Develope A Python Script

1. Python Script

Date	13 November,2022
Team ID	PNT2022TMID31526
Project name	project-Real Time River Water Quality Monitoring And Control System
Maximum Marks	4 Mks

Coding:

Solution:

```
#include <Wire.h>
#include <PubSubClient.h>
#include <Adafruit_ADS1015.h>
Adafruit_ADS1115 ads(0x48);
float Voltage = 0.0;
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 18
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
#define senseInput
#define ORG "sovqa3"// IBM ORGANIZATION ID
#define DEVICE_TYPE "Iot-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
WiFiMulti wifiMulti; void
setup() {
pinMode(brdled,OUTPUT);
MY_SERIAL.begin(115200);
MY_SERIAL.println();
MY_SERIAL.println();
MY_SERIAL.println();
for(uint8_t t = 4; t > 0; t--) {
MY_SERIAL.printf("[SETUP] WAIT %d...\n",
t); MY_SERIAL.flush(); delay(1000);
}
wifiMulti.addAP("WorkSHop", "inf12345"); wifiMulti.addAP("J-
THEORY 3878", "98?J365o"); 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((wifiMulti.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=915c3fc3b97c621
9 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(senseTurbidity)
; MY_SERIAL.print("\n");
```

```
delay(1000); if
(senseTurbidity>=3.90 ){
MY_SERIAL.println("\t Water is clear \n");
} if (senseTurbidity<3.90 && senseTurbidity>=3.30
){
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 have a sixteen bit integer as a result adc0 =
ads.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]>buf[j]){
temp=buf[j];
buf[i]=buf[j];
buf[j]=temp;
}
}
} avgval=0; for(int
i=2;i<8;i++){avgval+=buf[i]; } float
ads_avg= avgval/6; float phvol=
(ads_avg * 0.1875)/1000; float
phval= -3.7429*phvol + 15.791;
MY_SERIAL.print("Sensor = ");
MY_SERIAL.println(phval);
MY_SERIAL.print("Voltage = ");
MY_SERIAL.println(phvol); delay(1000);
if (phval <=1 || phval>13.90){
MY_SERIAL.print("Check the pH
meter"); return 13.89 ; } return phval;
```

```
}
////////////////////Temperature
Sensor////////////////// float mytemp(){ float temp;
DS18B20.requestTemperatures();
temp=DS18B20.getTempCByIndex(0);
MY_SERIAL.print("Temperature: ");
return temp;
}
```