# INTERNET OT THINGS LAB MANUAL


***Department of Electronics & Communication Engineering***

## VEMU INSTITUTE OF TECHNOLOGY::P.KOTHAKOTA

**NEAR PAKALA, CHITTOOR-517112**

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

# INTERNET OF THINGS LAB MANUAL


**Name:**_____

**H.T.No:**_____

**Year/Semester:**_____

***Department of Electronics & Communication Engineering***

## VEMU INSTITUTE OF TECHNOLOGY::P.KOTHAKOTA

**NEAR PAKALA, CHITTOOR-517112**

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

## VEMU Institute of Technology
## Dept. of Electronics and Communication Engineering

### Vision of the institute

To be one of the premier institutes for professional education producing dynamic and vibrant force of technocrats with competent skills, innovative ideas and leadership qualities to serve the society with ethical and benevolent approach.

### Mission of the institute

**Mission_1:** To create a learning environment with state-of-the art infrastructure, well equipped laboratories, research facilities and qualified senior faculty to impart high quality technical education.
**Mission_2:** To facilitate the learners to inculcate competent research skills and innovative ideas by Industry-Institute Interaction.
**Mission_3:** To develop hard work, honesty, leadership qualities and sense of direction in learners by providing value based education.

### Vision of the department

To develop as a center of excellence in the Electronics and Communication Engineering field and produce graduates with Technical Skills, Competency, Quality, and Professional Ethics to meet the challenges of the Industry and evolving Society.

### Mission of the department

**Mission_1:** To enrich Technical Skills of students through Effective Teaching and Learning practices to exchange ideas and dissemination of knowledge.
**Mission_2:** To enable students to develop skill sets through adequate facilities, training on core and multidisciplinary technologies and Competency Enhancement Programs.
**Mission_3:** To provide training, instill creative thinking and research attitude to the students through Industry-Institute Interaction along with Professional Ethics and values.

### Programme Educational Objectives (PEOs)

**PEO 1:** To prepare the graduates to be able to plan, analyze and provide innovative ideas to investigate complex engineering problems of industry in the field of Electronics and Communication Engineering using contemporary design and simulation tools.

**PEO-2:** To provide students with solid fundamentals in core and multidisciplinary domain for successful implementation of engineering products and also to pursue higher studies.

**PEO-3:** To inculcate learners with professional and ethical attitude, effective communication skills, teamwork skills, and an ability to relate engineering issues to broader social context at work place

### Programme Outcomes(Pos)

| | |
|---|---|
| **PO_1** | **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |

| | |
|---|---|
| **PO_2** | **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **PO_3** | **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| **PO_4** | **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO_5** | **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| **PO_6** | **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| **PO_7** | **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO_8** | **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **PO_9** | **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO_10** | **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| **PO_11** | **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **PO_12** | **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

**Programme Specific Outcome(PSOs)**

| | |
|---|---|
| **PSO_1** | **Higher Education :** Qualify in competitive examination for pursuing higher education by applying the fundamental concepts of Electronics and Communication Engineering domains such as Analog & Digital Electronics, Signal Processing, Communication & Networking, Embeded Systems, VLSI Design and Control systems etc., |
| **PSO_2** | **Employment**: Get employed in allied industries through their proficiency in program specific domain knowledge, Specalized software packages and Computer programming or became an entrepreneur. |

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

## IV B.Tech. I-Sem (ME)
## (19A05406P) INTERNET OF THINGS LABORATORY

### COURSE OUTCOMES(COs)

| CO No. | Description | Blooms Level |
|---|---|---|
| | | |
| | | |
| | | |

**Practicals:**

1. Select any one development board (Eg., Arduino or Raspberry Pi) and control LED using the board.
2. Using the same board as in (1), read data from a sensor. Experiment with both analog and digital sensors.
3. Control any two actuators connected to the development board using Bluetooth.
4. Read data from sensor and send it to a requesting client. (using socket communication)

Note: The client and server should be connected to same local area network.

5. Create any cloud platform account, explore IoT services and register a thing on the platform.
6. Push sensor data to cloud.
7. Control an actuator through cloud.
8. Access the data pushed from sensor to cloud and apply any data analytics or visualization services.
9. Create a mobile app to control an actuator.
10. Design an IoT based air pollution control system which monitors the air pollution by measuring carbon monoxide, ammonia, etc and gives alarm or sends message when the pollution level is more than permitted range.
11. Design an IoT based system which measures the physical and chemical properties of the water and displays the measured values.
12. Identify a problem in your local area or college which can be solved by integrating the things you learned and create a prototype to solve it (Mini Project).
13. Design a business model canvas for a digital display

**Course outcomes:**

At the end of the course, students will be able to

- Choose the sensors and actuators for an IoT application (L1)
- Select protocols for a specific IoT application (L2)
- Utilize the cloud platform and APIs for IoT application (L3)
- Experiment with embedded boards for creating IoT prototypes (L3)
- Design a solution for a given IoT application (L6)

# VEMU INSTITUTE OF TECHNOLOGY::P.KOTHAKOTA

NEAR PAKALA, CHITTOOR-517112
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)
**Dept. of Electronics and Communication Engineering**

## (19A05406P) INTERNET OF THINGS LABORATORY

## IV B.Tech. I-Sem (ME)

## LIST OF EXPERIMENTS TO BE CONDUCTED

1. Led Control Using Arduino Board
2. Potentiometer And Ir Sensor Interfacing With Arduino
3. Controlling Two Actuators Using Arduino
4. Creation of Things Speak Account
5. Actuator Controlling Through Cloud
6. Dht11sensor Data To Cloud
7. Iot Based Air Pollution Control System
8. Tds Sensor Interfacing With Arduino
9. Actuator Controllingby Mobile Using Arduino

# CONTENTS

# DOS & DONTS IN LABORATORY

## DO's

1. Students should be punctual and regular to the laboratory.
2. Students should come to the lab in-time with proper dress code.
3. Students should maintain discipline all the time and obey the instructions.
4. Students should carry observation and record completed in all aspects.
5. Students should be at their concerned experiment table, unnecessary moment is restricted.
6. Students should follow the indent procedure to receive and deposit the components from lab technician.
7. While doing the experiments any failure/malfunction must be reported to the faculty.
8. Students should check the connections of circuit properly before switch ON the power supply.
9. Students should verify the reading with the help of the lab instructor after completion of experiment.
10. Students must endure that all switches are in the lab OFF position, all the connections are removed.
11. At the end of practical class the apparatus should be returned to the lab technician and take back the indent slip.
12. After completing your lab session SHUTDOWN the systems, TURNOFF the power switches and arrange the chairs properly.
13. Each experiment should be written in the record note book only after getting signature from the lab in charge in the observation notebook.

## DON'Ts

1. Don't eat and drink in the laboratory.
2. Don't touch electric wires.
3. Don't turn ON the circuit unless it is completed.
4. Avoid making loose connections.
5. Don't leave the lab without permission.
6. Don't bring mobiles into laboratory.
7. Do not open any irrelevant sites on computer.
8. Don't use a flash drive on computers.

# SCHEME OF EVALUATION

| S.No | Program | Date | Marks Awarded | | | | Total 30(M) |
|---|---|---|---|---|---|---|---|
| | | | Record (10M) | Obs. (10M) | Viva (5M) | Attd. (5M) | |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| ADVANCED EXPERIMENTS | | | | | | | |
| 1 | | | | | | | |
| 2 | | | | | | | |

**Signature of Lab In-charge**

**Exp:01** **Date:**

# 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.

## Conclusion:

## Viva questions:

**Exp: 02** **Date:**

## 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 | | |
| 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.

**Conclusions:**

**Viva Questions:**

**Exp: 03** **Date:**

## 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 Code.

**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 == '1'){ // 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 Actuatorsare controlled by smart phone using Bluetooth module.

**Conclusions:**

**Viva Questions:**

**Exp:04** **Date:**

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

**Conclusion:**

.

**Viva questions:**

**Exp:05** **Date:**

## 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:

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

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


int ch,ch1,mode=1;


const int relay1=12;
const int relay2=13;
int i;



void setup(){
wifi.begin(115200);
Serial.begin(115200);

pinMode(relay1,OUTPUT);
pinMode(relay2,OUTPUT);

digitalWrite(relay1,LOW);
digitalWrite(relay2,LOW);
delay(100);
```

```
Serial.println("WiFi Module initilizing.....");
wifi.println("AT+RST");
delay(4000);
wifi.println("AT+CWMODE=3");//AT+CWJAP="SSID","PASWD"
delay(4000);

wifi.print("AT+CWJAP=");
wifi.write("");
wifi.print("STTMANI2");
wifi.write("");
wifi.write(',');

wifi.write("");
wifi.print("hailucky123,./");
wifi.write("");
wifi.println();
delay(2000);
}



void loop(){
String  cmd = "AT+CIPSTART=\"TCP\",\"";
cmd += "184.106.153.149"; // api.thingspeak.com
cmd += "\",80";
wifi.println(cmd);
delay(1000);

String
```

```
getStr ="GET /talkbacks/47047/commands/execute?api_key=5G9L2SELAH60VR3F\r\n\r\n";

  // send data length
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
wifi.println(cmd);

wifi.end();
delay(1000);
wifi.begin(115200);
delay(1500);
wifi.println(getStr);

 for(i=0;i<=1000;i++)
 {
  if(wifi.available())
  {
ch=wifi.read();
Serial.write(ch);
  }
 }

delay(1200);
 for(i=0;i<=1000;i++)
 {
  if(wifi.available())
  {
ch=wifi.read();
  if(ch==':')
```

```
gotoxx;
  ch1=ch;
Serial.write(ch);
  }
 }
xx: ch1=wifi.read();

if(mode==1)
{
  if(ch1=='1')
  {
digitalWrite(relay1,HIGH);
Serial.println("relay1 ON");
  }

   if(ch1=='2')
  {
digitalWrite(relay1,LOW);
Serial.println("relay1 OFF");
  }
   if(ch1=='3')
  {
digitalWrite(relay2,HIGH);
Serial.println("relay2 ON");
  }
   if(ch1=='4')
  {
digitalWrite(relay2,LOW);
Serial.println("relay2 OFF");
  }
```

}

**}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.

**Conclusion:**

**Viva questions:**

**Exp:06** **Date:**

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

**Conclusion:**

**Viva questions:**

**Exp:07** **Date:**

# IOT BASED AIR POLLUTION CONTROL SYSTEM

**Aim:** To Interface MQ-7and MQ-135 Sensor Using ArduinoUno boardto 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** **Date:**

## TDS SENSOR INTERFACING WITH ARDUINO

**Aim:** To Interface TDS SensorUsing 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:**

```
#include<LiquidCrystal.h>

int relay = 8;
const int TDS_sensor = 9;


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



void setup()
{
 lcd.begin(16, 2);
 pinMode(relay, OUTPUT);
 digitalWrite(relay, LOW);
 project_Name();
}

void loop()
{
 int TDS_sensor_data = analogRead(A0);
 lcd.setCursor(0,0);
 lcd.print("TDS Data:       ");
 lcd.setCursor(0,1);
 lcd.print("                ");
 lcd.setCursor(9,0);
 lcd.print(TDS_sensor_data);
 if(TDS_sensor_data >150)
 digitalWrite(relay, HIGH);
 else
 digitalWrite(relay, LOW);
 delay(500);

void project_Name(){
 lcd.setCursor(0,0);
 lcd.print("TDS SENSOR      ");
 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:**TDS Sensor data was successfully measuredwith Arduino.

**Conclusion:**

**Viva questions:**

**Exp: 09** **Date:**

# ACTUATOR CONTROLLINGBY MOBILE USING ARDUINO

**Aim:** To Interface RGB LEDUsingArduinoUno 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.