IOT LAB FILE

Submitted by:

Student Name -Mohd Haaris

Roll No. - 20118054

Semester- VII Branch- Information Technology

National Institute of Technology, Raipur

Submitted to: Dr. Chandrashekar Jatoth

Exp:01

LED CONTROL USING ARDUINO BOARD

Aim: To control LED Using Arduino Uno board

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	Led		1
4	12V Adaptor		1
5	Power jack		1
6	USB Cable		1
7	Jumper Wires		Required

Hardware Procedure:

- LED pin is Connected to Arduino Uno pin of 2.
- Power jack is connected to the Arduino Uno.
- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save
6. Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable.

Program:

```
const int led = 2;


void setup() {


  pinMode(led, OUTPUT);



}
```

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

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: LED is successfully controlled by Arduino microcontroller Board.

Exp: 02

POTENTIOMETER AND IR SENSOR INTERFACING WITH ARDUINO

Aim: To Interface Potentiometer and IR Sensor Using Arduino Uno board

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	POT sensor		1
4	IR Sensor		1
5	12V Adaptor		1
6	Power jack		1
7	USB Cable		1
8	Jumper Wires	Required	

Hardware Procedure:

- LED pin is Connected to Arduino Uno pin of 11 & 12.
- POT pin is connected to the Arduino pin A1.
- IR Sensor Pin is connected to the Arduino Pin 4.
- Power jack is connected to the Arduino.

- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save
6. Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable

Program:

```
#define LED_PIN 11

#define POTENTIOMETER_PIN A1

void setup() {

     // put your setup code here, to run once:

   pinMode(4, INPUT);

     pinMode(12, OUTPUT);  // LED

     pinMode(LED_PIN, OUTPUT);

}
```

```
void loop() {
    // put your main code here, to run repeatedly: potentiometer loop
    int potentiometerValue = analogRead(POTENTIOMETER_PIN);
    int brightness = potentiometerValue / 4;
    analogWrite(LED_PIN, brightness);
    // ir loop
    if (digitalRead(4) == LOW) {
        digitalWrite(12, HIGH);
    } else {
        digitalWrite(12, LOW);
    }
}
```

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: Both Analog and Digital Sensors data are successfully measured by Arduino.

Exp: 03

CONTROLLING TWO ACTUATORS USING ARDUINO

Aim: To Interface Actuators Using Arduino Uno board

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	Realys,Battaries,Stepper Motor		2
4	12V Adaptor		1
5	Power jack		1
6	USB Cable		1
7	Jumper Wires		Required

Hardware Procedure:

- Relay 1 pin is connected to Arduino Uno pin 9
- Relay 2 pin is connected to Arduino Uno pin 10
- Power jack is connected to the Arduino.
- Attach the Bluetooth Module.
- USB connector is connected to Arduino Uno to monitor.

- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file 3.Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save 6.Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable.
8. Install Serial Bluetooth Terminal app on mobile phone.
9. Pair your phone with Bluetooth Module and open Bluetooth app then give commands As per the

Program:

```
char data;  // Variable for storing received data void setup()
{
    Serial.begin(9600);   // Sets the baud for serial data transmission
    pinMode(13, OUTPUT);  // Sets digital pin 13 as output pin
    pinMode(12, OUTPUT);  // Sets digital pin 12 as output pin
}
```

```
void loop() {
    if (Serial.available() & gt;
      0)  // Send data only when you receive data:
    {
        data = Serial.read();  // Read the incoming data and store it
                            // into variable data
        Serial.print(data);  // Print Value inside data in Serial
                           // monitor Serial.print(&quot;\n&quot;);
        if (data == &#39;
          0 & #39;) {  // Checks whether value of data is equal to 0
            digitalWrite(13, HIGH);  // If value is 0 then LED at
                                  // 13th pin turns ON
            digitalWrite(12, LOW);        // and 12th pin turns off
        } else if (data == &#39; 1 & #39;) {  // Checks whether value of
                                        // data is equal to 1
            digitalWrite(12, HIGH);  // If value is 1 then LED at
                                  // 12th pin turns ON
            digitalWrite(13, LOW);  // and LED at 13th pin turns OFF
        }
        else {
                    digitalWrite(
                       13, LOW);  // if any other value both LED turns off
                    digitalWrite(12, LOW);
              }
        }
}
```

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: Two Actuators are controlled by smart phone using Bluetooth module.

Exp:04

CREATION OF THINGS SPEAK ACCOUNT

Aim: To create Things Speak account for uploading the sensors data.

Apparatus:

S. No.	APPARATUS RANGE/RATING	QUANTITY
1	Universal Board	1
2	Arduino board	1
3	Any sensor (DHT11)	1
4	LCD	1
5	WIFI Module	1
7	12V Adaptor	1
8	Power jack	1
9	USB Cable	1
10	Jumper Wires	Required

Hardware Procedure:

- LCD pins connected to Arduino Uno pin 2, 3, 4, 5, 6, and 7.
- DHT11 pin connected to the 10 pin of Arduino board.
- Wifi module pins RX & TX are connected to 8 and 9 pin of Arduino (RX = 8, TX = 9).
- USB connector is connected to Arduino Uno to monitor.

- Place Wifi Module in IOT development Board.
- Connect the 12V power supply to development board.
- Power jack is connected to the Arduino Uno.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file 3.Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save 6.Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable.

THINGS SPEAK ACCOUNT CREATION PROCEDURE:

1. First, open ThingsSpeak.com website, and then create an account.
2. And then click on verify on gmail.
3. Login to things speak website and create channel and save it.
4. Go to the API key and then copy “write API key”.
5. And paste that API key in Arduino code
6. After successful code uploading and circuit connections.
7. Open Things Speak account private view, Sensor data will be shown in graph.

Program:

```
#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

#include <dht.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

SoftwareSerial wifi(8, 9);           // TX, RX

String apiKey = "TRNIC1L9BXBXT322";  /// Write API Key dht DHT;

#define DHT11_PIN 10

const int buzzer = 13;

void setup() {

        lcd.begin(16, 2);

        pinMode(buzzer, OUTPUT);

        digitalWrite(buzzer, 0);

        project_Name();

        Serial.begin(9600);

        Serial.println("AT");

        delay(1000);

        Serial.println("AT+CMGF=1");

        delay(1000);

        Serial.println("AT+CNMI=2,2,0,0,0");

        delay(1000);

        lcd.setCursor(0, 0);
        lcd.print("WiFi module");
        lcd.setCursor(0, 1);
        lcd.print("Initilizing      ");
        wifi.begin(115200);
        wifi.println("AT+RST");
        delay(4000);
        wifi.println("AT+CWMODE=3");
        delay(4000);
        wifi.print("AT+CWJAP=");
        wifi.write('"');
        wifi.print("STTMANI");
        wifi.write('"');
        wifi.write(',');
        wifi.write('"');
        wifi.print("hailucky123,./");
        wifi.write('"');
        wifi.println();
        delay(1000);
        lcd.setCursor(0, 0);
        lcd.print("WiFi module  ");
        lcd.setCursor(0, 1);
        lcd.print("Initilized       ");
```

```
    delay(1000);
    lcd.clear();
}
void loop() {
    int chk = DHT.read11(DHT11_PIN);
    // SendWiFi_Data();
    // delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("Temperature:              ");
    lcd.setCursor(0, 1);
    lcd.print("Humidity:        ");
    lcd.setCursor(12, 0);
    lcd.print(DHT.temperature);
    lcd.setCursor(9, 1);
    lcd.print(DHT.humidity);
    delay(500);
    /* Tempurature Data Process*/
    if (DHT.temperature > 45) {
        buzzer_sound();
    }
    /* Humidity Data Process*/
    if (DHT.humidity < 30) {
        buzzer_sound();
    }
    lcd.setCursor(15, 1);
    lcd.write(0x20);
    SendWiFi_Data();
    delay(1000);
}

void SendWiFi_Data() {
    String cmd = "AT+CIPSTART=\"TCP\",\"";
    cmd += "184.106.153.149";  // api.thingspeak.com
    cmd += "\",80";
    wifi.println(cmd);
    delay(1500);
    String getStr = "GET /update?api_key=";
    getStr += apiKey;
    getStr += "&field1=";
    getStr += String(DHT.temperature);
    getStr += "&field2=";
    getStr += String(DHT.humidity);
    getStr += "\r\n\r\n";
    // send data length
    cmd = "AT+CIPSEND=";
    cmd += String(getStr.length());
```

```
        wifi.println(cmd);
        delay(1500);
        wifi.println(getStr);
        delay(1000);
}
void buzzer_sound() {
        digitalWrite(buzzer, HIGH);
        delay(600);
        digitalWrite(buzzer, LOW);
        delay(400);
        digitalWrite(buzzer, HIGH);
        delay(600);
        digitalWrite(buzzer, LOW);
        delay(400);
}
void project_Name() {
        lcd.setCursor(0, 0);
        lcd.print("          ESP8266        ");
        lcd.setCursor(0, 1);
        lcd.print(" Interfacing ");
        delay(3000);
        lcd.clear();}
```

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: Things Speak account will be successfully created. DHT11 Sensor data will be uploaded to cloud.

Exp:05

ACTUATOR CONTROLLING THROUGH CLOUD

Aim: To control the actuator from cloud (thingsspeak.com website).

Apparatus:

S. No.	APPARATUS RANGE/RATING	QUANTITY
1	Universal Board	1
2	Arduino board	1
3	WIFI Module	1
4	12V Adaptor	1
5	Power jack	1
7	USB Cable	1
8	Jumper Wires	Required

Hardware Procedure:

- Relay1 pin is connected to the 12 pin of Arduino board.
- Relay2 pin is connected to the 13 pin of Arduino board.
- Wifi module pins RX & TX are connected to 8 and 9 pin of Arduino (RX = 8, TX = 9).
- USB connector is connected to Arduino Uno to monitor.
- Place Wifi Module in IOT development Board.

- Connect the 12V power supply to development board.
- Power jack is connected to the Arduino Uno.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save
6. Login to ThingsSpeak account and then go to apps and create talkback.
7. Copy those Talkback id and Read API key and paste it on the Arduino Code
8. Give commands from cloud which was given in the code.

10. Click on save & Click on Verify.
11. Click on Upload code into Arduino Uno by using USB cable.
12. Relays are turned on and turned off while giving the commands.

Program:

Precautions:

- Take care about given power supply (12V)
- Jumper wires given carefully whenever given circuit connection

RESULT: Two relays are controlled from cloud by giving the commands.

Exp:06

DHT11SENSOR DATA TO CLOUD

Aim: To Interface DHT11 Using Arduino Uno board and upload sensor data to Cloud..

Apparatus :

S. No.	APPARATUS RANGE/RATING	QUANTITY
1	Universal Board	1
2	Arduino board	1
3	Any sensor (DHT11)	1
4	LCD	1
5	WIFI Module	1
7	12V Adaptor	1
8	Power jack	1
9	USB Cable	1
10	Jumper Wires	Required

Hardware Procedure:

- LCD pins connected to Arduino Uno pin 2,3, 4, 5, 6, and 7.
- DHT11 pin connected to the 10 pin of Arduino board.
- Wifi module pins RX & TX are connected to 8 and 9 pin of Arduino (RX = 8, TX = 9).
- USB connector is connected to Arduino Uno to monitor.

- Place Wifi Module in IOT development Board.
- Connect the 12V power supply to development board.
- Power jack is connected to the Arduino Uno.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save
6. Create an Account in Things Speak, then create a channel.
7. Go to API keys in that Channel and then copy “Write API key”.
8. Go to Arduino code and paste Write API key.
9. Give your mobile hotspot name and password in Arduino code.
10. Click on save & Click on Verify.
11. Click on Upload the code into Arduino Uno by using USB cable.
12. After that open things Speak account and click on private view
13. DHT11 Sensor data will be uploaded and it will be shown as graph in Private view on Things Speak account.

Program:

```
#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

#include <dht.h>

LiquidCrystallcd(2, 3, 4, 5, 6, 7);

SoftwareSerialwifi(8, 9);           // TX, RX

String apiKey = "TRNIC1L9BXBXT322";  /// Write API Key dhtDHT;

#define DHT11_PIN 10

const int buzzer = 13;

void setup() {

      lcd.begin(16, 2);

      pinMode(buzzer, OUTPUT);

      digitalWrite(buzzer, 0);

      project_Name();

      Serial.begin(9600);

      Serial.println("AT");

      delay(1000);

      Serial.println("AT+CMGF=1");

      delay(1000);

      Serial.println("AT+CNMI=2,2,0,0,0");

      delay(1000);

      lcd.setCursor(0, 0);

      lcd.print("WiFi module");
      lcd.setCursor(0, 1);

      lcd.print("Initilizing ");

      wifi.begin(115200);

      wifi.println("AT+RST");

      delay(4000);

      wifi.println("AT+CWMODE=3");

      delay(4000);

      wifi.print("AT+CWJAP=");

      wifi.write('"');

      wifi.print("STTMANI");

      wifi.write('"');

      wifi.write(',');

      wifi.write('"');

      wifi.print("hailucky123,./");

      wifi.write('"');

      wifi.println();

      delay(1000);

      lcd.setCursor(0, 0);

      lcd.print("WiFi module  ");

      lcd.setCursor(0, 1);

      lcd.print("Initilized              ");

      delay(1000);
```

```
  lcd.clear();
}
void loop() {
    int chk = DHT.read11(DHT11_PIN);
    // SendWiFi_Data();
    // delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("Temperature");
    lcd.setCursor(0, 1);
    lcd.print("Humidity");
    lcd.setCursor(12, 0);
    lcd.print(DHT.temperature);
    lcd.setCursor(9, 1);
    lcd.print(DHT.humidity);
    delay(500);
    /* Tempurature Data Process*/
    if (DHT.temperature > 45) {
        buzzer_sound();
    }
    /* Humidity Data Process*/
```

```
     if (DHT.humidity < 30) {
        buzzer_sound();
    }
    lcd.setCursor(15, 1);
    lcd.write(0x20);
    SendWiFi_Data();
    delay(1000);
}
void SendWiFi_Data() {
    String cmd = "AT+CIPSTART=\"TCP\",\"";
    cmd += "184.106.153.149";  // api.thingspeak.com
    cmd += "\",80";
    wifi.println(cmd);
    delay(1500);
    String getStr = "GET /update?api_key=";
    getStr += apiKey;
    getStr += "&field1=";
    getStr += String(DHT.temperature);
    getStr += "&field2=";
    getStr += String(DHT.humidity);
    getStr += "\r\n\r\n";
    // send data length
    cmd = "AT+CIPSEND=";

    cmd += String(getStr.length());

    wifi.println(cmd);

    delay(1500);

    wifi.println(getStr);

    delay(1000);

}

void buzzer_sound() {

    digitalWrite(buzzer, HIGH);

    delay(600);

    digitalWrite(buzzer, LOW);

    delay(400);

    digitalWrite(buzzer, HIGH);

    delay(600);

    digitalWrite(buzzer, LOW);

    delay(400);

}
```

```
void project_Name() {

        lcd.setCursor(0, 0);

        lcd.print("              ESP8266           ");

        lcd.setCursor(0, 1);

        lcd.print(" Interfacing ");

        delay(3000);

        lcd.clear();

}
```

Precautions:

- Take care about given power supply (12V)
- Jumper wires given carefully whenever given circuit connection

RESULT: DHT11 sense the surrounding temperature and measure humidity in surrounding air that temperature and humidity shown by LCD display and Sensor data will be successfully uploaded on Things Speak account.;

Exp:07

IOT BASED AIR POLLUTION CONTROL SYSTEM

Aim: To Interface MQ-7and MQ-135 Sensor Using Arduino Uno board to measure Carbon monoxide and Ammonia gas.

Apparatus:

S. No.	APPARATUS RANGE/RATING	QUANTITY
1	Universal Board	1
2	Arduino board	1
3	MQ7 Sensor	1
4	MQ – 135 Sensor	1
5	12V Adaptor	1
7	Power jack	1
8	USB Cable	1
9	Jumper Wires	Required

Hardware Procedure:

- LCD connected to Arduino Uno pin 2, 3, 4, 5, 6&7.
- MQ - 7 sensor pin is connected to Arduino pin A0.
- MQ – 135 sensor pin is connected to Arduino pin A1.
- Power jack is connected to the Arduino Uno.

- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save
6. Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable.

Program:

```
#include <LiquidCrystal.h>

LiquidCrystallcd(2, 3, 4, 5, 6, 7);

void setup() {

    lcd.begin(16, 2);

    project_Name();

}
```

```
void loop() {
    int mq7_gas_sensor_data = analogRead(A0);
    int mq135_gas_sensor_data = analogRead(A1);
    lcd.setCursor(0, 0);
    lcd.print("MQ7 Data:       ");
    lcd.setCursor(9, 0);
    lcd.print(mq7_gas_sensor_data);
    lcd.setCursor(0, 1);
    lcd.print("MQ135 Data:     ");
    lcd.setCursor(12, 1);
    lcd.print(mq135_gas_sensor_data);
    delay(1000);
}
void project_Name() {
    lcd.setCursor(0, 0);
    lcd.print(" AIR POLLUTION ");
    lcd.setCursor(0, 1);
    lcd.print("CONTROL SYSTEM");
    delay(3000);
    lcd.clear();
}
```

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: MQ – 7 gas sensor detects the Carbon Monoxide gas. And MQ – 135 gas sensor detects Ammonia gas. Whenever the gas value exceeds the threshold value, buzzer starts sounding. And MQ-7 and MQ-135 sensors data also uploaded to cloud successfully.

Exp: 08

TDS SENSOR INTERFACING WITH ARDUINO

Aim: To Interface TDS Sensor Using Arduino Uno board

Apparatus:

S. No.	APPARATUS RANGE/RATING	QUANTITY
1	Universal Board	1
2	Arduino board	1
3	TDS Sensor	1
4	12V Adaptor	1
5	Power jack	1
7	USB Cable	1
8	Jumper Wires	Required

Hardware Procedure:

- LCD pins connected to Arduino Uno pin 2,3,4,5,6,7.
- TDS(Total Dissolved Solids) pin is connected to the Arduino pin A0.
- RELAY pin is connected to the Arduino 8
- Power jack is connected to the Arduino.
- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections
5. Click on Save
6. Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable.

Program:

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: TDS Sensor data was successfully measured with Arduino.

Exp: 09

ACTUATOR CONTROLLINGBY MOBILE USING ARDUINO

Aim: To Interface RGB LED Using Arduino Uno board

Apparatus:

S. No.	APPARATUS RANGE/RATING	QUANTITY
1	Universal Board	1
2	Arduino board	1
3	RGB LED	1
4	12V Adaptor	1
5	Power jack	1
7	USB Cable	1
8	Jumper Wires	Required

Hardware Procedure:

- Actuator pin is connected to Arduino Uno pin 9.
- Power jack is connected to the Arduino.
- Insert Bluetooth Module in Bluetooth Jack.
- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE

2. Click on file

3 .Click on New

4. Write a Program as per circuit Pin connections

5. Click on Save

6. Click on Verify

7 Click on Upload the code into Arduino Uno by using USB cable.

Program:

```
const int Actuator = 9;

void setup() {

    Serial.begin(9600);

    pinMode(Actuator, OUTPUT);

}


void loop() {

    byte brightness;

    if (Serial.available()) {

        brightness = Serial.read();

        Serial.println(brightness);

    }
    if (brightness == 'a')
        digitalWrite(Actuator, HIGH);
    else if (brightness == 'b')
        digitalWrite(Actuator, LOW);
}
```

Precautions:

- Take care about given power supply (12V).
- Jumper wires given carefully whenever given circuit connection.

RESULT: Actuator is controlled by smart phone using Bluetooth module.