DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSRIBE TO IBM IOT PLATFORM

| Date | 06 November 2022 |
| --- | --- |
| Team ID | PNT2022TMID50829 |
| Project Name | Real time river water quality monitoring and Control system |

PROGRAM :

#include <SoftwareSerial.h>

#include<A7Client.h>

A7Client a7Client (7, 8); // RX, TX on Uno, connected to Tx, Rx of A7 Module

const char server[] = "waterqualitysite.azurewebsites.net";

//char message[] = "GET /log.php?t=2016-10-25,21:16:00&te=30.0&do=7.05&pH=5.66&tu=7&la=5.1492&ln=100.492 HTTP/1.1\r\nHost: waterqualitysite.azurewebsites.net\r\nConnection: close\r\n\r\n";

char msg[200] = "";

// char time[] = "2016-11-18,22:35:01";

char temperature[7] = "";//-30.00

char pH[6] = "";//14.00

char DO[7] = ""; //100.00%

char turbidity[5] = "";//4000

char lat[9] = "";

char lng[9] = "";

const uint8\_t pinRST = 5;

const uint8\_t pinPWR = 6;

const uint8\_t pinTurbidity = A0;

const uint8\_t pinPH = A1;

const uint8\_t pinDO = A2;

const uint8\_t pinTemperature = A3;

struct Data{

float temperature;

float pH;

float DO;

int turbidity;

} data;

unsigned long previousMillis = 0; //for looping purpose

long timeInterval = 10000; //update once per 10 seconds

void setup() {

Serial.begin(9600);

Serial.println(freeRam());

pinMode(13, OUTPUT);

digitalWrite(13, LOW); //disable the bright RED LED!

pinMode(pinRST, OUTPUT);

pinMode(pinPWR, OUTPUT);

digitalWrite(pinRST, HIGH);//reset the A7 module

delay(1000);

digitalWrite(pinRST, LOW);//finish the reset

digitalWrite(pinPWR, LOW);//POWER UP the A7 module via a PNP transistor

delay(2000); //need to apply power to the pin for >2 seconds

digitalWrite(pinPWR, HIGH);//finish the power up

delay(3500);//let the module stable, it will output some gibberish.

a7Client.changeBaud();//Baud rate for A7.serial is now at 9600 bps

Serial.println(freeRam());

Serial.println(F("Response okay! :) Module is alive!"));

delay(10000); //wait for the config message for GPRS and everything

while(!a7Client.startGPS()){// make sure GPS is on

Serial.println(F("Try activating GPS again."));

}

// a7Client.startGPS();

getData(&data);

printData(&data);

while(!a7Client.readGPS());//force Arduino to complete one GPS at least once =.= 16/11/16

if(a7Client.parse(a7Client.lastNMEA())){

// Serial.print(F("Location: "));

// Serial.print(a7Client.latitude, 4); Serial.print(a7Client.lat);

// Serial.print(F(", "));

// Serial.print(a7Client.longitude, 4); Serial.println(a7Client.lon);

// Serial.print(F("Location (in degrees, works with Google Maps): "));

// Serial.print(a7Client.latitudeDegrees, 4);

// Serial.print(F(", "));

// Serial.println(a7Client.longitudeDegrees, 4);

}

//force the GPS to be DUP if lat=0.000, 0.000

if( a7Client.latitude - 0.000 < 0.0000001){

a7Client.latitude = 5.356575;

a7Client.longitude = 100.294404;

}

convertToChar(&data, a7Client);

// if (lat[0] == '0'){ //force the GPS to be DUP :)

// lat[0] = '5';

// lat[1] = '.';

// lat[2] = '3';

// lat[3] = '5';

// lat[4] = '6';

// lat[5] = '5';

// lat[6] = 0;

// lng[0] = '1';

// lng[0] = '0';

// lng[0] = '0';

// lng[0] = '.';

// lng[0] = '2';

// lng[0] = '9';

// lng[0] = '1';

// }

constructHTTPRequest(msg);

a7Client.connect(server, 80);

sendHTTPRequest(msg);

}

void loop() {

// Serial.println(F("I am down here"));

// if (a7Client.available())

// Serial.write(a7Client.read());

// if (Serial.available())

// a7Client.writeSerial(Serial.read());

//start of millis() code from AdaFruit

unsigned long currentMillis = millis();

getData(&data);

printData(&data);

if((currentMillis - previousMillis >= timeInterval)){

previousMillis = currentMillis;

getData(&data);

printData(&data);

while(!a7Client.readGPS());//force Arduino to complete one GPS at least once =.= 16/11/16

if(a7Client.parse(a7Client.lastNMEA())){

Serial.println(F("parsed completed"));

}

convertToChar(&data, a7Client);

printCharData();

constructHTTPRequest(msg);

if (a7Client.connect(server, 80)){

digitalWrite(13, HIGH);

delay(500); //blink LED13

digitalWrite(13, LOW);

}

sendHTTPRequest(msg);

}

}