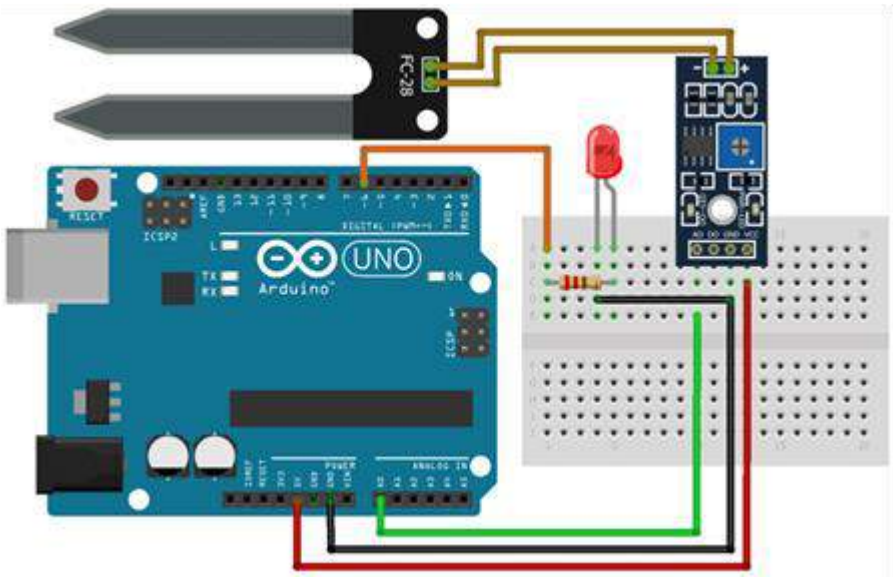

Fig. 22: Interfacing Moisture sensor with Arduino UNO

Procedure

- Connect the 5-volts pin of the Arduino to the V_{CC} pin of the soil moisture sensor.
- Join the GND pin of the Arduino to the GND pin of the soil moisture sensor.
- Join the A0 pin of the Arduino to the A0 pin of the soil moisture sensor.
- Connect the RED LED Positive anode to pin 12 of Arduino and negative cathode to GND.
- Connect the positive wire of the buzzer to the Arduino Pin 10 and other side of wire to the GND.
- Now upload the code to the Arduino.

Particulars

- Arduino UNO:1
- Soil Moisture Sensor - 1
- DC 5 V Supply -1
- LED Light -1
- Buzzer -1
- USB cable for uploading the code-1
- Bread Board -1

Program Code

```
int sensor_pin = A0; // Soil Sensor input at Analog PIN A0
int output_value;
void setup() {
   Serial.begin(9600);
   Serial.println("Reading From the Sensor ...");
   delay(2000);
   }
 void loop() {
   output_value= analogRead(sensor_pin);
   output_value = map(output_value,500,10,0,100);
   Serial.print("Mositure : ");
   Serial.print(output_value);
   Serial.println("%");
   delay(1000);
   }
```

Interfacing LED

```
int led_pin =13;
int sensor_pin =8;
void setup() {
  pinMode(led_pin, OUTPUT);
  pinMode(sensor_pin, INPUT);
}
void loop() {
  if(digitalRead(sensor_pin) == HIGH){
    digitalWrite(led_pin, HIGH);
  } else {
    digitalWrite(led_pin, LOW);
    delay(1000);
  }}
```

Applications:

- Detect moisture of soil
- Detect the growth of plants.
- Arduino and IOT based smart irrigation system.

Result:

Date: **Signature of Faculty**

Experiment No -8

LDR Sensor

Aim: Interfacing LDR sensor with Arduino UNO to switch on/off LED or 230V bulb using a 5V Relay.

Theory: LDR sensor module is used to detect the intensity of light. It is associated with both analog output pin and digital output pin labelled as AO and DO respectively on the board. When there is light, the resistance of LDR will become low according to the intensity of light. The greater the intensity of light, the lower the resistance of LDR. The sensor has a potentiometer knob that can be adjusted to change the sensitivity of LDR towards the light.

Fig 23: LDR Sensor

Light-dependent resistors have a low cost and simple structure. These resistors are frequently used as light sensors. These resistors are mainly used when there is a need to sense the absence and presence of the light such as burglar alarm circuits, alarm clocks, and light intensity meters. LDRs (light-dependent resistors) are used to detect light levels, for example, in automatic security lights. Their resistance decreases as the light intensity increases in the dark and at low light levels, the resistance of an LDR is high and little current can flow through it.

Fig. 24: Relay

Vcc - 5V on the Arduino Board

Out – pinout on Arduino board (whichever has been defined)
Gnd – Gnd on the Arduino board

Relay is one kind of electro-mechanical component that functions as a switch. The relay coil is energized by DC so that contact switches can be opened or closed. A single channel 5V relay module generally includes a coil, and two contacts normally open (NO) and normally closed (NC).

Signal Pin: The signal pin is mainly used for controlling the relay. This pin works in two cases active low otherwise active high. So, in the active-low case, the relay activates once we provide an active low signal toward the signal pin, whereas, in an active high case, the relay will trigger once we provide a high signal toward the signal pin.

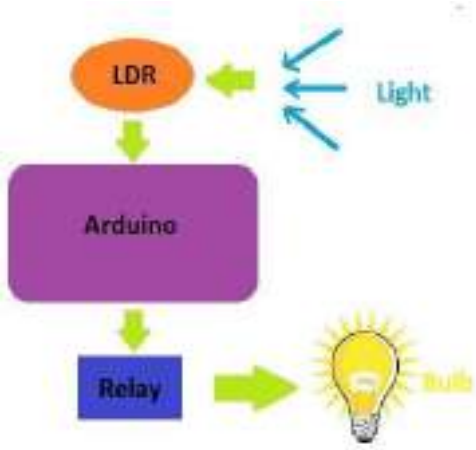

Fig. 25: Arduino LDR Sensor

This system works by sensing the intensity of light in its environment. The sensor that can be used to detect light is an LDR. The LDR gives out an analog voltage when connected to V_{CC} (5V), which varies in magnitude in direct proportion to the input light intensity on it. That is, the greater the intensity of light, the greater the corresponding voltage from the LDR will be. Since the LDR gives out an analog voltage, it is connected to the analog input pin on the Arduino. The Arduino, with its built-in ADC (analog-to-digital converter), then converts the analog voltage (from 0-5V) into a digital value in the range of (0-1023). When there is sufficient light in its environment or on its surface, the converted digital values read from the LDR through the Arduino will be in the range of 800-1023. Furthermore, we then program the Arduino to turn on a relay. Correspondingly, turn on an appliance (light bulb), when the light intensity is low (this can be done by covering the surface of the LDR with any object), that is, when the digital values read are in a higher range than usual.

Circuit Diagram

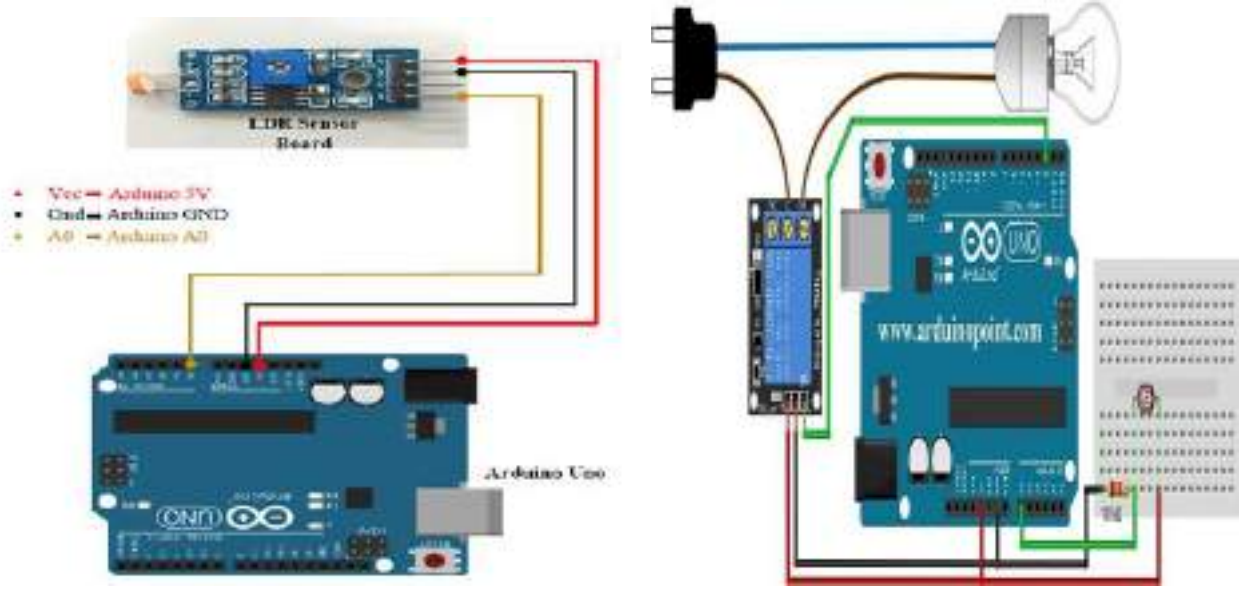

Fig. 26 Circuit diagram for interfacing LDR sensor with Arduino UNO and LDR connected to 5V relay to switch on/off 230V bulb.

Particulars

Arduino UNO:1
LDR sensor
Led pin-1
DC-5V supply
Relay-5V
230V bulb
USB cable for uploading the code-1
Breadboard

Procedure

- Connect the 5-volts pin of the Arduino board to the V_{CC} pin of the Relay module.
- Connect the GND pin of the Arduino board to the GND pin of the Relay module.
- LDR connected to analog pin A0 of the Arduino board.
- Attach the Relay IN pin of the Relay module with the digital- pin of the Arduino as shown above.
- A bulb connects with a 5V relay module's NO pin and NC pin.
- Now upload the code to the Arduino.

Program Code:

```
const int ledPin = 2;
const int ldrPin = A0;
void setup()
{
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
pinMode(ldrPin, INPUT);
}
void loop()
{
int ldrStatus = analogRead(ldrPin);
if (ldrStatus <= 200)
{
digitalWrite(ledPin, LOW);
Serial.print("Its BRIGHT, Turn OFF the LED : ");
Serial.println(ldrStatus);
}
else
 {
digitalWrite(ledPin,HIGH);
Serial.print("Its DARK, Turn ON the LED : ");
Serial.println(ldrStatus);
}
}
```

Applications:

- Night Light Control
- Street Light Control
- Automatic Headlight Dimmer
- Oil Burner Flame Out

Result:

Date: **Signature of Faculty**

Experiment No -9

Smoke sensor

Aim: Interfacing Smoke sensor with Arduino UNO to detect fire by sensing small gases and heat in the air

Theory: A smoke detector is a device that senses smoke, typically as an indicator of fire. Commercial smoke detectors issue a signal to a fire alarm control panel as part of a fire alarm system. Household smoke detectors, also known as smoke alarms, generally issue an audible or visual alarm from the detector itself or several detectors if there are multiple devices interlinked.

Smoke detectors are usually housed in plastic enclosures, typically shaped like a disk about 150 millimeters (6 in) in diameter and 25 millimeters (1 in) thick, but shape and size vary. Smoke can be detected either optically (photoelectric) or by physical process (ionization). Detectors may use one or both sensing methods. Sensitive alarms can be used to detect and deter smoking in banned areas. Smoke detectors in large commercial and industrial buildings are usually connected to a central fire alarm system. Domestic smoke detectors range from individual battery-powered units to several interlinked units with battery backup. With interlinked units, if any of them detects smoke, all of the alarms will trigger. This happens even if household power has gone out.

The two most commonly recognized smoke detection technologies are ionization smoke detection and photoelectric smoke detection.

Fig. 27: Smoke Sensor in diagram

Ionization smoke alarms are generally more responsive to flaming fires. How they work: Ionization-type smoke alarms have a small amount of radioactive material between two electrically charged plates, which ionizes the air and causes current to flow between the plates. When smoke enters the chamber, it disrupts the flow of ions, thus reducing the flow of current and activating the alarm. Download this chart on ionization smoke alarms (PDF, 943 KB).

Photoelectric smoke alarms are generally more responsive to fires that begin with a long period of smoldering (called “smoldering fires”). How they work: Photoelectric-type alarms aim a light source into a sensing chamber at an angle away from the sensor. Smoke enters the chamber, reflecting light onto the light sensor, triggering the alarm. Download this chart on photoelectric smoke alarms (PDF, 782 KB).

For each type of smoke alarm, the advantage it provides may be critical to life safety in some fire situations. Home fatal fires, day or night, include a large number of smouldering fires and a large number of flaming fires. You cannot predict the type of fire you may have in your home or when it will occur. Any smoke alarm technology, to be acceptable, must perform acceptably for both types of fires in order to provide early warning of fire at all times of the day or night and whether you are asleep or awake.

For best protection, use both types of smoke alarm technologies. For best protection, it is recommended both (ionization and photoelectric) technologies be used in homes. In addition to individual ionization and photoelectric alarms, combination alarms that include both technologies in a single device are available.

Internal Structure of Smoke Sensor

They enclose the sensor in two layers of fine stainless-steel mesh which is known as an Anti-explosion network. It assures that the heater element inside the sensor will not cause an explosion. The sensing element forms the star-shaped structure. Six connecting legs expand beyond the Bakelite base. From those six legs, two leads heat the sensing element and are connected through the well-known collective alloy called Nickel-Chromium coil. The remaining four legs are actually responsible for output signals. They made the sensing element from Aluminium Oxide (AL_2O_3) and have a coating of Tin Dioxide (SnO_2), which is sensitive to combustible gases. The working principle of the MQ-2 sensor is based on the resistor. When it detects smoke or fire, the value of its resistance changes. That's how the gas or smoke gets detected. The sensor works on a very low voltage of around 5Volts. The given potentiometer can change the sensitivity of the digital pin, it draws a current of 500mA.

Features of Smoke Sensor

- Operating Voltage is +5V.
- Can Measure or detect LPG, Alcohol, Propane, Hydrogen, CO and even methane.
- Analog output voltage: 0V to 5V
- Digital Output Voltage: 0V or 5V (TTL Logic)
- Preheat duration 20 seconds.
- Can be used as a Digital or analog sensor.
- It can vary the Sensitivity of Digital pin using the potentiometer.
- Wide detecting Scope
- Fast response and High sensitivity
- Stable and long-life Simple drive circuit

Pin out of Smoke Sensor

Pin Name	Pin Number	Description
V_{CC}	1	Power pin requires an operating voltage of 5V
GND	2	Ground Pin
D_O	3	Digital output pin to get the digital output from the sensor
A_O	4	Analog output pin. Its output is based on the intensity of the gas.

Pinout of Smoke Sensor

Buzzer

The vibrating disk in a magnetic buzzer is attracted to the pole by the magnetic field. When an oscillating signal is moved through the coil, it produces a fluctuating magnetic field which vibrates the disk at a frequency equal to that of the drive signal. A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons.

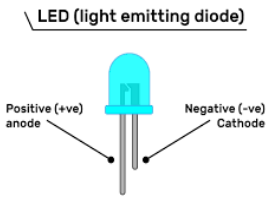

Fig. 28: Lead

Fig. 29: Buzzer

Circuit Diagram

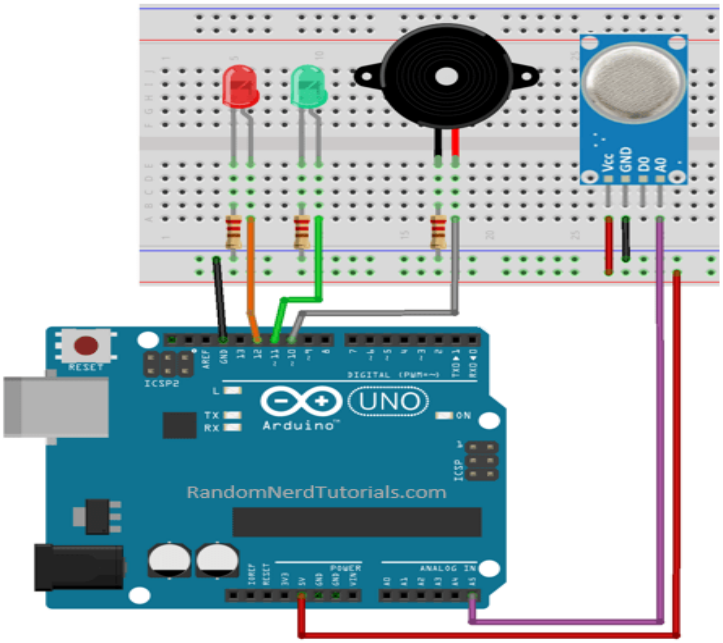

Fig. 30: Interfacing Smoke sensor with Arduino UNO

Particulars

- Arduino UNO:1
- Flame Sensor - 1
- DC 5 V Supply -1
- LED -1
- Buzzer -1
- USB cable for uploading the code-1
- Bread Board -1

Procedure

- Connect the 5-volts pin of the Arduino to the V_{CC} pin of the smoke sensor.
- Join the GND pin of the Arduino to the GND pin of the flame sensor.
- Connect the RED LED Positive anode to pin 12 of Arduino and negative cathode to GND.
- Connect the positive wire of the buzzer to the Arduino Pin 10 and other side of wire to the GND.
- Now upload the code to the Arduino.

Program Code:

```
int redLed=12;
int greenLed=11;
int buzzer=10;
int smokeA0=A5;
int sensorThres=150; // Your threshold value
void setup() {
  // put your setup code here, to run once:
  pinMode(redLed,OUTPUT);
  pinMode(greenLed,OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(smokeA0,INPUT);
  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  int analogSensor=analogRead(smokeA0);
  Serial.print("Pin A0");
  Serial.println(analogSensor);
  if(analogSensor>sensorThres) // Checks if it has reached the thr
  {
    digitalWrite(redLed,HIGH);
    digitalWrite(greenLed,LOW);
    tone(buzzer,1000,200);
  }
  else
  {
    digitalWrite(redLed,LOW);
    digitalWrite(greenLed,HIGH);
    noTone(buzzer);
  }
  delay(100);
}
```

Applications:

- Liquid level control
- People detection for counting.
- Vehicle detection for car wash and automotive assembly
- Presence detection

Result:

Date: **Signature of Faculty**

Experiment No -10

IR sensor

Aim: Interfacing IR sensor with Arduino and servo motor

Theory: IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An **IR sensor** can measure the heat of an object as well as detects the motion. Usually, in the **infrared spectrum**, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations. On the back of the module, you will find MCP6001 Op-Amp IC, a few resistors, and capacitors. This makes up the R/C filter network. There is also a reverse protection diode to prevent damage if you connect the power leads reverse.

Fig. 31: IR Sensor

The emitter is simply an IR LED **(Light Emitting Diode)** and the detector is simply an IR photodiode. Photodiode is sensitive to IR light of the same wavelength which is emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.

Circuit Diagram:

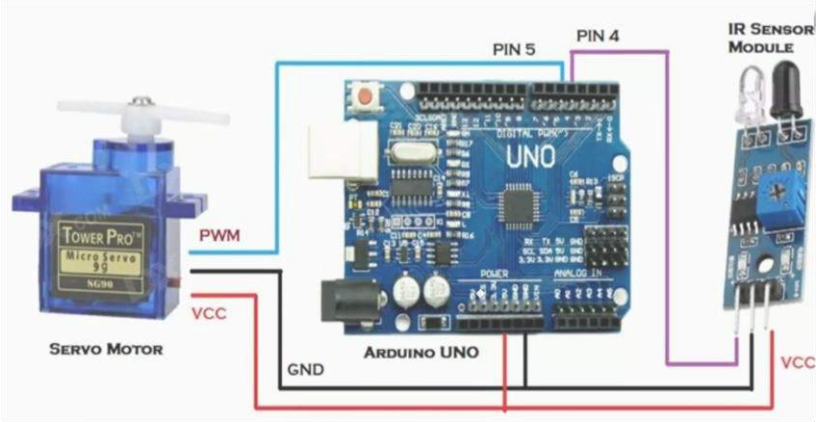

Fig. 32: Circuit diagram interfacing IR sensor with Arduino UNO and Servo motor

- VCC pin (typically red) needs to be connected to **VCC** (5V)
- GND pin (typically black or brown) needs to be connected to **GND** (0V)
- Signal pin (typically yellow or orange) receives the PWM control signal from an Arduino's pin.

Particulars

- Arduino UNO:1
- 5V Battery
- USB cable for uploading the code-1
- Servomotor
- IR sensor

Procedure

- Connect the V_{CC} pin of the Sensor to Arduino 5V Pin.
- Connect the GND pin of sensor to GND.
- Connect the Analog output pin of the sensor to the pin No 4 of the Arduino.
- Connect the Servo motor to the Arduino according to the connections shown in the above diagram.
- Now upload the Program.

Applications:

- Proximity sensor
- Item counter
- Alarm system
- Human body detection
- Gas analyser

Program Code:

```
#include <Servo.h>
#include <Servo.h>
Servo tap_servo;
int sensor_pin = 4;              // defines pin numbers
int tap_servo_pin =5;            // defines pin numbers
int val;
void setup(){
pinMode(sensor_pin,INPUT);       // declare sensor pin as input
tap_servo.attach(tap_servo_pin);
}
void loop(){
val = digitalRead(sensor_pin);
if (val==0) {
tap_servo.write(0);             // turns the motor off
}
if (val==1)
 {    tap_servo.write(180);  // turns the motor on
  } }
```

Result:

Date: **Signature of Faculty**

Demonstration Experiment No -1

Door Sensor

Aim: **Interfacing Door sensor with Arduino UNO to detect opening of the door and to turn LED on until door is open**

Theory: The door sensor is an essential component used in-home security system. The designing of these sensors can be done with two parts that are arranged in parallel to each other. So that the circuit can be formed. When someone tries to open the door then these parts will get separated and breaks the circuit. So, the control panel will activate to generate an alarm. These sensors are very easy to install and portable. There are different types of door sensors available in the market. Most of these sensors use a reed switch & a magnet to decide the status of a door whether it is opened or closed. Reed switch is applicable in countless devices like doorbells to laptops which depend on electrical connectors.

When the reed switch & a magnet are arranged close to each other, then the switch will be closed then an electrical current will be supplied. Similarly, when the switch is opened, the two parts will be detached then the circuit will deactivate due to no current flow. The door sensors come with two pieces a read switch as well as a magnet. Here the read switch is connected to the door and magnet is arranged to parallel to the switch. So that closed-circuit can be formed when the door is closed. When the door gets open, the switch and the magnet will separate to break the circuit. So, the sensor gives an indication to the control panel to generate an alarm.

Circuit Diagram:

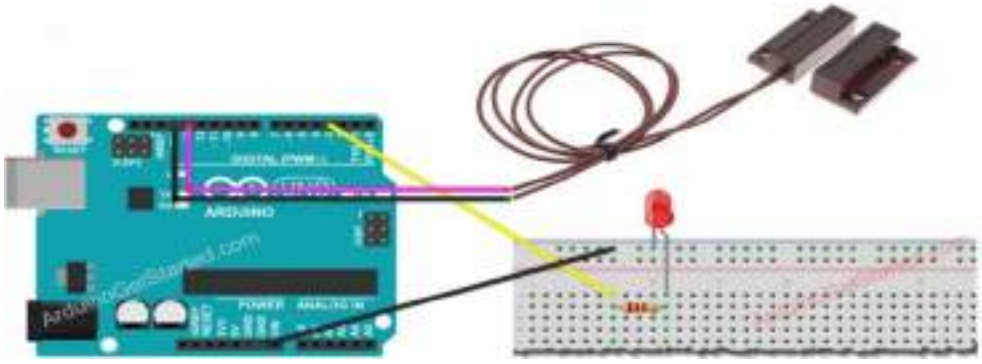

Fig.33: Circuit diagram for interfacing Door sensor with Arduino UNO

Particulars

Arduino UNO:1
Door Sensor - 1
LED Pin -1
USB cable for uploading the code-1
Jumper wires -1

Applications:

- Preventing cargo theft
- Protecting refrigerated loads
- Reducing energy and food waste
- Minimizing the costs associated with load loss.

Procedure

- Connect Arduino to PC Via USB Cable
- Open Arduino IDE, Select the Arduino UNO board and Port.
- Write the Code using Arduino IDE Software.
- Initialize door sensor and LED Pin to Arduino Data Pin 13 and 3 respectively.
- Now upload the code to the Arduino
- Make the Connections as shown in the Circuit Diagram.

Program Code:

```
const int door_sensor_pin=13; // Arduino pin connected to door sensor's pin
const int led_pin=3;
int doorstate;
void setup() {
Serial.begin(9600);     // initialize serial
pinMode(door_sensor_pin,INPUT_PULLUP); // set arduino pin to input pull-up mode
pinMode(led_pin,OUTPUT);
}
void loop() {
  doorstate=digitalRead(door_sensor_pin); // read state
  if(doorstate==HIGH)
  {
    Serial.println("The door is open");
    digitalWrite(led_pin,HIGH);
    delay(1000);
  }
  else
  {
  Serial.println("The door is closed");
    digitalWrite(led_pin,LOW);
delay(1000);
  }
}
```

Result:

The Program to run Door sensor to detect opening of the door has been written and executed successfully using Arduino UNO.

Date: **Signature of Faculty**