Coding:

Solution:

#include <Wire.h>

#include <PubSubClient.h>

#include <Adafruit\_ADS1015.h> Adafruit\_ADS1115 ads(0x48);

float Voltage = 0.0;

#include <OneWire.h>

#include <Dallas Temperature.h>

#define ONE WIRE BUS 18

OneWire oneWire(ONE\_WIRE\_BUS);

Dallas Temperature DS18B20(&oneWire);

#define senseinput

#define ORG "sovqa3"// IBM ORGANIZATION ID

#define DEVICE\_TYPE "lot-Rtrwqmacs"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM #define DEVICE\_ID "24681012"//DEVICE ID MENTIONED IN IOT WATSON PLATEFORM

#define TOKEN "12345678"//Token

String data3;

float dist;

//-------customize the above value

char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name

char publishtopic[]="rtrwqmacs/evt/Data/fmt/json";/\*topic name and type of event perform

and format in which data to be send\*/

char subscribetopic[]="rtrwqmacs/cmd/test/fmt/String":"cmd REPRESENT Command tupe and

COMMAND IS TEST OF FORMAT STRING"/

char authMethod[]="use-token-auth"://authentication method

char token[]=TOKEN;

char clientid[]="d:" ORG ":" DEVICE\_TYPE":" DEVICE\_ID://CLIENT ID

WiFiClient wifiClient;// creating an instance for wificlient

PubSubClient client(server, 1883, callback, wifiClient);

int senseRawValue; //Some variable

float senseTurbidity; //Some floating variable

#define analogpin

const int trigPin = 12;

const int echopin = 13;

// defines variables

long duration;

int distance;

int tankheight=27;

int mydistance;

int buf[10],temp;

int sensorval=0;

long int avgval;

int brdled =02;

////////// for http Client///////

#include <Arduino.h>

#include <WiFi.h>

#include <WiFiMulti.h>

#include <HTTPClient.h>

#define MY\_SERIAL Serial

void setup() {

pinMode(brdled,OUTPUT);

WiFiMulti wifiMulti; MY\_SERIAL.begin(115200); MY\_SERIAL.println(); for(uint8\_t t=4; t>0;t--) { MY\_SERIAL.printf("[SETUP] WAIT %d...\n", t);

MY\_SERIAL.println();

MY\_SERIAL.println();

MY\_SERIAL.flush();

delay(1000);

}

wifiMulti.addAP("WorkSHop", "inf12345");

wifiMulti.addAP("J-THEORY 3878", "987/3650");

while (wifiMulti.run() != WL\_CONNECTED) { //Check for the connection

delay(1000);

MY SERIAL.println("Connecting to WiFi..");

}

MY\_SERIAL.println("WiFi network connected");

////////////Setup for the sensors and ads1115//////// pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output pinMode(echopin, INPUT); // Sets the echopin as an Input

ads.begin(); // enables the ADC1115

MY SERIAL.println("Initializing All Sensors................");

delay(3000);

void loop() {

float mtemp,turb,ph,level; /// variables to hold sensor values(data)

mtemp= mytemp(); // hold temperature data:

turb=myturb(); // hold turbidity data

ph-myph(); //hold pH data

level=mylevel(); // hold water level data

//wait for WiFi connection

if((wifi Multi.run() == WL\_CONNECTED)) {

digitalWrite(brdled,HIGH);

delay(5000);

digitalWrite(brdled,LOW);

HTTPClient http:

MY SERIAL.print("[HTTP] begin...\n");

// configure traged server and url

http.begin("http://api.openweathermap.org/data/2.5/forecast?id=524901&appid=915c3fc3b97c6219

9e657fd7ad0c4edf"); //HTTP //defining a variabble to hold all values from sensors

String ourdata

=String(mtemp)+","+String(turb)+","+String(ph)+","+String(level);

MY SERIAL.println(ourdata);

MY SERIAL.print("[HTTP] POST...\n");

// start connection and send HTTP header

http.addHeader("Content-Type","text/plain");

int httpCode = http.POST(ourdata);

//httpCode will be negative on error

if(httpCode > 0) {

// HTTP header has been send and Server response header has been handled

MY\_SERIAL.printf("[HTTP] POST... code: %d\n", httpCode);

// file found at server

if(httpCode == HTTP\_CODE\_OK) {

String payload= http.getString();

MY\_SERIAL.println(payload);

else {

MY\_SERIAL.printf("[HTTP] POST... failed, error: %s\n",

http.errorToString(httpCode).c\_str());

wifiMulti.run();

if (wifiMulti.run()!= WL\_CONNECTED) { //Check for the connection

delay(1000);

wifiMulti.run();

MY SERIAL.println("Reconnecting to WiFi..");

} else {

MY\_SERIAL.println("Reconnected");

digitalWrite(brdled,HIGH);

delay(2000);

digitalWrite(brdled,LOW);

http.end();

}

delay(20000);

}

/////////Turbidity Sensor////////////

float myturb(){

int16\_t adc1; // we read from the ADC, we have a sixteen bit integer as a result adc1 = ads.readADC\_SingleEnded(1);

float voltage = (adc1\* 0.1875)/1000; //converting analog reading to voltage

(digital value)

senseTurbidity= voltage+1; // converting sensor voltage to 5V

return senseTurbidity:

MY\_SERIAL.print("TURBIDITY VALUE: "); //Print the output data to the

serial

MY SERIAL.println(sense Turbidity);

MY\_SERIAL.print("\n");

delay(1000);

if (sense Turbidity>=3.90){ MY\_SERIAL.println("\t Water is clear \n");

if (sense Turbidity<3.90 && sense Turbidity>=3.30 X MY\_SERIAL.println("\t Water is normal clear \n");

else if(senseTurbidity<3.30)

MY\_SERIAL.println("\t Warning. Water is muddy or very cloudy!!!!!!!\n");

/////////Ultrasonic Sensor/////////////

float mylevel(){

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10):

digitalWrite(trigPin, LOW);

duration = pulsein(echopin, HIGH);

return distance;

MY SERIAL.println(distance);

MY\_SERIAL.print("Distance: ");

MY SERIAL.println(distance);

if (distance<=10&& distance>=5){

MY\_SERIAL.println("The water level: FULL");

else if (distance> 10 && distance<=16}{ MY\_SERIAL.println("The water level: NORMAL");

else if (distance>16]{

MY\_SERIAL.println("The water level: LOW");

delay(1000);

/////////// pH Sensor ///

float myph() //////// using the ads1115 for the ph meter

int16\_t adc0; // we read from the ADC, we

adc0ads.readADC\_SingleEnded(0);

for(int i=0;i<10;i++){

//buf[i]= analogRead(analogpin);

buf[i]=adc0;

delay(100);

for(int i=0;i<9;i++){

for(int j=i;j<10;j++){

if(buf[i]>buflil){

temp=buf[j];

buf[i]=buf[j];

buf[j]=temp;

have a sixteen bit integer as a result

MY\_SERIAL.print("Distance: ");

MY SERIAL.println(distance);

if (distance<=10&& distance>=5){

MY\_SERIAL.println("The water level: FULL");

else if (distance> 10 && distance<=16}{ MY\_SERIAL.println("The water level: NORMAL");

else if (distance>16]{

MY\_SERIAL.println("The water level: LOW");

delay(1000);

/////////// pH Sensor ///

float myph() //////// using the ads1115 for the ph meter

int16\_t adc0; // we read from the ADC, we

adc0ads.readADC\_SingleEnded(0);

for(int i=0;i<10;i++){

//buf[i]= analogRead(analogpin);

buf[i]=adc0;

delay(100);

for(int i=0;i<9;i++){

for(int j=i;j<10;j++){

if(buf[i]>buflil){

temp=buf[j];

buf[i]=buf[j];

buf[j]=temp;

have a sixteen bit integer as a result

}