

## Sprint 2

<b>Date</b>	14.11.2022
<b>Team ID</b>	PNT2022TMID28239
<b>Project Name</b>	Real-Time Water QualityMonitoring And Control System

```
#include <ESP8266HTTPClient.h>
#include <FirebaseArduino.h>
#include <DNSServer.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
#include <OneWire.h>
#include <TimeLib.h> //library to get time and date
#include <WiFiUdp.h>
#include <OneWire.h>
#include <Servo.h>

// Set these to run example.
#define FIREBASE_HOST "iot839-a034d.firebaseio.com"
#define FIREBASE_AUTH
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "yyy"

#define StartConvert 0
#define ReadTemperature 1
```

```

#define ecSwitch D6
#define tempSwitch D7
#define turbiditySwitch D8

/* EC and Temp */
const byte numReadings = 20;    //the number of sample times
byte ECSensorPin = A0; //EC Meter analog output,pin on analog 1
byte DS18B20_Pin = D2; //DS18B20 signal, pin on digital 2
unsigned int
AnalogSampleInterval=25,printInterval=700,tempSampleInterval=850; //analog
sample interval;serial print interval;temperature sample interval
unsigned int readings[numReadings];    // the readings from the analog input
byte indx = 0;           // the indx of the current reading
unsigned long AnalogValueTotal = 0;           // the running total
unsigned int AnalogAverage = 0,averageVoltage=0;           // the average
unsigned long AnalogSampleTime,printTime,tempSampleTime;
float temperature,ECcurrent;

//Temperature chip i/o
OneWire ds(DS18B20_Pin); // on digital pin 2

#define pHsensorPin A0
#define phSwitch D5
unsigned long int avgValueForPH; //Store the average value of the ph sensor
feedback
int pHbuffer[10],tempValueForPH;
float phValue;

```

```
#define turbiditysensorPin A0      //turbidity meter Analog output to Arduino
Analog Input 0
unsigned long int avgValueForTurbidity; //Store the average value of the turbidity
sensor feedback
int turbiditybuffer[10],tempValueForTurbidity;
float turbidityValue;

#define MUX_A D3
#define MUX_B D4

Servo myservo;
#define servoPin D1

// NTP Servers:
static const char ntpServerName[] = "asia.pool.ntp.org";
const int timeZone = +6; // Convert to Bangladesh Standard Time (BST)

WiFiUDP Udp;
unsigned int localPort = 8888; // local port to listen for UDP packets

time_t getNtpTime();
void sendNTPpacket(IPAddress &address);

void setup()
{
```

```
Serial.begin(115200);  
// connect to wifi using WifiManager library.  
WiFiManager wifiManager;  
//wifiManager.autoConnect("AutoConnectAP");  
wifiManager.autoConnect("PureraWater");
```

```
Serial.println();  
Serial.print("connected: ");  
Serial.println(WiFi.localIP());
```

```
for (byte thisReading = 0; thisReading < numReadings; thisReading++)  
    readings[thisReading] = 0;  
TempProcess(StartConvert); //let the DS18B20 start the convert  
AnalogSampleTime=millis();  
printTime=millis();  
tempSampleTime=millis();
```

```
pinMode(MUX_A, OUTPUT);  
pinMode(MUX_B, OUTPUT);  
pinMode(phSwitch,OUTPUT);  
pinMode(ecSwitch,OUTPUT);  
pinMode(turbiditySwitch,OUTPUT);
```

```
myservo.attach(servoPin);  
Udp.begin(localPort);
```

```
//Serial.print("Local port: ");
//Serial.println(Udp.localPort());
//Serial.println("waiting for sync");
setSyncProvider(getNtpTime);
setSyncInterval(300);

Serial.begin(115200);
}

void loop()
{

    /* READING PH */
    motorOn();
    delay(3000);
    digitalWrite(phSwitch, HIGH); //power up ph sensor
    changeMux(LOW, LOW); // selector S1=0, S0 = 0; ph setup in Y0
    delay(3000);
    phRead(); //taking reading
    digitalWrite(phSwitch, LOW); //power down ph sensor after reading
    delay(3000); //reading complete, now preaparing to take next reading

    motorOFF();
    delay(3000);
    for (int i=0;i<3;i++){
        digitalWrite(ecSwitch,HIGH);
```

```

changeMux(LOW, HIGH); // selector S1=0, S0 = 0; ph setup in Y1
delay(3000);
EcAndTempReading();
digitalWrite(ecSwitch,LOW);
delay(3000);
}

while(ECcurrent<0){
    digitalWrite(ecSwitch,HIGH);
    changeMux(LOW, HIGH); // selector S1=0, S0 = 0; ph setup in Y1
    delay(3000);
    EcAndTempReading();
    digitalWrite(ecSwitch,LOW);
    delay(3000);
}
motorOn();
delay(3000);
/* READING turbidity */
digitalWrite(turbiditySwitch, HIGH); //power up turbidity sensor
changeMux(HIGH, LOW); // selector S1=1, S0 = 0; turbidity setup in Y2
delay(3000);
turbidityRead();//taking reading
digitalWrite(turbiditySwitch, LOW);//power down turbidity sensor after reading
delay(3000);//reading complete, now preaparing to take next reading

StaticJsonBuffer<200> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();

```

```
String date = (String) day()+'/' +month()+'/' +year();
String timee = (String) hour()+':' +minute()+':' +second();
root["date"] = date;
root["time"] = timee;
root["turbidity"] = turbidityValue;
root["ph"] = phValue;
root["temp"] = temperature;
root["ec"] = ECcurrent;
Serial.println("Firebase data: ");
Serial.print("Date: ");
Serial.println(date);
Serial.print("Time: ");
Serial.println(timee);
Serial.print("Turbidity: ");
Serial.println(turbidityValue);
Serial.print("PH: ");
Serial.println(phValue);
Serial.print("Temperature: ");
Serial.println(temperature);
Serial.print("EC: ");
Serial.println(ECcurrent);

}

void changeMux(int b, int a) {
    digitalWrite(MUX_A, a);
```

```

    digitalWrite(MUX_B, b);
}

void motorOn(){
    myservo.write(180);
}

void motorOFF(){
    myservo.write(0);
}

void phRead(){
    for(int i=0;i<10;i++)    //Get 10 sample value from the sensor for smooth the
    value
    {
        pHbuffer[i]=analogRead(pHsensorPin);
        // Serial.println(pHbuffer[i]);
        delay(10);
    }
    avgValueForPH=0;
    for(int i=2;i<8;i++)    //take the average value of 6 center sample
        avgValueForPH+=pHbuffer[i];
    pHValue=(float)avgValueForPH*3.33/1024/6; //convert the analog into millivolt
    pHValue=4.7*pHValue;    //convert the millivolt into pH value
    Serial.print("  pH:");
    Serial.print(pHValue,2);
    Serial.println(" ");}

```