VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Experiment - 2

Programme	:	B.Tech. CSE	Semester	:	Winter 2023-24
Course Title	:	Internet of Things Lab	Code	:	BECE352E
			Slot	:	L23-24
Register Number	:	21BAI1106	Name	:	Ojas Patil
Faculty (s)	:	Dr. Manimaran P	Date	:	25th Feb 2024

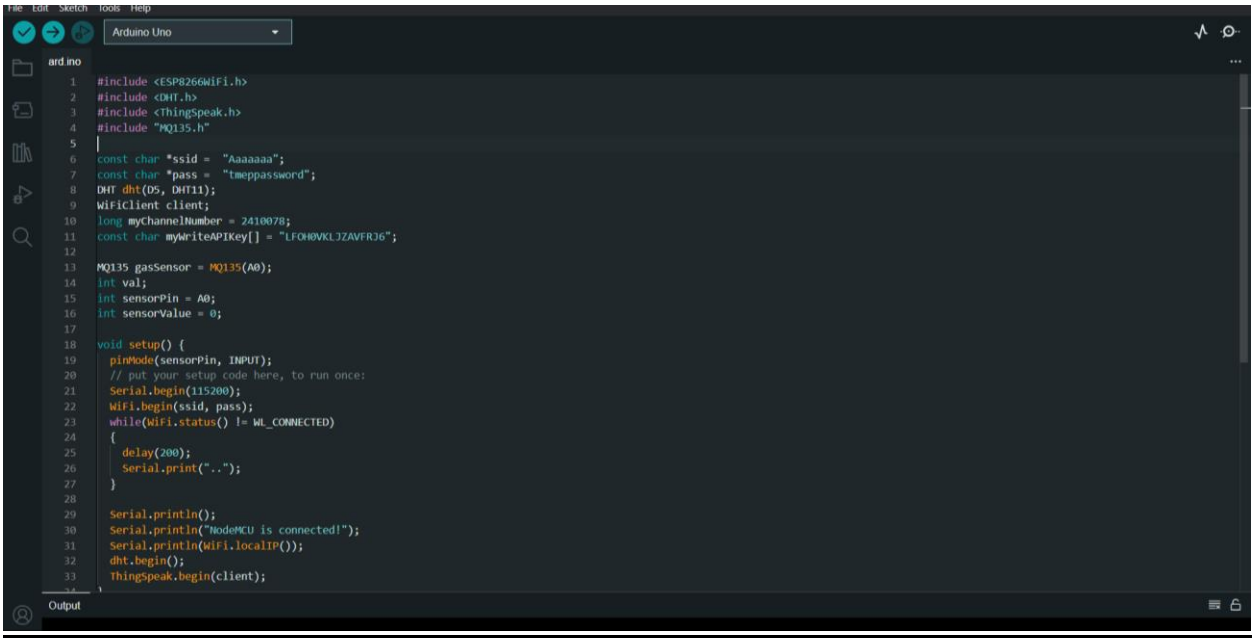
AIM

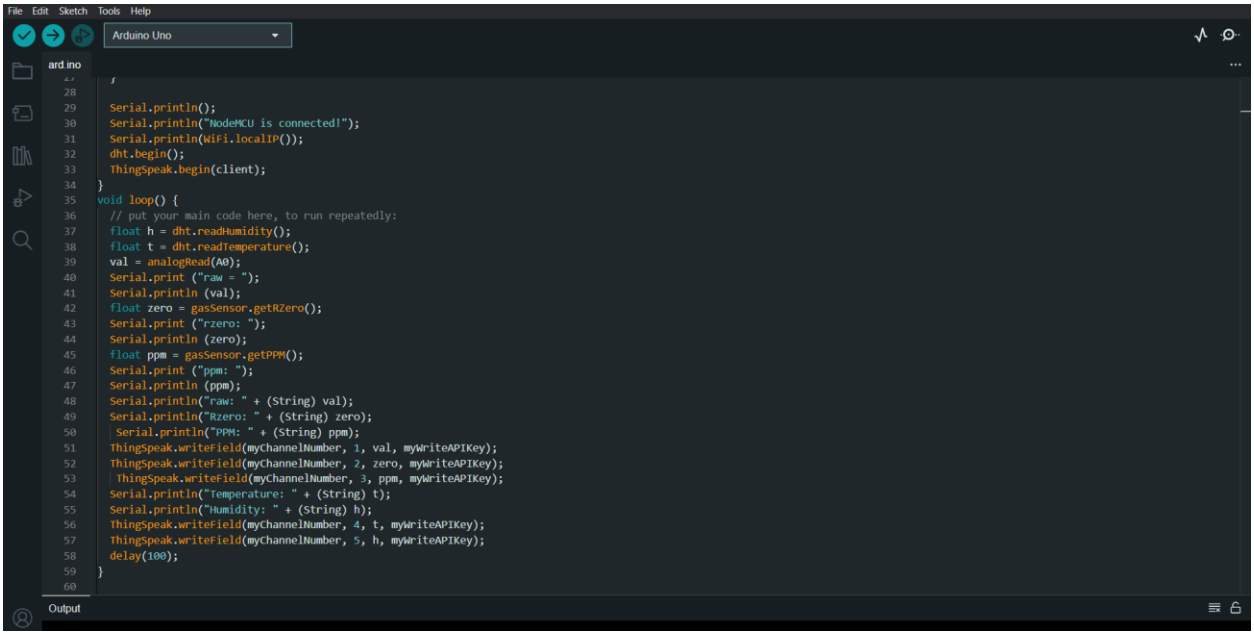
To develop an Air Quality Monitoring System using ESP8266 – NODEMCU with MQ135 and DHT 11 utilizing ThingSpeak Cloud Computing.

Steps to Follow

1) Initialize: Include libraries, define WiFi credentials, create sensor & communication objects, set sensor pin as input, initialize serial communication, connect to WiFi, initialize DHT sensor.
2) Main Loop: Read temperature, humidity, and raw analog value from sensors; calculate Rzero and PPM; print sensor readings (numerical & string) to serial monitor.
3) Serial Printing: Print raw sensor values and calculated values in both numerical and string formats to the serial monitor.
4) WiFi Connection: Attempt to connect to WiFi network, wait for connection & print dots, print confirmation message and local IP address upon successful connection.
5) ThingSpeak Communication: Use ThingSpeak library to write sensor readings (raw, Rzero, PPM, temperature, and humidity) to specific fields in your channel using channel number and API key.
6) Delay: Introduce a delay of 100 milliseconds before repeating the loop.

Code Screenshots





Source Code

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <ThingSpeak.h>
#include "MQ135.h"
```

```
const char *ssid =  "Aaaaaaa";
const char *pass =  "temppassword";
DHT dht(D5, DHT11);
WiFiClient client;
long myChannelNumber = 2410078;
const char myWriteAPIKey[] = "LFOH0VKLJZAVFRJ6";
MQ135 gasSensor = MQ135(A0);
int val;
int sensorPin = A0;
int sensorValue = 0;

void setup() {
  pinMode(sensorPin, INPUT);
  // put your setup code here, to run once:
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(200);
    Serial.print("..");
  }
  Serial.println();
  Serial.println("NodeMCU is connected!");
  Serial.println(WiFi.localIP());
  dht.begin();
  ThingSpeak.begin(client);
}
void loop() {
  // put your main code here, to run repeatedly:
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  val = analogRead(A0);
  Serial.print ("raw = ");
  Serial.println (val);
  float zero = gasSensor.getRZero();
  Serial.print ("rzero: ");
  Serial.println (zero);
  float ppm = gasSensor.getPPM();
  Serial.print ("ppm: ");
  Serial.println (ppm);
  Serial.println("raw: " + (String) val);
  Serial.println("Rzero: " + (String) zero);
   Serial.println("PPM: " + (String) ppm);
  ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 2, zero, myWriteAPIKey);
```

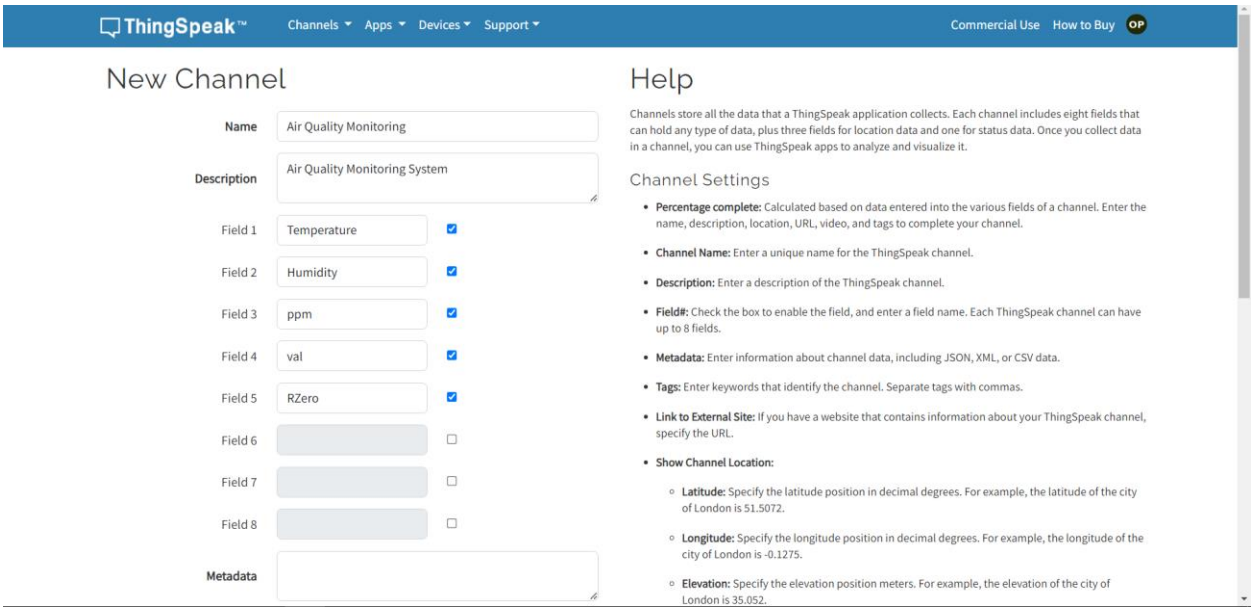
```
   ThingSpeak.writeField(myChannelNumber, 3, ppm, myWriteAPIKey);
  Serial.println("Temperature: " + (String) t);
  Serial.println("Humidity: " + (String) h);
  ThingSpeak.writeField(myChannelNumber, 4, t, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 5, h, myWriteAPIKey);
  delay(100);
}
```

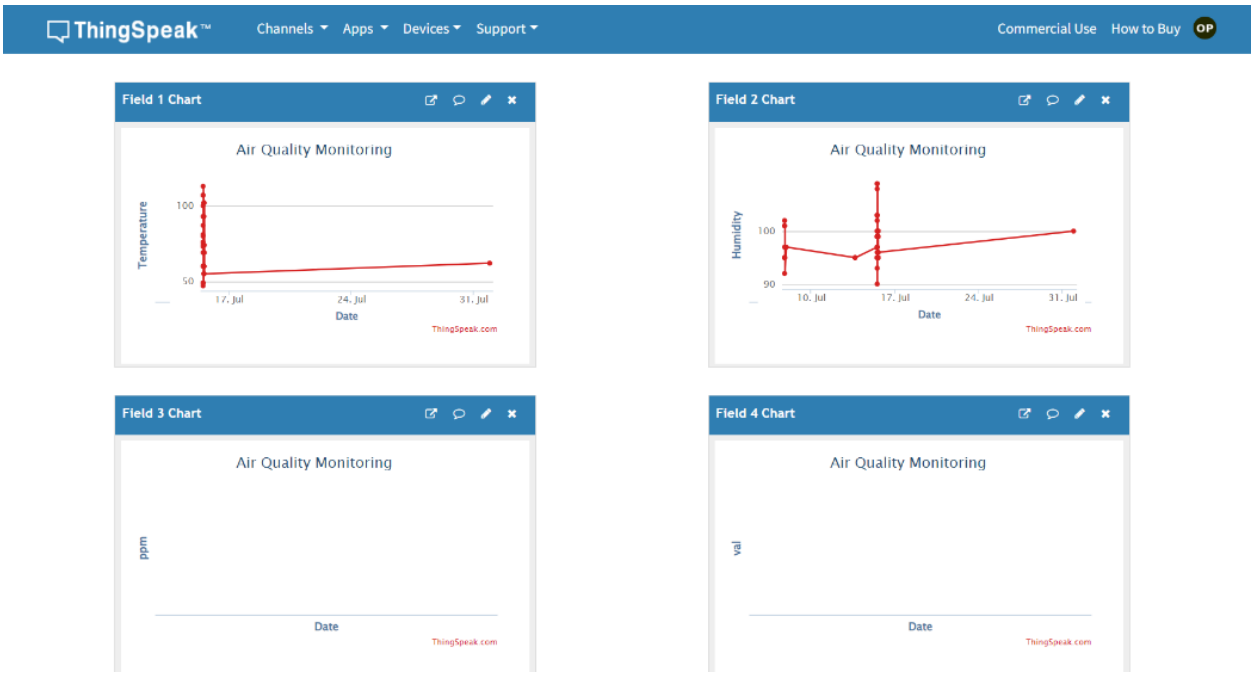
Features and Description

Sensor	NodeMCU 12E
DHT11	PIN in NodeMCU
+Vcc (+)	+3.3V
Dout/OUT	GPIO2 (D4) hardware connection In software: GPIO2 is Referred as PIN2 for digital input
GND (-)	GND (G)

Sensor	NodeMCU 12E
MQ135	PIN in NodeMCU
+Vcc (+)	+3.3V
A0 (analog out)	A0 In software: A0 is Referred as Analog output

Output Screenshot





Results

Thus, an Air Quality Monitoring System using ESP8266 – NODEMCU with MQ135 and DHT 11 utilizing ThingSpeak Cloud Computing has been developed and verified.