Government Polytechnic Kanpur

Department of [DIPLOMA IN INFORMATION TECHNOLOGY]

LAB MANUAL

Internet of Things

Lab Title: [**Internet of Things Lab Manual**]
Subject: [Internet of Things-**2268**]
Course: [Diploma in Information Technology]
Semester: [5th Semester)]
Session: [2024-2025]
Lab Instructor: [**Mr. Himanshu Singh**]

Submitted by:
Name:_____________________________________
Roll Number:________________________________
EnRollment Number:________________________
Class: [INFORMATION TECHNOLOGY 5TH Semester]

INDEX

IOT SYLLABUS:

Practical using Arduino-Interfacing Sensors :

1. Installation of Arduino IDE.
2. Interfacing Light Emitting Diode (LED)- Blinking LED.
3. Interfacing Button and LED – LED blinking when button is pressed.
4. Interfacing Light Dependent Resistor (LDR) and LED, displaying automatic night lamp
5. Interfacing Temperature Sensor (LM35) and/or humidity sensor (e.g. DHT11).
6. Interfacing Liquid Crystal Display (LCD) – display data generated by sensor on LCD.
7. Interfacing Air Quality Sensor-pollution (e.g. MQ135) - display data on LCD, switch on LED when data sensed is higher than specified value.
8. Interfacing Bluetooth module (e.g. HC05)- receiving data from mobile phone on Arduino and display on LCD.
9. Interfacing Relay module to demonstrate Bluetooth based home automation application. (using Bluetooth and relay).

INSTALLATION OF ARDUINO

// Installation of Arduino IDE
STEP 01:
VISIT https://www.arduino.cc/

STEP 02:
GOTO SOFTWARE SECTION(TAB).
https://www.arduino.cc/en/software

STEP 03:
GOTO DOWNLOAD SECTION.

STEP 04:
USE EITHER:
*************Arduino Web Editor*****************

Start coding online and save your sketches
in the cloud.
The most up-to-date version of the IDE includes
all libraries and also supports new Arduino boards.

OR

CHECK OUT ------->>>>>>

DOWNLOAD OPTIONS
->Windows
->Win 10 and newer, 64 bits

->Windows
->MSI installer

->Windows

ZIP file

->Linux
->AppImage 64 bits (X86-64)

->Linux
->ZIP file 64 bits (X86-64)

->macOS
->Intel, 10.14: “Mojave” or newer, 64 bits

->macOS
Apple Silicon, 11: “Big Sur” or newer, 64 bits

STEP 05:
CLICK ON ANY OF ABOVE ACCORDING TO YOUR OS MACHINE

STEP 06:
CLICK ON JUST DOWNLOAD
YOU WILL GET A MSI INSTALLER OR ZIP FOLDER.

STEP 07:
INSTALL THE PACKAGE ON YOUR MACHINE.

YOUR ARDUINO IDE WILL BE INSTALLED ON MACHINE.

LAB TASK 01

// Interfacing Light Emitting Diode (LED)- Blinking LED

// Write a program to Blink default Light Emitting Diode(LED) on Arduino board with the delay of 2 sec.

```
// DIGITAL PIN 13 OF ARDUINO BOARD WILL BE CONNECTED TO ANODE LONGER LEG OF LED.
// GND OF ARDUINO BOARD WILL BE CONNECTED TO CATHODE SORTER LEG OF LED
void setup()
{
pinMode(13, OUTPUT);
}
void loop()
{
digitalWrite(13, HIGH);
delay(2000);
digitalWrite(13, LOW);
delay(2000);
}
```

LAB TASK 02

```
// Interfacing Button and LED – LED blinking when button is pressed.


// Write a program to interface Button and LED, so that LED blinks/glow when button is pressed.

// Pin definitions
const int buttonPin = 2;   // Pin number for the button
const int ledPin = 13;     // Pin number for the LED

// Variables
int buttonState = 0;       // Variable to store the button state (LOW or HIGH)

void setup() {
  // Set the button pin as input
  pinMode(buttonPin, INPUT);

  // Set the LED pin as output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Read the state of the button (LOW when not pressed, HIGH when pressed)
  buttonState = digitalRead(buttonPin);

  // Check if the button is pressed
  if (buttonState == HIGH) {
    // Turn on the LED
    digitalWrite(ledPin, HIGH);
  } else {
    // Turn off the LED
    digitalWrite(ledPin, LOW);
  }
}
```

LAB TASK 03

```
//  Interfacing Light Dependent Resistor (LDR) and LED, displaying automatic night lamp

/*
Connections:

Connect one leg of the LDR to the 5V pin on the Arduino.
Connect the other leg of the LDR to one leg of the resistor.
Connect the other leg of the resistor to the A0 pin on the Arduino.
Connect the short leg (cathode) of the LED to a digital pin on the Arduino (e.g., D2).
Connect the long leg (anode) of the LED through a current-limiting resistor (around 220 ohms) to
the GND pin on the Arduino.
*/



const int ldrPin = A0;  // LDR connected to analog pin A0
const int ledPin = 2;   // LED connected to digital pin D2

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

void loop() {
  int ldrValue = analogRead(ldrPin);  // Read LDR value (0 to 1023)

  Serial.print("LDR Value: ");
  Serial.println(ldrValue);

  // Adjust the threshold value based on ambient light conditions
  int threshold = 500;

  if (ldrValue < threshold) {
    digitalWrite(ledPin, HIGH);  // Turn on the LED
    Serial.println("Dark, Turning ON LED");
  } else {
    digitalWrite(ledPin, LOW);   // Turn off the LED
    Serial.println("Light, Turning OFF LED");
  }

  delay(1000);  // Add a delay to prevent rapid switching due to sensor noise
}
```

LAB TASK 04

// Interfacing Temperature Sensor (LM35) and/or humidity sensor (e.g. DHT11)

// Write a program to interface Light Dependent Resistor (LDR) and LED with Arduino board. Whenever there is sufficient light falls on LDR the LED is off and when there is dark around LDR the LED is put on.

```
const int ldrPin = A0;  // Connect LDR to analog pin A0
const int ledPin = 13;  // Connect LED to digital pin 13

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

void loop() {
  int ldrValue = analogRead(ldrPin);  // Read LDR value
  Serial.println(ldrValue);  // Print LDR value to serial monitor (optional)

  // Adjust the threshold value according to your environment
  int threshold = 500;

  if (ldrValue < threshold) {
    digitalWrite(ledPin, HIGH);  // Turn on LED when it's dark
  } else {
    digitalWrite(ledPin, LOW);   // Turn off LED when there is sufficient light
  }

  delay(500);  // Adjust the delay time as needed
}
```

LAB TASK 05

```
// Interfacing Liquid Crystal Display (LCD) – display data generated by sensor on LCD
//GND-GND
//VCC-5V SUPPLY POWER
//SDA- A4
//SCL-A5
#include<Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x3F for a 16 chars and 2 line display

void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();      // Make sure backlight is on

  // Print a message on both lines of the LCD.

}

void loop() {
  lcd.setCursor(4,0);   //Set cursor to character 2 on line 0
  lcd.print("WELCOME");
  delay(1000);
  lcd.clear();

  lcd.setCursor(6,0);   //Set cursor to character 2 on line 0
  lcd.print("TO");
  delay(1000);
  lcd.clear();

  lcd.setCursor(4,1);   //Move cursor to character 2 on line 1
  lcd.print("PROGRAMMING");
  delay(1000);
  lcd.clear();
  }
```

LAB TASK 06

```
// Interfacing Air Quality Sensor-pollution (e.g. MQ135) - display data on LCD, switch on LED when data sensed is higher than specified value.

/*
Components Needed:
Arduino board
MQ135 air quality sensor
16x2 LCD display
LED
Resistor for the LED (e.g., 220Ω)
Jumper wires
Connect the MQ135 sensor to the Arduino.
Connect the LCD to the Arduino.
Connect the LED to a digital pin on the Arduino
*/

#include <LiquidCrystal_I2C.h>

// Include libraries for the LCD and MQ135
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_MQ135.h>

// Pin configuration
const int ledPin = 13; // Digital pin for the LED
const int thresholdValue = 500; // Set your threshold value here

// Create an instance of the Adafruit_MQ135 class
Adafruit_MQ135 mq135 = Adafruit_MQ135(0);

// Create an instance of the LiquidCrystal_I2C class
LiquidCrystal_I2C lcd(0x27, 16, 2); // Change the address if necessary

void setup() {
  // Initialize LCD
  lcd.begin(16, 2);
  // Initialize Serial Monitor for debugging
  Serial.begin(9600);

  // Set LED pin as output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Read the air quality value from the sensor
  float airQuality = mq135.readCO2();

  // Display the air quality value on the LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Air Quality:");
  lcd.setCursor(0, 1);
  lcd.print(airQuality);

  // Check if the air quality is above the threshold
  if (airQuality > thresholdValue) {
    // Turn on the LED
    digitalWrite(ledPin, HIGH);
  } else {
    // Turn off the LED
    digitalWrite(ledPin, LOW);
  }

  // Print the air quality value to Serial Monitor for debugging
  Serial.print("Air Quality: ");
  Serial.println(airQuality);

  // Delay for a moment before the next reading
  delay(1000);
}
```

LAB TASK 07

```
// Interfacing Bluetooth module (e.g. HC05)- receiving data from mobile phone on Arduino and display on LCD

// Write a program to interface LCD and Bluetooth module, to exhibit the values received from mobile handset via Bluetooth on LCD.

#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Set up the LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);  // I2C address 0x27, 16 column and 2 rows

// Set up SoftwareSerial for Bluetooth communication
SoftwareSerial bluetooth(10, 11);  // RX, TX pins for Bluetooth module

void setup() {
  // Start the LCD and Bluetooth
  lcd.begin(16, 2);
  bluetooth.begin(9600);

  // Display a welcome message on the LCD
  lcd.print("Bluetooth LCD");
  lcd.setCursor(0, 1);
  lcd.print("Data: ");

  delay(2000);
  lcd.clear();
}

void loop() {
  // Check if there is data available from the Bluetooth module
  if (bluetooth.available() > 0) {
    // Read the incoming data
    char data = bluetooth.read();

    // Display the received data on the LCD
    lcd.setCursor(6, 1);  // Set cursor to the second line
    lcd.print(data);
  }
}



// *************OR******************ANOTHER CODE

/*

Components Needed:
Arduino board (e.g., Arduino Uno)
Bluetooth module (e.g., HC-05)
LCD display (e.g., 16x2 character LCD)
Mobile phone with Bluetooth capabilities
Jumper wires
Steps:
1. Connect the HC-05 Bluetooth Module to Arduino:
Connect VCC to 5V on Arduino.
Connect GND to GND on Arduino.
Connect TXD to digital pin 10 on Arduino.
Connect RXD to digital pin 11 on Arduino.
2. Connect the LCD to Arduino:
Connect VCC to 5V on Arduino.
Connect GND to GND on Arduino.
Connect SDA to a digital pin (e.g., A4 on Arduino Uno).
Connect SCL to another digital pin (e.g., A5 on Arduino Uno).
3. Install Required Libraries:
Install the "LiquidCrystal_I2C" library for the LCD. You can do this through the Arduino IDE's
Library Manager.
Make sure to have the "SoftwareSerial" library for the Bluetooth module.
4. Write the Arduino Code:
Here's a simple example code to get you started.
This code assumes that the Bluetooth module
 is connected to pins 10 (TX) and 11 (RX),
 and the LCD is connected via I2C.

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

```
#include <SoftwareSerial.h>

SoftwareSerial bluetoothSerial(10, 11);  // RX, TX for Bluetooth
LiquidCrystal_I2C lcd(0x27, 16, 2);       // I2C address 0x27, 16 column and 2 rows

void setup() {
  lcd.begin(16, 2);
  lcd.print("Bluetooth Demo");

  bluetoothSerial.begin(9600);
}

void loop() {
  if (bluetoothSerial.available() > 0) {
    char incomingChar = bluetoothSerial.read();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Received: ");
    lcd.setCursor(0, 1);
    lcd.print(incomingChar);
  }
}
```

5. Pair Your Mobile Phone with HC-05:
Enable Bluetooth on your mobile phone.
Scan for devices and pair with the HC-05 module.
6. Upload the Code to Arduino:
Connect your Arduino to your computer using a USB cable.
Select the correct board and port in the Arduino IDE.
Upload the code to your Arduino.
7. Test the Setup:
Send data from your mobile phone via Bluetooth to the HC-05 module.
The received data should be displayed on the LCD.
This is a basic example

LAB TASK 08

// Interfacing Relay module to demonstrate Bluetooth based home automation application.
(using Bluetooth and relay).

/*

Components:

Arduino (or any other microcontroller)
Bluetooth module (HC-05 or HC-06)
Relay module
Power supply for the relay module (if required)
Home appliances or devices to control (e.g., a lamp)
Steps:

Set Up Arduino:

Connect the Arduino to your computer.
Install the necessary drivers and the Arduino IDE.
Connect Bluetooth Module:

Connect the Bluetooth module (HC-05 or HC-06) to the Arduino using the serial pins (TX and RX). Make sure to connect VCC and GND appropriately.
Connect Relay Module:

Connect the relay module to the Arduino. Typically, relay modules have a VCC, GND, and signal pin. Connect the VCC and GND to the appropriate power supply, and connect the signal pin to a digital pin on the Arduino.
Connect Home Appliance to Relay:

Connect the home appliance or device that you want to control to one of the relay outputs on the relay module.
Upload Arduino Code:

Write an Arduino sketch to control the relay based on Bluetooth commands. You can use the SoftwareSerial library to communicate with the Bluetooth module. Here's a basic example:

```
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(10, 11); // RX, TX
```

```
int relayPin = 2; // Digital pin connected to relay module

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

void loop() {
  if (bluetooth.available() > 0) {
    char command = bluetooth.read();
    if (command == '1') {
      digitalWrite(relayPin, HIGH); // Turn on relay
    } else if (command == '0') {
      digitalWrite(relayPin, LOW); // Turn off relay
    }
  }
}
```

Pair Bluetooth Module:

Pair your Bluetooth-enabled device (e.g., smartphone) with the Bluetooth module on the Arduino.
Control Using Bluetooth:

Use a Bluetooth terminal app on your smartphone to send '1' to turn on the relay (and the connected appliance) and '0' to turn it off.
Test:

Test the system by sending commands via Bluetooth and observing the relay's behavior.
Keep in mind that this is a basic example, and you may need to adapt the code and connections based on your specific modules and requirements. Additionally, ensure that you follow safety precautions when working with high-voltage appliances and circuits.

*/

ARDUINO BY HIMANSHU SINGH

LECTURER GOVERNMENT POLYTECHNIC KANPUR