* **Source Code**

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16,2);

#include <DHT.h>

#define dhtpin 2

#define dhttype DHT11

DHT dht (dhtpin, dhttype);

//#define gas sensorpin 34

#define gastype mq3

#include "ThingSpeak.h"

#include <WiFi.h>

int valve=12;

char ssid[] ="User Id"; // your network SSID (name)

char pass[] ="Pass Key"; // your network password

int keyIndex = 0; // your network key Index number (needed only for WEP)

WiFiClient client;

unsigned long myChannelNumber = “ ”;

const char \* myWriteAPIKey =" ";

String myStatus = "";

//float t, h, val, volt, info, solar;

float t, h, val, p, info, ppm;

void setup()

{

lcd.init();

lcd.backlight();

lcd.begin(16,2);

dht.begin();

ThingSpeak.begin(client);

Serial.begin(115200);

pinMode(valve,OUTPUT);

delay(100);

}

void loop()

{

if(WiFi.status() != WL\_CONNECTED)

{

Serial.print("Attempting to connect to SSID: ");

Serial.println(ssid);

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

{

WiFi.begin(ssid, pass);

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

}

t=dht.readTemperature();

h=dht.readHumidity();

Serial.print("Temperature: ");

Serial.print(t);

Serial.print("\xC2\xB0");

Serial.print("C");

Serial.print("\t\t");

Serial.print("Humidity: ");

Serial.print(h);

Serial.println("%");

/\*if(mqtt.connected())

//if(gas sensor.connected())

int Gas\_value = analogRead(34);

Serial.println(Gas\_value);

Serial.print ("...") ;

//if (Gas\_data.publish(Gas\_value))

//void GASLevel() {

int value = analogRead(sensor);

value = map(value, 0, 4095, 0, 100);

Serial.print("bad: ");

Serial.print(PPM);

Serial.println();

delay(500);

\*/

val = analogRead(35);

Serial.println(info);

p = val\*5.0/4096.0;

Serial.print("bad smile: ");

Serial.print(p);

Serial.println("P");

info = analogRead(34);

Serial.println(info);

ppm= info\*66.0/4096.0;

//v = map(value, 0, 4095, 0, 100);

Serial.print("full: ");

Serial.print(ppm);

Serial.println("PPM");

Serial.println();

delay(500);

lcd.setCursor(0,0);

lcd.print("T: ");

lcd.print(t);

lcd.setCursor(9,0);

lcd.print("H: ");

lcd.print(h);

lcd.setCursor(0,1);

lcd.print("P: ");

lcd.print(p);

lcd.setCursor(9,1);

lcd.print("PPM: ");

lcd.print(ppm);

if(t>40)

{

digitalWrite(valve,1);

// lcd.setCursor(0,1);

//lcd.print("valve on");

// lcd.print(" ");

}

if(t<31)

{

digitalWrite(valve,0);

// lcd.setCursor(0,1);

//lcd.print("valve off");

// lcd.print(" ");

}

ThingSpeak.setField(1, t);

ThingSpeak.setField(2, h);

ThingSpeak.setField(3, p);

ThingSpeak.setField(4, ppm);

// ThingSpeak.setField(5, i);

// write to the ThingSpeak channel

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

if(x == 200)

{

Serial.println("Channel update successful.");

Serial.println();

}

else

{

Serial.println("Problem updating channel. HTTP error code " + String(x));

Serial.println();

}

// change the values

ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

delay(15000); // Wait 15 seconds to update the channel again

}