**Experiment 1 : Make the LED glow and fade using Arduino UNO**

**Aim:** The principal aim of this experiment is to interface a LED with the microcontroller and to Make LED glow using Arduino Uno.

**1.1.COMPONENTS REQUIRED**

1. Arduino UNO
2. Breadboard
3. Jumper wires
4. 100 Ohm resistors

a. ARDUINO UNO:

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

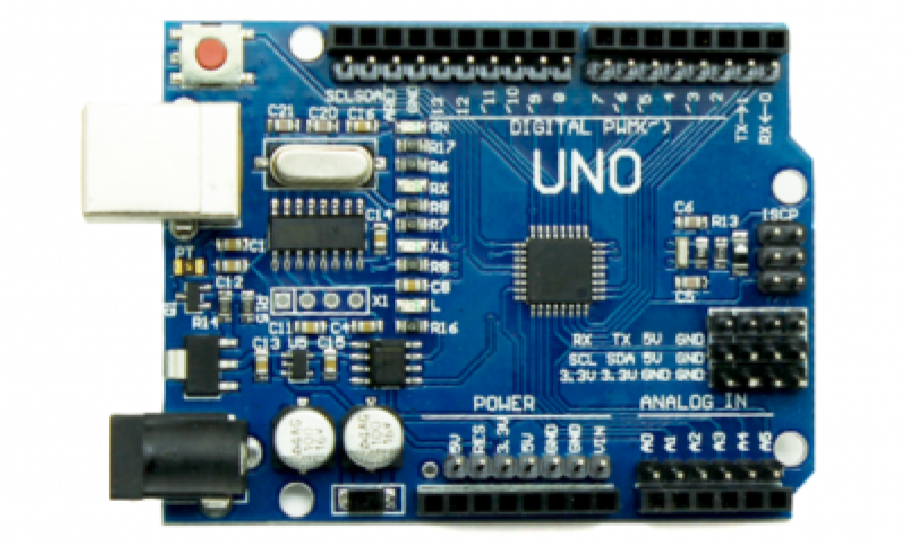


Figure 1.0 - Arduino UNO

b. BREADBOARD:

Breadboards are one of the most fundamental pieces when learning how to build circuits. Breadboards are commonly utilized while prototyping temporary circuits. It is useful to designers because it allows components to be removed and replaced easily.

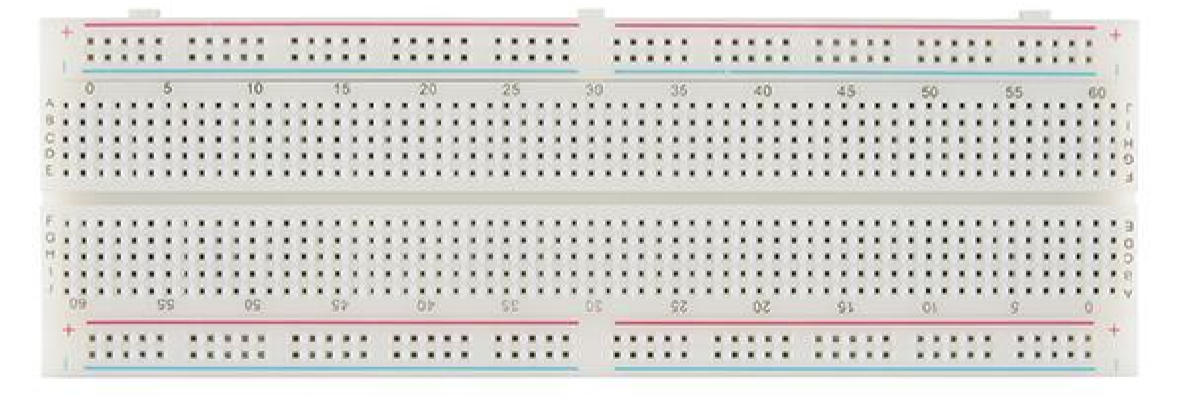


Figure 1.1 – Breadboard

**1.2.SOFTWARE**

Software is a generic term to refer to the scripts and programs that run on a microprocessor or microcontroller and execute specific tasks.

2.1 GET START WITH ARDUINO IDE

Follow the steps to install Arduino IDE:

Step 1: Browse for the URL - ' https://www.arduino.cc/en/software '

Step 2: In DOWNLOAD OPTIONS, choose Windows/Linux/Mac OS accordingly.

Step 3: Select - JUST DOWNLOAD. The download will start!

Step 4: Run the downloaded setup file.

**1.3. PROGRAM to implement Blinking LED Using Arduino**

void setup() {

pinMode(13,OUTPUT);

}

void loop() {

digitalWrite(13,HIGH);

delay(1000);

digitalWrite(13,LOW);

delay(500);

}

**1. 4**. **Results :**

Blinking LED Using Arduino is successfully implemented

**Experiment 2 : Arduino UNO - Read digital and analog signal from a sensor module**

**Aim:** The principal aim of this experiment is to interface a sensor with the microcontroller and to experiment with its analog and digital values..

**2.1. COMPONENTS REQUIRED**

1. Arduino UNO
2. Breadboard
3. soil moisture sensor
4. Jumper wires

a. ARDUINO UNO:

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

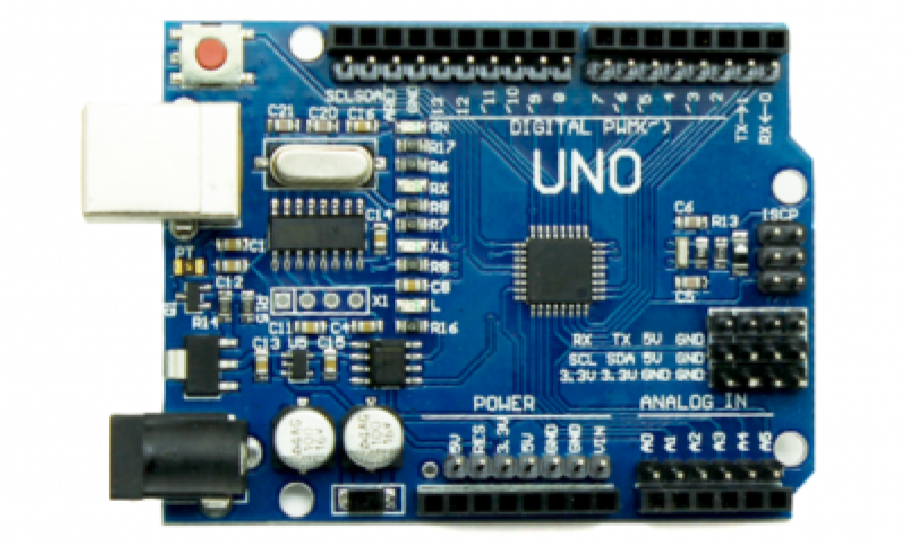


Figure 1.0 - Arduino UNO

b. BREADBOARD:

Breadboards are one of the most fundamental pieces when learning how to build circuits. Breadboards are commonly utilized while prototyping temporary circuits. It is useful to designers because it allows components to be removed and replaced easily.

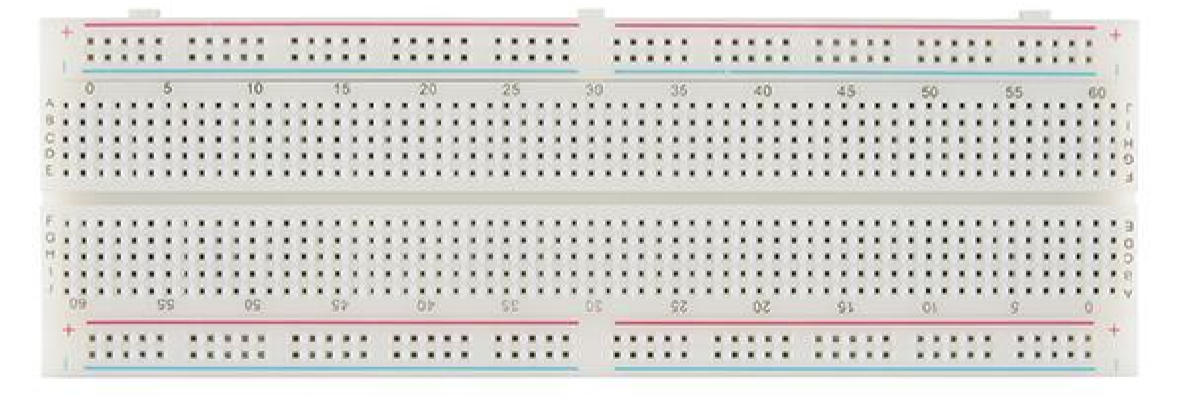
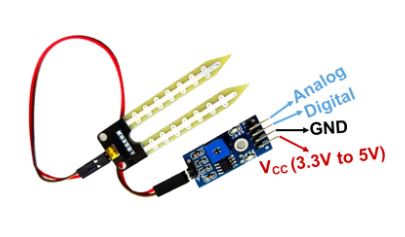


Figure 1.1 – Breadboard

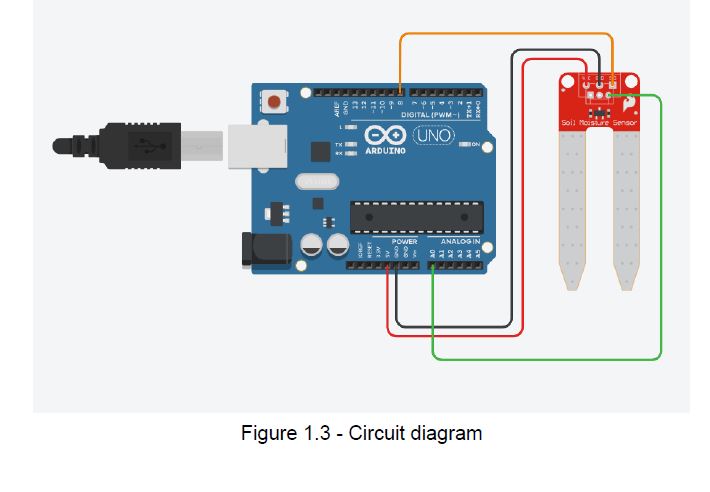
**c. soil moisture sensor :**

The soil moisture sensor is the first thing that springs to mind when it comes to building your smart irrigation system or automatic plant watering system. With this sensor in place and a little Arduino support, we can design a system that can water your plants when it's needed, avoiding overwatering and under watering.



**d.** **WIRING**

**●** Connect the “vcc” of the sensor to the “5v” of the Arduino UNO, GND to GND pin of the microcontroller, D0 pin to the D8 pin of the microcontroller, and A0 to the A0 pin of the microcontroller..



**2.2.SOFTWARE**

Software is a generic term to refer to the scripts and programs that run on a microprocessor or microcontroller and execute specific tasks.

2.1 GET START WITH ARDUINO IDE

Follow the steps to install Arduino IDE:

Step 1: Browse for the URL - ' https://www.arduino.cc/en/software '

Step 2: In DOWNLOAD OPTIONS, choose Windows/Linux/Mac OS accordingly.

Step 3: Select - JUST DOWNLOAD. The download will start!

Step 4: Run the downloaded setup file.

**2.3. PROGRAM to implement Read digital and analog signal from a sensor module.**

void setup() {

 pinMode(8,INPUT);

Serial.begin(9600);

}

void loop() {

  int soil\_digital\_value = digitalRead(8);

  int analog\_value = analogRead(A0);

Serial.print(soil\_digital\_value);

Serial.print("  ");

Serial.println(analog\_value);

**2.4. Results :**

Read digital and analog signal from a sensor module using Arduino UNO is successfully implemented.

**Experiment 3 : Build a mobile application for controlling the peripheral devices via Bluetooth**

**Aim:**

The aim of this Experiment is to Build a mobile application for controlling the peripheral devices via Bluetooth

**3.2.SOFTWARE**

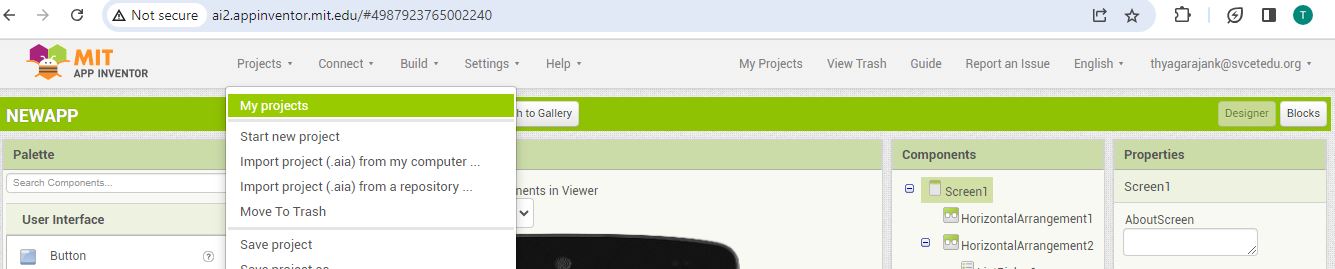
MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology

**2.1 GET START WITH MIT APP Inventor**

Follow the steps to Login MIT APP Inventor:

**Step 1:** Browse for the URL - ' https://appinventor.mit.edu/'

**Step 2:** Go to App Inventor and open a project (or create a new one — use Project > Start New Project and give your project a name)



**Step 3:** Use a horizontal arrangement component to display this component from left to right in the Mobile Display.

**Step 4:** Go to Properties Set *AlignHorizontal(*option).In this the choices are: **1 = left aligned**, 2 = right aligned, 3 = horizontally cantered. Alignment has no effect if the Horizontal Arrangement’s Width is Automatic.

**Step 5:** Go to Properties Set **AlignVertical.** **The choices are: 1 = aligned at the top**, **2 =** vertically centered, 3 = aligned at the bottom. Alignment has no effect if the HorizontalArrangement’s Height is Automatic.

**Step 6:** Specifies the background color of the HorizontalArrangement as an alpha-red-green-blue integer.

**Step 7:** Specifies the HorizontalArrangement’s vertical height, measured in100 pixels.

**Step 8 :** Specifies the horizontal width of the HorizontalArrangement, measured in 100 percent.

**Step 9:** repeat the step 3 to 8

**Step 10 :** In User Interface – Select -ListPicker -Is a button that, when clicked on,

displays a list of texts for the user to choose among.

**Step 11 :** Go to Properties Specifies the ListPicker’s background color as an alpha-red-green-blue integer.

**Step 12 :**  Specifies the text displayed by the ListPicker as SCAN FOR DEVICE and **TextAlignment as** 1 (center)

**Step 13:** Specifies the text color of the ListPicker as an alpha-red-green-blue integer.

**Step 14:** repeat the step 3 to 8

**Step 15:** In User Interface – Select-Button-Button with the ability to detect clicks

**Step 16 :** Go to Properties Specifies the Button background color as an alpha-red-green-blue integer.

**Step 17 :** Specifies the text displayed by the Button as TURN ON and **TextAlignment as** 1 (center)

**Step 18:** In User Interface – Select-Button- Go to Properties Specifies the Button background color as an alpha-red-green-blue integer.

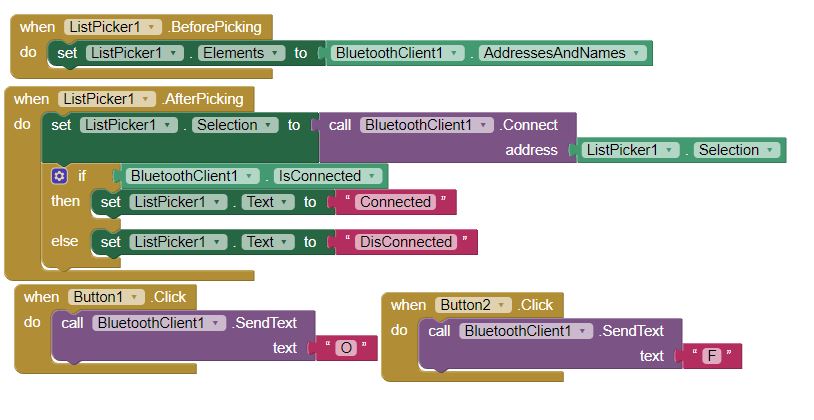
**Step 19 :** Specifies the text displayed by the Button as TURN OFF and **TextAlignment as** 1 (center)

**Step 20 :** Select Connectivity -Drag and drop in Mobile display Its Use BluetoothClient to connect your device to other devices using Bluetooth.

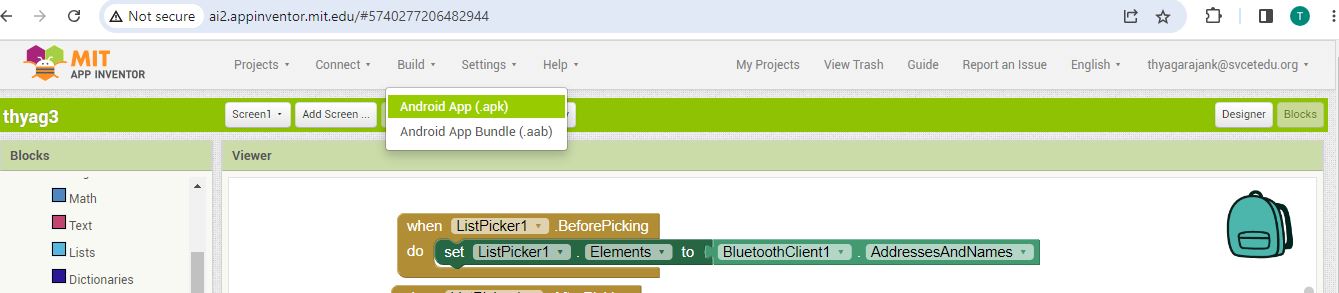
**User Interface in Mobile :**

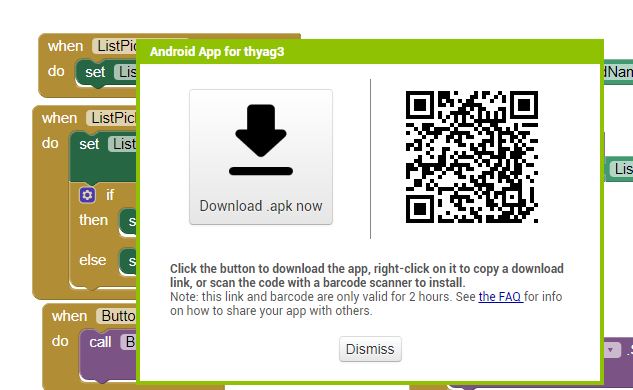


Code: Go to Blocks and Drag and drop the respective code as give bellow.



**Step 21 :** Go to Build option Select Android APP(.apk) which generates Android app and it need to be downloaded and installed in the mobile phone.





**Result:**

Fetched the data successfully from the sensor using the microcontroller unit in both analog and digital form.

**Experiment 4:** **write an Arduino program to turn LED ON and OFF using Push Button with Arduino Uno.**

**INTERFACING PUSH BUTTON**

**AIM:** To write an Arduino program to turn LED ON and OFF using Push Button with Arduino Uno.

**EQUIPMENTS REQUIRED:**

1. PC

2. Arduino IDE

3. LED

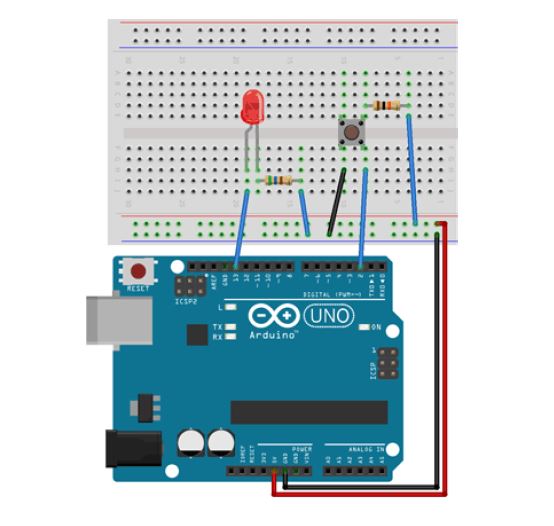
4. Bread Board-1No

5. Resistor-560Ω, 10K Ω -1No

6. Push Button-1No

7. Wires

**CIRCUIT DIAGRAM:**



1. Digital pin 13 pin to LED positive
2. Ground (GND) to LED Negative
3. Digital 2 pin to PUSS BUTTON Negative
4. 5V to PUSS BUTTON Negative
5. PUSH BUTTON Positive to LED Negative

**PROCEDURE:**

1. Connect the circuit as per the circuit diagram.

2. Connect Arduino to your PC.

3. Open the Arduino IDE in computer and write the program.

4. Compile the program for any errors and upload it to the Arduino.

5. Observe the output LED ON when pressing Push Button.

**PROGRAM:**

// Turns on and off a light emitting diode(LED) connected to digital

pin 13, when pressing a pushbutton attached to pin 2.

// set pin numbers:

**const int buttonPin = 2;** // the number of the pushbutton pin

**const int ledPin = 13;** // the number of the LED pin // variables will change:

**int buttonState = 0**; // variable for reading the pushbutton status

**void setup() {**

// initialize the LED pin as an output:

**pinMode(ledPin, OUTPUT);**

// initialize the pushbutton pin as an input:

**pinMode(buttonPin, INPUT);**

**}**

**void loop() {**

// read the state of the pushbutton value:

**buttonState = digitalRead(buttonPin);**

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

**if (buttonState == HIGH) {**

// turn LED on:

**digitalWrite(ledPin, HIGH);**

**} else {**

// turn LED off:

**digitalWrite(ledPin, LOW);**

**}**

**}**

**RESULT:** Thus the LED is switched ON, OFF by Push Button using Arduino Uno is successfully implemented.

**Experiment 5: Interface a sensor with the microcontroller and transmit the data to cloud**

**Aim :** The aim of this Experiment is to send the sensor data from the ESP8266 Wi-Fi module to the Thing Speak Cloud Platform and analyze the data using graphs and charts provided by Thing Speak.

**MODULE 2 - HARDWARE**

Hardware refers to any physical components/particulars of a system containing ICs, electronics, sensors, and circuit boards. Without hardware, an IoT system cannot exist, and the software developed won't be able to run.

**2.1 COMPONENTS REQUIRED**

● NodeMCU ESP8266

● USB Data cable

● Soil moisture sensor

● Breadboard

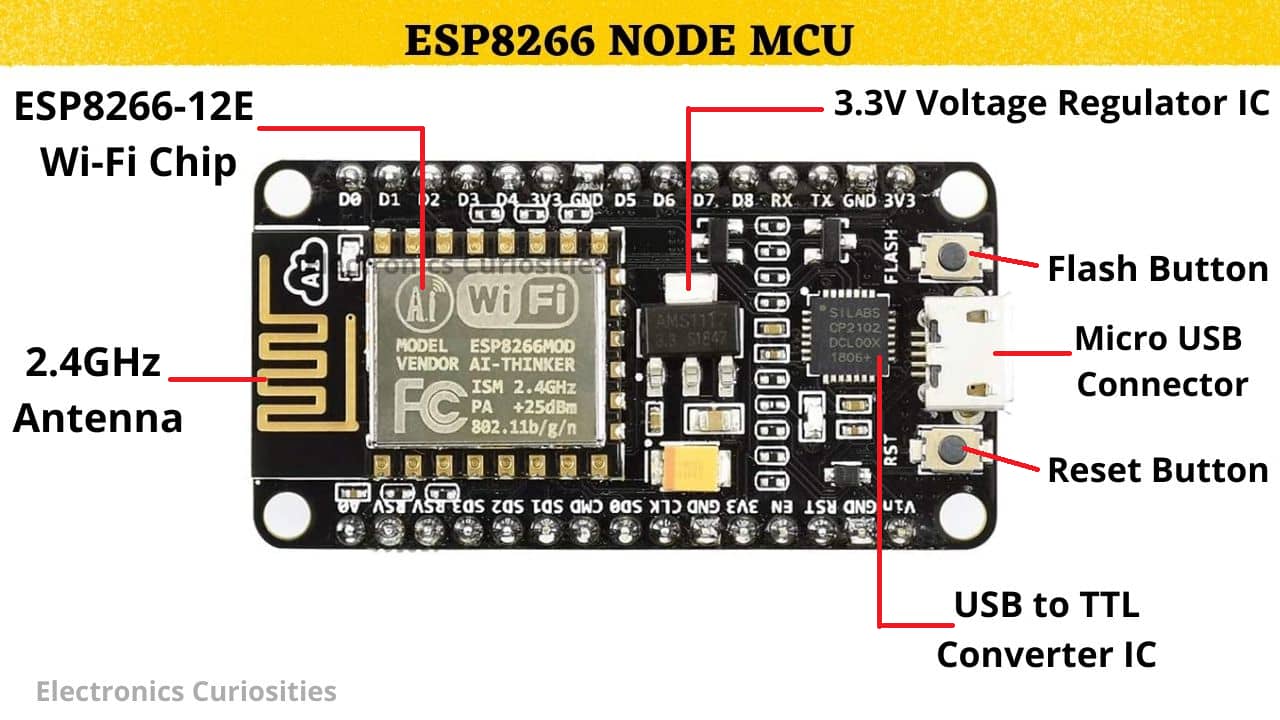
● Jumper wires

**2.2 ANNEXURE**

A. NodeMCU ESP8266:

The Node MCU ESP8266 is a popular WiFi development board that is based on the

ESP8266 WiFi module. It features a microcontroller, flash memory, and a built-in WiFi module, which makes it ideal for creating IoT devices that require wireless connectivity. The Node MCU ESP8266 is programmable using the Arduino IDE, Lua scripting language, or other programming languages such as Micro Python.



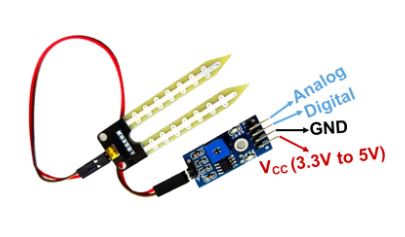
**B. BREADBOARD:**

Breadboards are one of the most fundamental pieces when learning how to build

circuits. Breadboards are commonly utilized while prototyping temporary circuits. It is useful to designers because it allows components to be removed and replaced easily.

**C. SOIL MOISTURE SENSOR**:

A soil moisture sensor is a device designed to measure the amount of moisture present in soil. It is typically made up of two metal probes that are inserted into the soil, with a small electronic circuit in between them. When the probes come into contact with the soil, the electronic circuit measures the electrical resistance between the probes. The resistance is inversely proportional to the moisture content of the soil, which means that as the soil becomes drier, the resistance between the probes increases.



**2.3 WIRING**

1.Connect the VCC of the sensor to "VCC" of ESP8266 module -3V,

2 The GND OF the sensor to the "G" of the ESP8266 module and

3.The signal (ANALONG) pin to the "A0" of the Node MCU

**MODULE 3 - SOFTWARE**

Software is a generic term to refer to the scripts and programs that run on a microprocessor or microcontroller and execute specific tasks.

**3.1 GET START WITH ARDUINO IDE**

To set up the Arduino IDE for using ESP8266, follow these steps:

● Download and install the latest version of the Arduino IDE from the official website: https://www.arduino.cc/en/software.

● Open the Arduino IDE and go to File > Preferences.

● In the Preferences window, find the "Additional Boards Manager URLs" field and add the following URL: http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

● Click "OK" to close the Preferences window.

● Next, go to Tools > Board > Boards Manager. In the Boards Manager, search for "esp8266" and install the "esp8266 by ESP8266 Community" package.

● Once the installation is complete, go back to Tools > Board and select "NodeMCU 1.0(ESP-12E Module)" or the appropriate board that you are using.

● Finally, connect your ESP8266 board to your computer using a USB cable and select the appropriate port from the Tools > Port menu.

You are now ready to start programming your ESP8266 board using the Arduino IDE!

**3.2 PROGRAM**

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include "ThingSpeak.h"

const char\* ssid = "ssid"; //WiFi SSID

const char\* pass = "123123123"; //WiFi Passcode

WiFiClient client;

unsigned long channelNumber = 2074614; // Channel ID

const char\* APIwriteKey = "OT1FHJAES32JHLS4"; // API WriteKey

int data = 0;

void setup() {

Serial.begin(9600);

Serial.println("Connecting to the network");

WiFi.begin(ssid,pass);

while(WiFi.status() != WL\_CONNECTED){

Serial.print(".");

delay(150);

}

Serial.println();

Serial.println("Connected to the network");

ThingSpeak.begin(client);

}

void loop() {

data = data + 5;

ThingSpeak.setField(1, data);

ThingSpeak.writeFields(channelNumber, APIwriteKey);

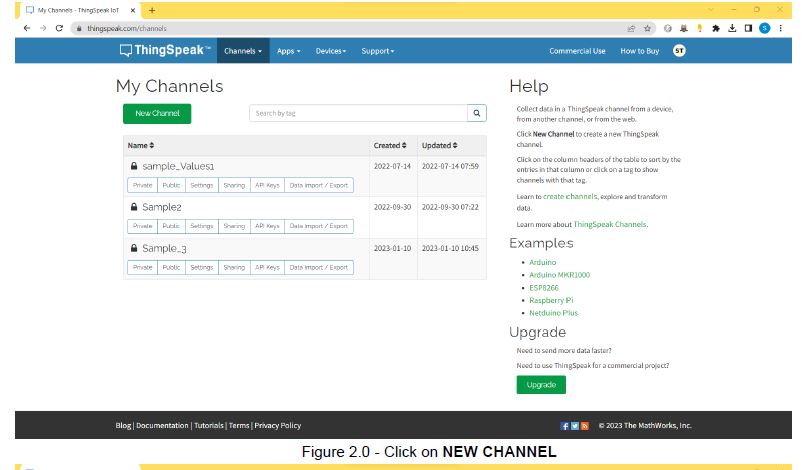
delay(15000);

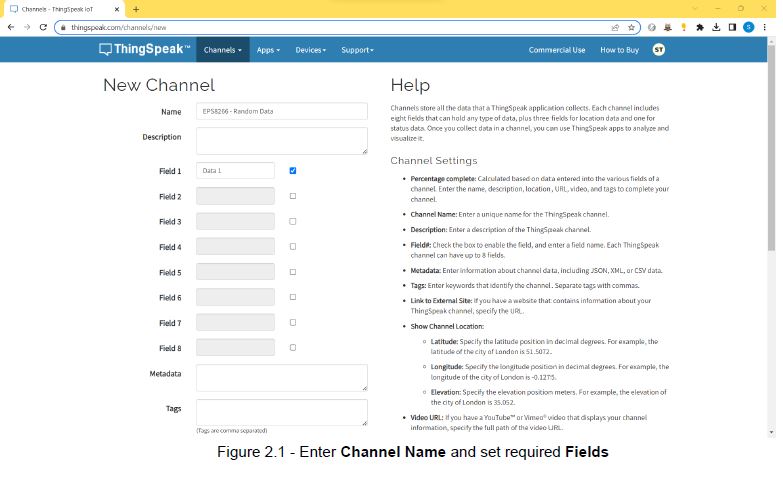
}

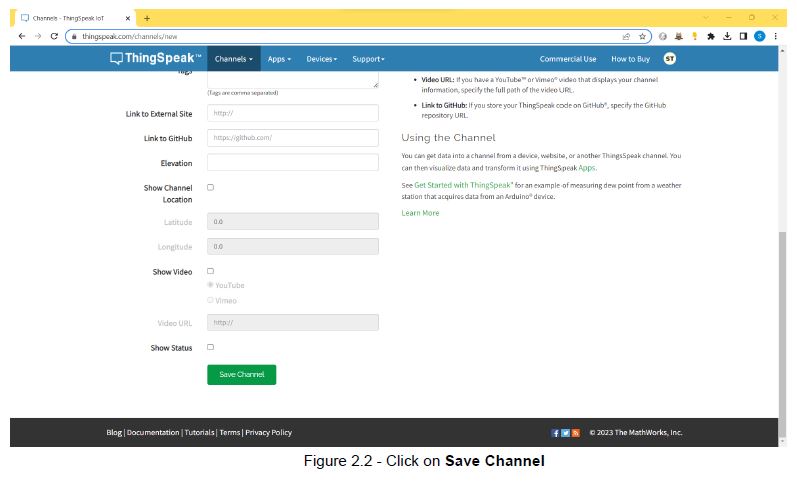
**3.3 THINGSPEAK CLOUD PLATFORM SET-UP**

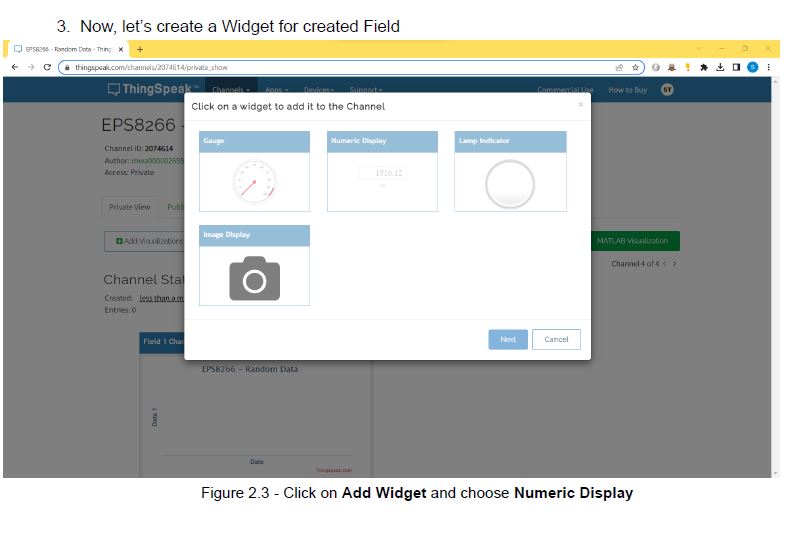
1. Go to https://thingspeak.com/ and create an account.

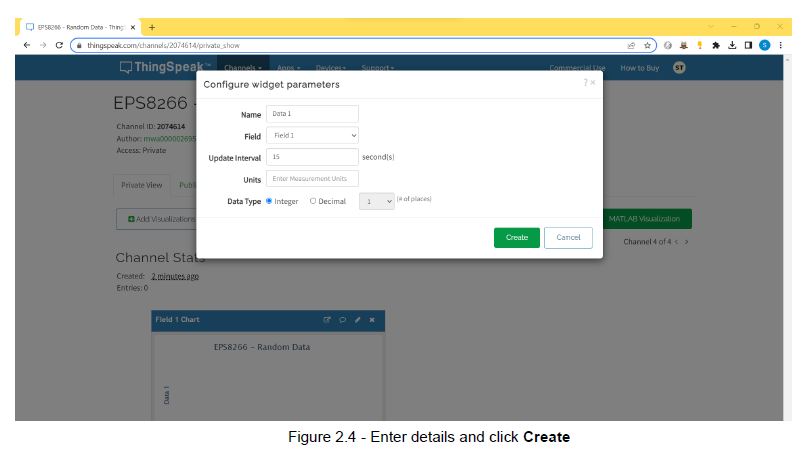
2. After creating the account, create a new channel. In this channel, we will store the data sent by the ESP8266.

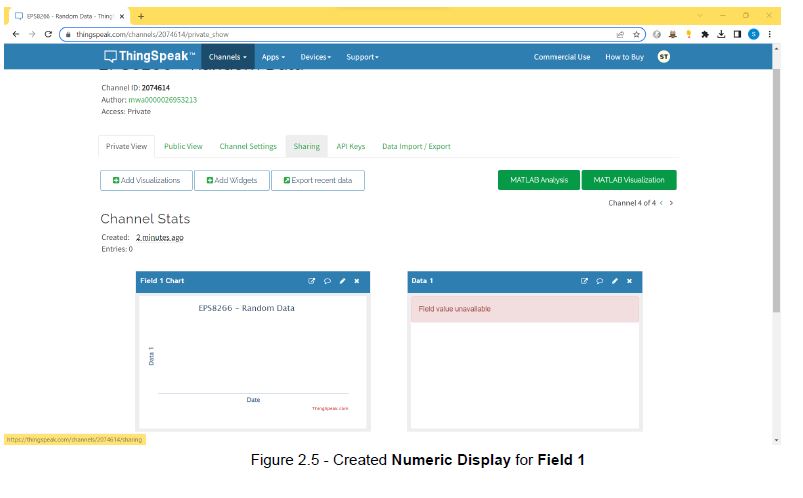


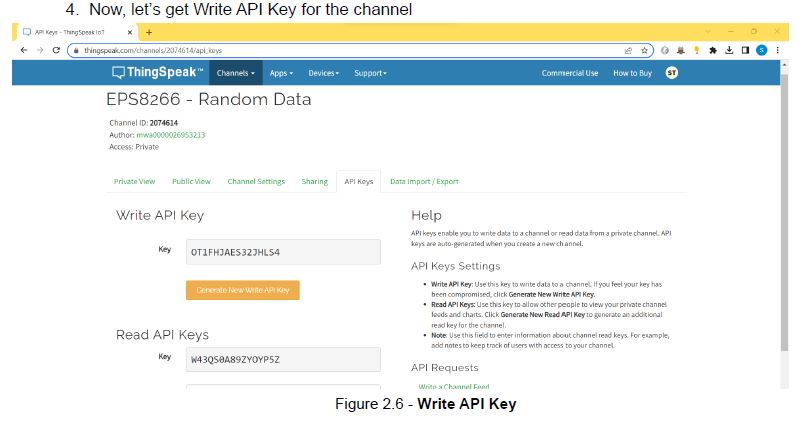












**MODULE 4 - METHODOLOGY**

● Set-up Thing Speak Cloud Platform

● Install the Arduino IDE software and set-up the environment for ESP8266 module

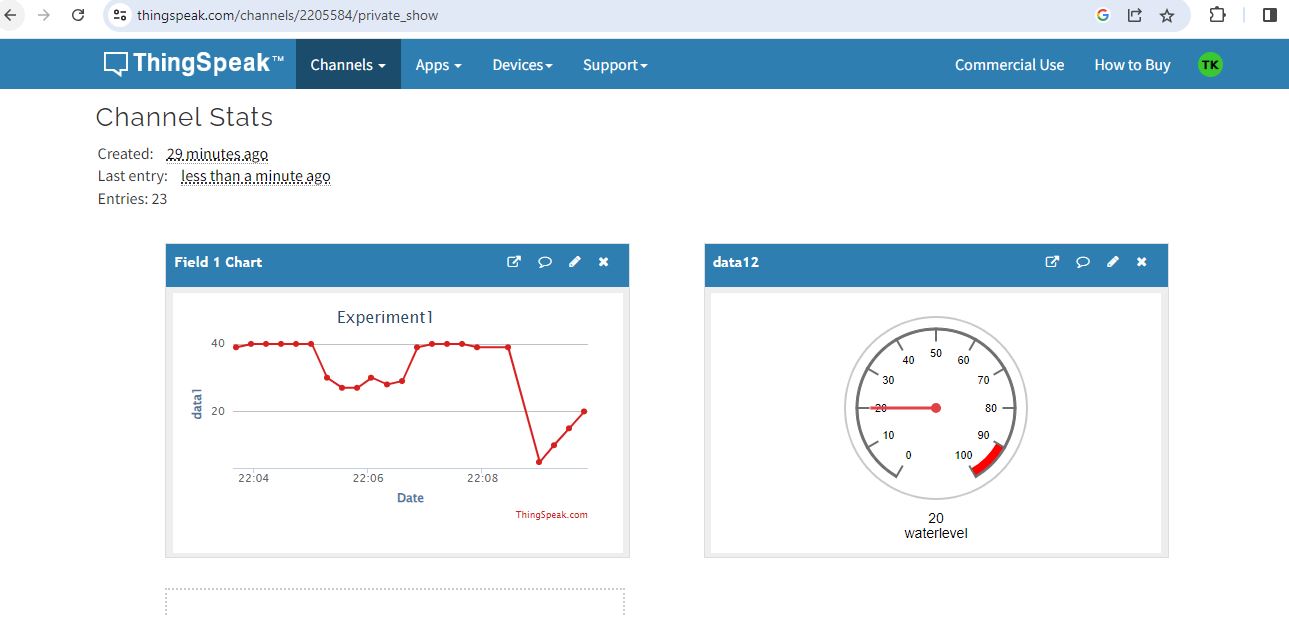
● Write the program and save the file

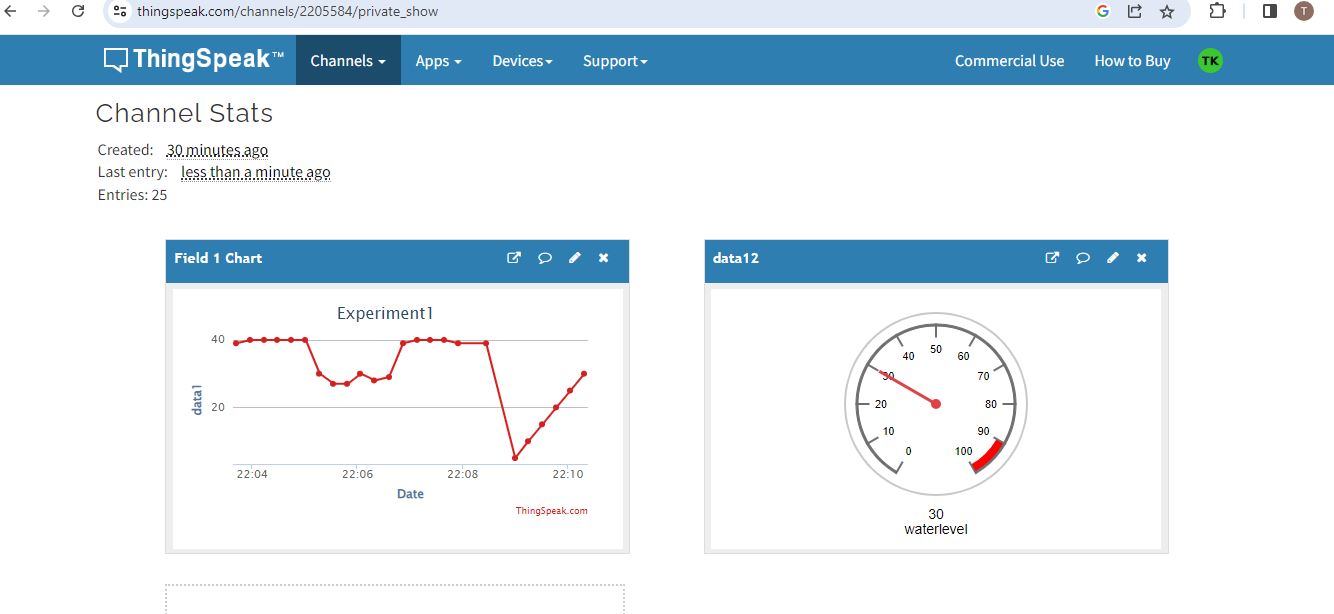
● Select respective Board and Port

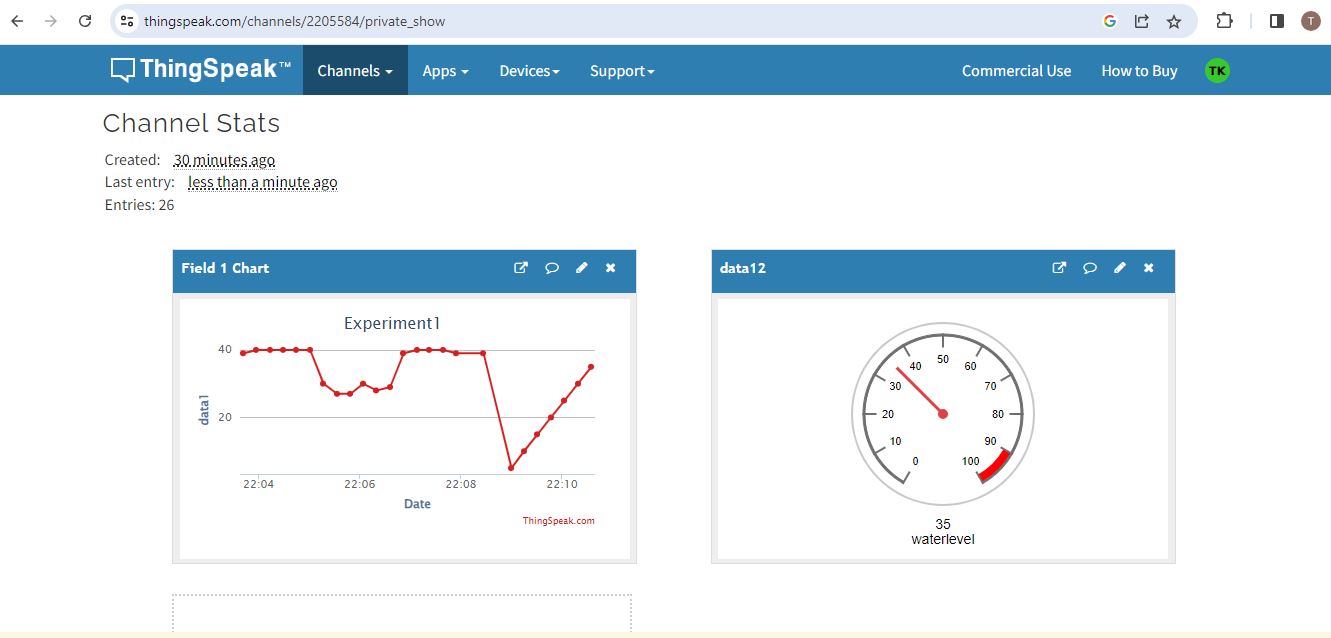
● Finally, upload the code to the microcontroller

**MODULE 5 - RESULT**

This Experiment has demonstrated how to send the sensor data from the ESP8266 Wi-Fi module to the Thing Speak Cloud Platform.







**Experiment 6. set-up a local server using the ESP8266 WiFi module.**

**1.1 AIM**

The aim of this Experiment is to set-up a local server using the ESP8266 WiFi module.

**MODULE 1 - HARDWARE**

Hardware refers to any physical components/particulars of a system containing ICs, electronics,

sensors, and circuit boards. Without hardware, an IoT system cannot exist, and the software

developed won't be able to run.

**2.1 COMPONENTS REQUIRED**

● Node MCU ESP8266

● USB Data cable

**2.2 ANNEXURE**

**A. Node MCU ESP8266:**

The Node MCU ESP8266 is a popular WiFi development board that is based on the ESP8266 WiFi module. It features a microcontroller, flash memory, and a built-in WiFi module, which makes it ideal for creating IoT devices that require wireless connectivity. The Node MCU ESP8266 is programmable using the Arduino IDE, Lua scripting language, or other programming languages such as Micro Python.

2.3 WIRING : Connect the ESP8266 Node MCU with the system using USB type B data cable

**MODULE 2 - SOFTWARE**

Software is a generic term to refer to the scripts and programs that run on a microprocessor or

microcontroller and execute specific tasks.

**2.1 GET START WITH ARDUINO IDE**

To set up the Arduino IDE for using ESP8266, follow these steps:

● Download and install the latest version of the Arduino IDE from the official website:

https://www.arduino.cc/en/software.

● Open the Arduino IDE and go to File > Preferences.

● In the Preferences window, find the "Additional Boards Manager URLs" field and add the

following URL: http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

● Click "OK" to close the Preferences window.

● Next, go to Tools > Board > Boards Manager. In the Boards Manager, search for

"esp8266" and install the "esp8266 by ESP8266 Community" package.

● Once the installation is complete, go back to Tools > Board and select "NodeMCU 1.0

(ESP-12E Module)" or the appropriate board that you are using.

● Finally, connect your ESP8266 board to your computer using a USB cable and select the

appropriate port from the Tools > Port menu.

You are now ready to start programming your ESP8266 board using the Arduino IDE!

**2.2 PROGRAM ESP8266 Code:**

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include "index.h"

ESP8266WebServer myserver(80);

const char\* ssid = "Computercentre";

const char\* pass = "password";

void setup() {

  Serial.begin(9600);

  Serial.println("Connecting to WiFi network");

  WiFi.begin(ssid,pass);

  while(WiFi.status() != WL\_CONNECTED){

    Serial.print(".");

    delay(200);

  }

  Serial.println();

  Serial.println("Connected to the network");

  Serial.print("IP: ");

  Serial.println(WiFi.localIP());

  myserver.on("/",handle);

  myserver.begin();

}

void handle(){

  String home = homepage;

  myserver.send(200, "text/html", home);

}

void loop() {

  myserver.handleClient();

}

const char homepage[] PROGMEM = R"=====(

<html>

  <head>

    <title>SVCET- CSE </title>

    <style>

      body {

        font-family: Arial, sans-serif;

        background: linear-gradient(#341757,#1d053b,#100221,#100221);

        margin: 0;

        padding: 0;

      }

      #navbar {

        position: absolute;

        top: 0;

        width: 100%;

        flex-direction: row;

        align-items: center;

        justify-content: center;

        display: flex;

      }

      #navbar a{

        float: left;

        color: #fff;

        text-align: center;

        padding: 5px 16px;

        margin: 20px 20px;

        text-decoration: none;

        font-size: 17px;

        border: 2px solid white;

        background-color: transparent;

        border-radius: 15px;

        transition: background-color 0.5s ease;

        transition:  color 0.5s ease;

      }

      #navbar a:hover{

        color: #341757;

        background: white;

        transition: background 0.5s ease;

        transition:  color 0.5s ease;

      }

      .container {

        max-width: 1100px;

        margin: 0 auto;

        height: 100vh;

      }

      h1 {

        font-size: 4em;

        color: #fff;

        animation: zoom-animation 1s infinite;

        text-align: center;

        margin-top: 250px;

      }

      @keyframes zoom-animation {

      0% {

          transform: scale(1);

      }

      50% {

          transform: scale(1.1);

         }

      100% {

          transform: scale(1);

         }

      }

      p {

        font-size: 1.5em;

        color: whitesmoke;

        text-align: center;

        margin: 0px 20px;

        margin-top: 100px;

      }

    </style>

  </head>

  <body>

    <div id="navbar">

      <a href="#">Home</a>

      <a href="#">About</a>

      <a href="#">Products</a>

      <a href="#">Services</a>

      <a href="#">Events</a>

      <a href="#">Contact</a>

</div>

    <div class="container">

      <h1>SRI VENKATESWARA COLLEGE OF ENGINEERING</h1>

      <p>We provide economical, basic, premium variants of COE IoT lab with facilities of more than sensors to read the different parameters. We have vast experience on most popularly used wireless technologies like LoRaWAN, WiFi, BLE, GSM etc. IoT COE lab provides more than 50 different sensors, various development kits to read the data and do real time IoT projects.</p>

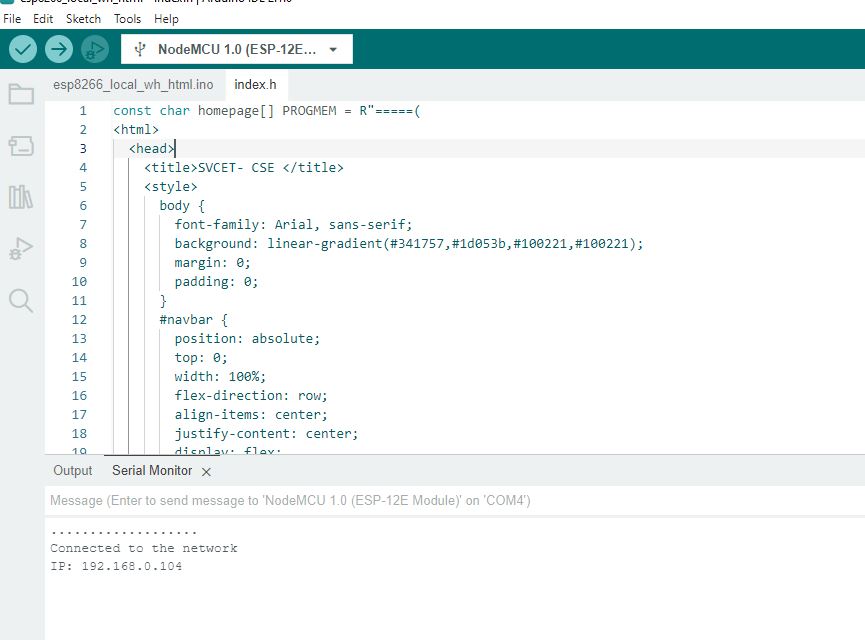
    </div>

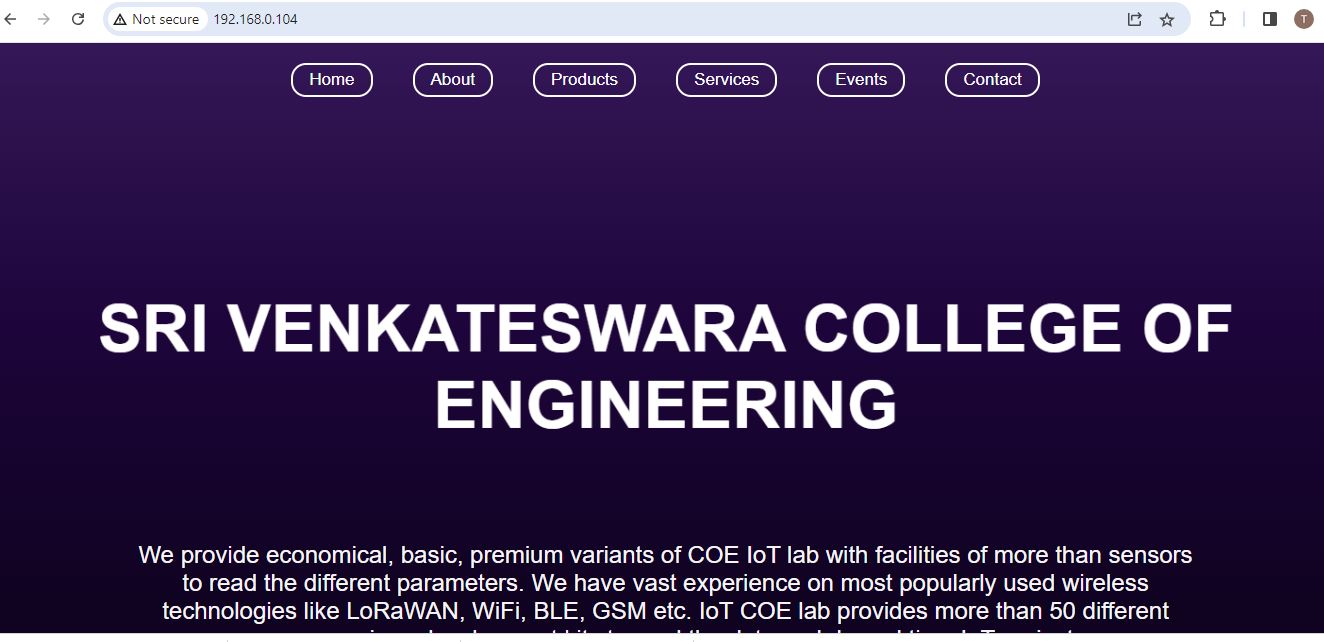
  </body>

</html>

)=====";

**OUTPUT :**





**Result:**

Set-up a local server using the ESP8266 WiFi module is successfully implemented.

**Experiment 7: Measure the room light intensity and output the data to the web API**

**Aim :** The aim of this Experiment is to send the LDR sensor data from the ESP8266 Wi-Fi module to the Thing Speak Cloud Platform and analyze the data using graphs and charts provided by Thing Speak.

**MODULE 2 - HARDWARE**

Hardware refers to any physical components/particulars of a system containing ICs, electronics, sensors, and circuit boards. Without hardware, an IoT system cannot exist, and the software developed won't be able to run.

**2.1 COMPONENTS REQUIRED**

● NodeMCU ESP8266

● USB Data cable

● LDR- Light Dependent Resistor

● Breadboard

● Jumper wires

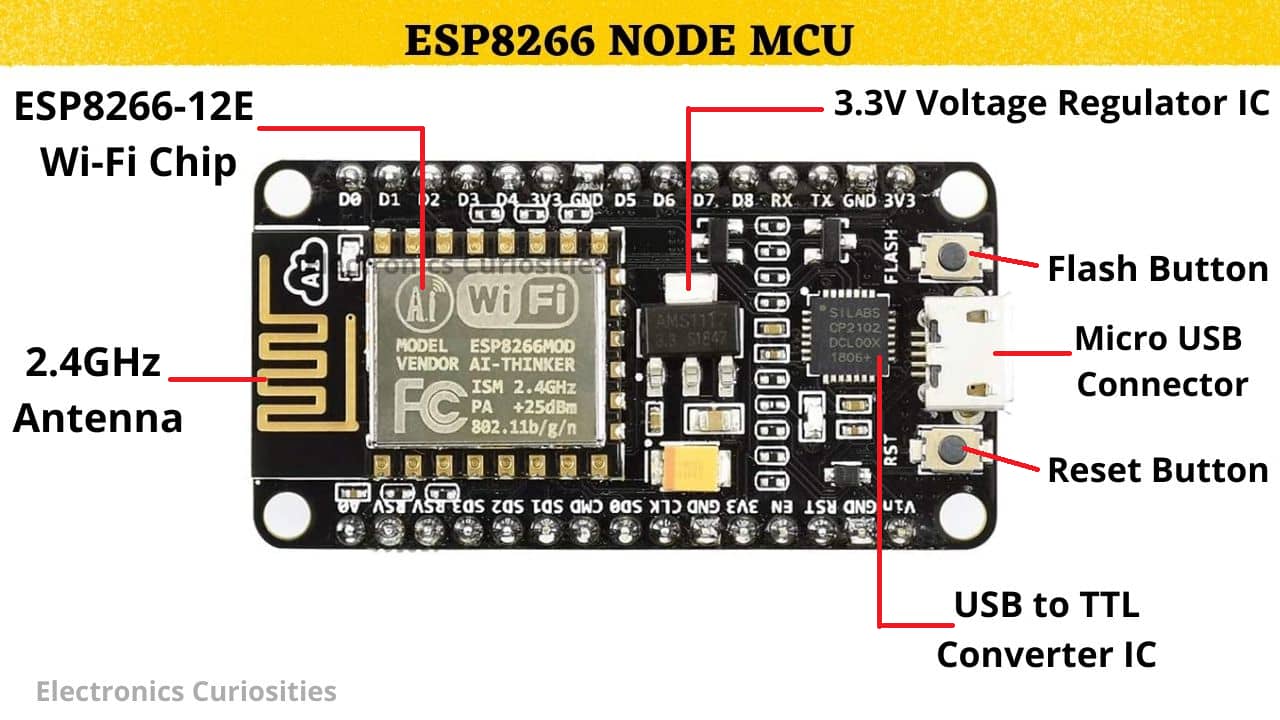
● 1K resistor

2.2 ANNEXURE

A. NodeMCU ESP8266:

The Node MCU ESP8266 is a popular WiFi development board that is based on the

ESP8266 WiFi module. It features a microcontroller, flash memory, and a built-in WiFi module, which makes it ideal for creating IoT devices that require wireless connectivity. The Node MCU ESP8266 is programmable using the Arduino IDE, Lua scripting language, or other programming languages such as Micro Python.



**B. BREADBOARD:**

Breadboards are one of the most fundamental pieces when learning how to build

circuits. Breadboards are commonly utilized while prototyping temporary circuits. It is useful to designers because it allows components to be removed and replaced easily.

**C. LDR SENSOR:**

A photoresistor (also known as a Photocell, or light-dependent resistor, LDR, or

photo-conductive cell) is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface.

**2.3 WIRING**

Connect one of the LDR terminals to "3.3V" of ESP8266 module and the other to the "GND" followed by a "1k" resistor and draw a wire between the LDR and resistor terminal and connect it to the "A0" of the NodeMCU.

**MODULE 3 - SOFTWARE**

Software is a generic term to refer to the scripts and programs that run on a microprocessor or microcontroller and execute specific tasks.

**3.1 GET START WITH ARDUINO IDE**

To set up the Arduino IDE for using ESP8266, follow these steps:

● Download and install the latest version of the Arduino IDE from the official website: https://www.arduino.cc/en/software.

● Open the Arduino IDE and go to File > Preferences.

● In the Preferences window, find the "Additional Boards Manager URLs" field and add the following URL: http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

● Click "OK" to close the Preferences window.

● Next, go to Tools > Board > Boards Manager. In the Boards Manager, search for "esp8266" and install the "esp8266 by ESP8266 Community" package.

● Once the installation is complete, go back to Tools > Board and select "NodeMCU 1.0(ESP-12E Module)" or the appropriate board that you are using.

● Finally, connect your ESP8266 board to your computer using a USB cable and select the appropriate port from the Tools > Port menu.

You are now ready to start programming your ESP8266 board using the Arduino IDE!

3.2 PROGRAM

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include "ThingSpeak.h"

const char\* ssid = "ssid"; //WiFi SSID

const char\* pass = "123123123"; //WiFi Passcode

WiFiClient client;

unsigned long channelNumber = 2074614; // Channel ID

const char\* APIwriteKey = "OT1FHJAES32JHLS4"; // API WriteKey

int data = 0;

void setup() {

Serial.begin(9600);

Serial.println("Connecting to the network");

WiFi.begin(ssid,pass);

while(WiFi.status() != WL\_CONNECTED){

Serial.print(".");

delay(150);

}

Serial.println();

Serial.println("Connected to the network");

ThingSpeak.begin(client);

}

void loop() {

Int data = analogRead(A0);

ThingSpeak.setField(1, data);

ThingSpeak.writeFields(channelNumber, APIwriteKey);

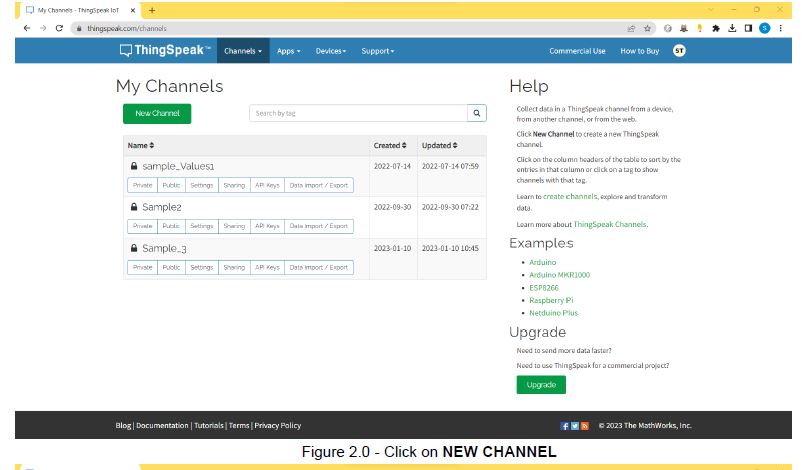
delay(15000);

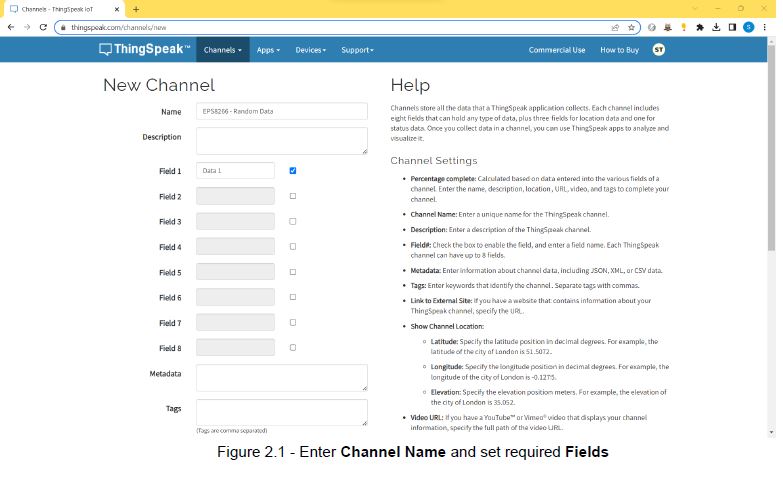
}

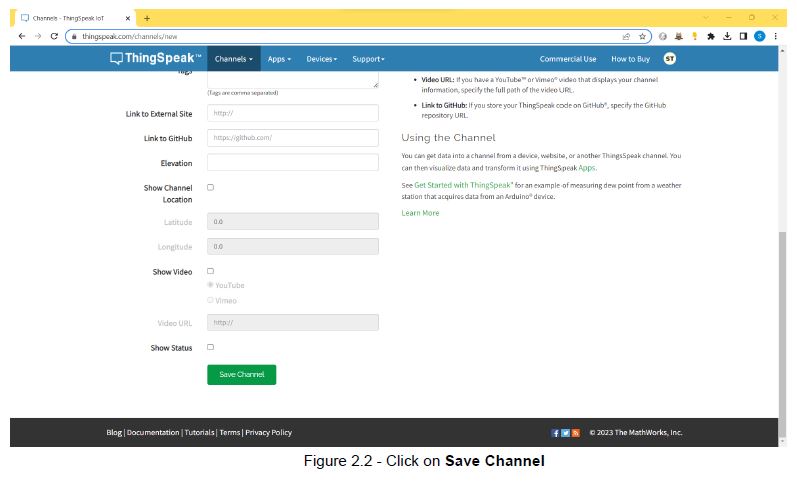
**3.3 THINGSPEAK CLOUD PLATFORM SET-UP**

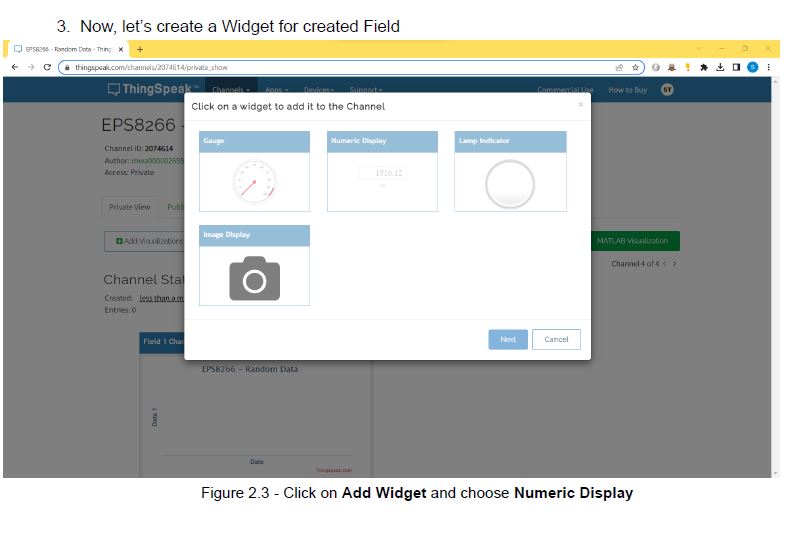
1. Go to https://thingspeak.com/ and create an account.

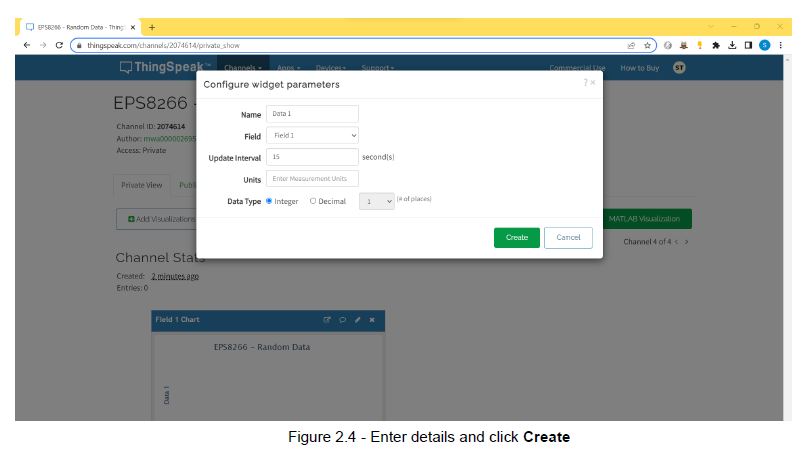
2. After creating the account, create a new channel. In this channel, we will store the data sent by the ESP8266.

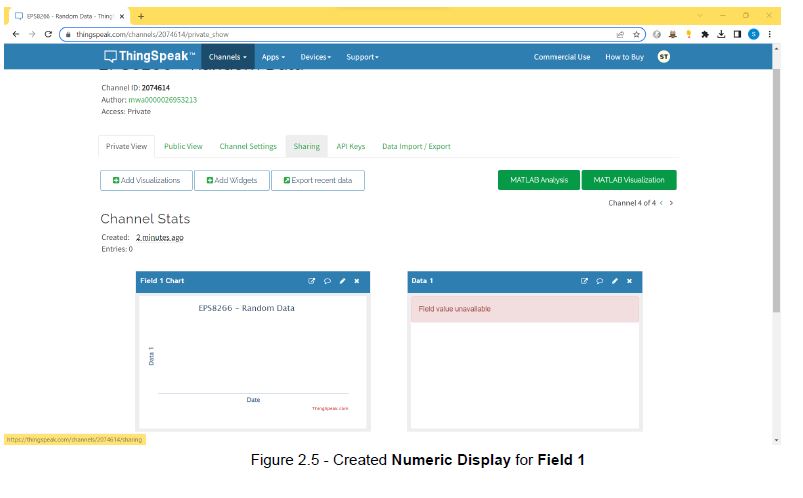


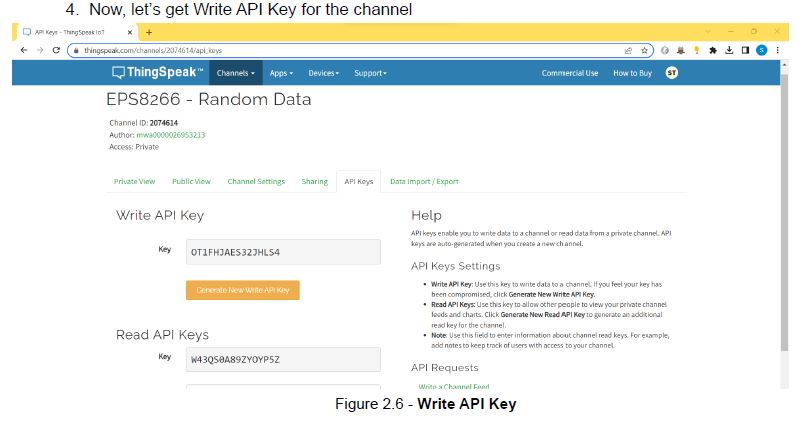












**MODULE 4 - METHODOLOGY**

● Set-up Thing Speak Cloud Platform

● Install the Arduino IDE software and set-up the environment for ESP8266 module

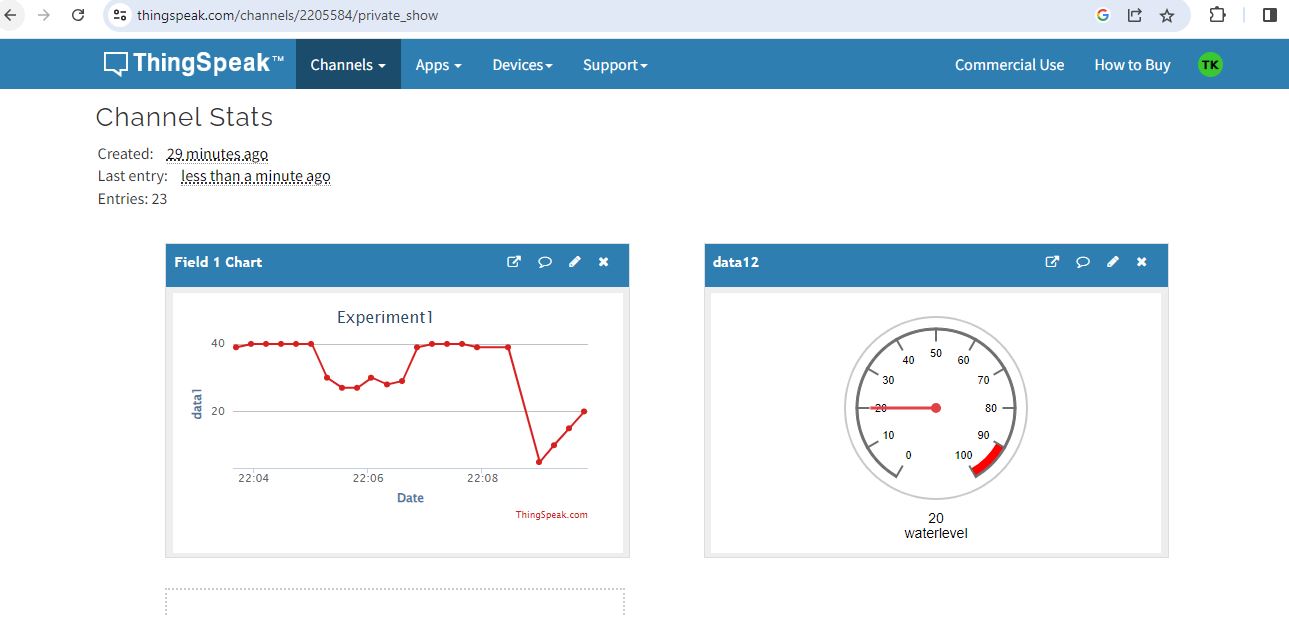
● Write the program and save the file

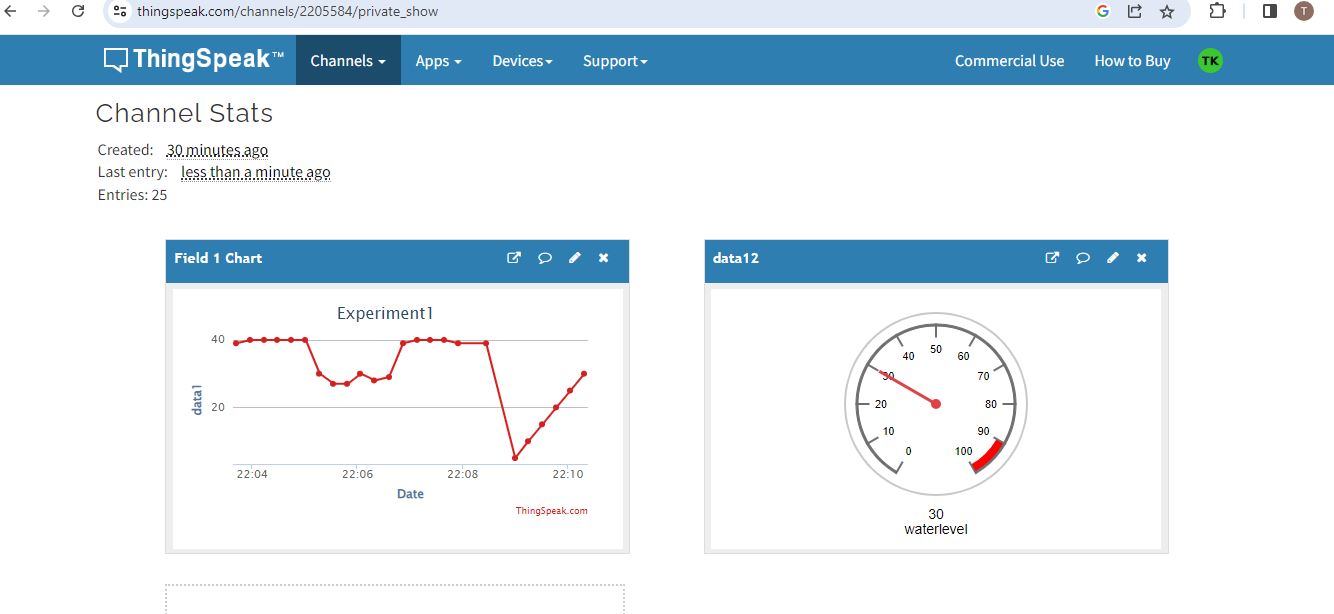
● Select respective Board and Port

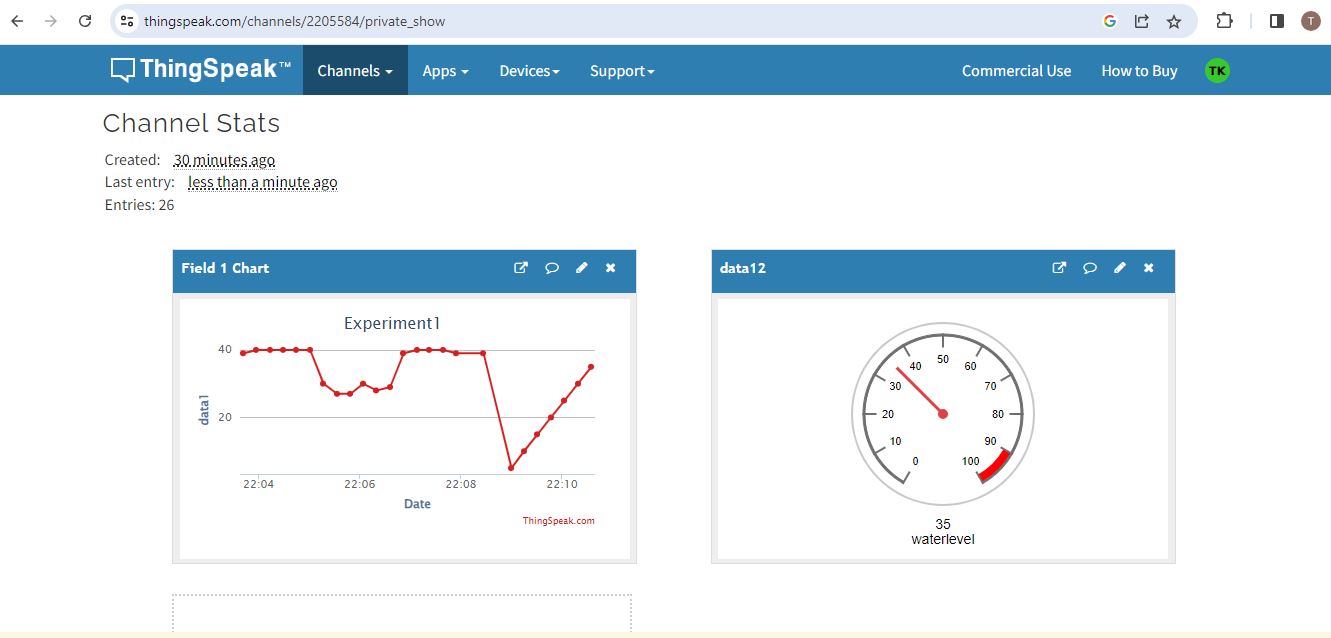
● Finally, upload the code to the microcontroller

**MODULE 5 – RESULT:**

This Experiment has demonstrated how to send the LDR sensor data from the ESP8266 Wi-Fi module to the Thing Speak Cloud Platform.







**Experiment 8: write an Arduino program to turn LED ON and OFF using LDR with Arduino Uno.**

**INTERFACING LDR**

AIM: To write an Arduino program to turn LED ON and OFF using LDR with Arduino Uno.

**APPARATUS REQUIRED:**

1. PC

2. Arduino IDE

3. LDR

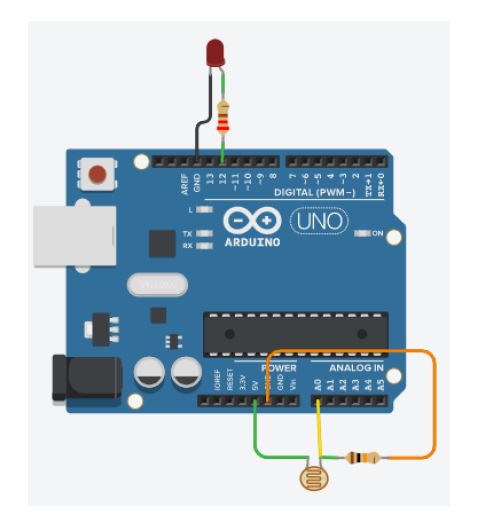
4. Bread Board-1No

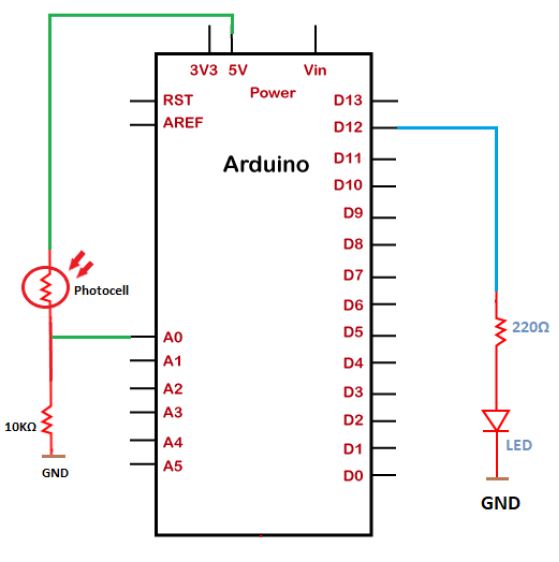
5. Resistors-10K Ω, 220Ω-1No

6. LDR-1No

7. Wires

**CIRCUIT DIAGRAM:**





**PROCEDURE:**

1. Connect the circuit as per the circuit diagram.

2. Connect Arduino to your PC.

3. Open the Arduino IDE in computer and write the program.

4. Compile the program for any errors and upload it to the Arduino.

5. Observe the output LED ON AND OFF.

**PROGRAM:**

const int ledPin = 13;

const int ldrPin = A0;

void setup() {

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

pinMode(ldrPin, INPUT);

}

void loop() {

int ldrStatus = analogRead(ldrPin);

if (ldrStatus <= 300) {

digitalWrite(ledPin, HIGH);

Serial.print("Its DARK, Turn on the LED : ");

Serial.println(ldrStatus);

} else {

digitalWrite(ledPin, LOW);

Serial.print("Its BRIGHT, Turn off the LED : ");

Serial.println(ldrStatus);

}

}

**RESULT:** Thus the LED is switched ON, OFF by LDR using Arduino Uno.