#include <SoftwareSerial.h>

//#define DEBUG 1

#define Offset 0.00 //deviation compensate

#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0

#define printInterval 5000

SoftwareSerial TDS\_Serial(2, 3);// Create software serial port object (RX, TX)

struct Measurements {

float TDS;

float Temp;

float PH;

};

class TDS\_Sensor {

private:

void TDS\_sendHexData(String hexString) {

int hexStringLength = hexString.length();

if (hexStringLength % 2 != 0) {

// Ensure that the length of the hexadecimal string is an even number

hexString = "0" + hexString;

hexStringLength++;

}

for (int i = 0; i < hexStringLength; i += 2) {

// Extract a pair of characters from a hexadecimal string

String hexPair = hexString.substring(i, i + 2);

// Convert hexadecimal string to bytes

byte hexValue = (byte)strtol(hexPair.c\_str(), NULL, 16);

// Send bytes

TDS\_Serial.write(hexValue);

}

}

bool check()

{

if (TDS\_Serial.available() >= 6) {

byte resp[6];

for (int i = 0; i < 6; i++) {

resp[i] = TDS\_Serial.read();

}

if (resp[0] == 0xAC &&

resp[1] == 0x00 &&

resp[2] == 0x00 &&

resp[3] == 0x00 &&

resp[4] == 0x00 &&

resp[5] == 0xAC) {

return true;

} else if (resp[0] == 0xAC)

{

#ifdef DEBUG

switch (resp[1])

{

case 0x01: Serial.println("TDS command frame abnormality"); break;

case 0x02: Serial.println("TDS device busy"); break;

case 0x03: Serial.println("TDS calibration failed"); break;

case 0x04: Serial.println("TDS detects temperature out of range"); break;

default: Serial.println("Unknown Error");

}

#endif

return false;

}

}

return false;

}

public:

TDS\_Sensor() {}

void init()

{

TDS\_Serial.begin(9600);

delay(100);

String Calibration\_data = "A600000000A6";

TDS\_sendHexData(Calibration\_data);

while (!check()) ; // wait

String Set\_res = "A3000186A0CA";

TDS\_sendHexData(Set\_res);

while (!check()) ; // wait

String Set\_B\_NTC = "A50F0A0000BE";

TDS\_sendHexData(Set\_B\_NTC);

while (!check()) ; // wait

}

void read(Measurements &Measure)

{

String Detection\_data = "A000000000A0";

TDS\_sendHexData(Detection\_data); // start reading

while (TDS\_Serial.available() < 6) ; //wait

if (TDS\_Serial.available() > 0) {

byte start = TDS\_Serial.read();

if (start == 0xAA) {

byte tdsHi = TDS\_Serial.read();

byte tdsLo = TDS\_Serial.read();

int tdsValue = (tdsHi << 8) + tdsLo;

byte tempHi = TDS\_Serial.read();

byte tempLo = TDS\_Serial.read();

int tempAdc = (tempHi << 8) + tempLo;

float temp = tempAdc / 100.0;

byte checksum = TDS\_Serial.read();

byte sum = start + tdsHi + tdsLo + tempHi + tempLo;

if ((sum & 0xFF) == checksum) {

Measure.TDS = tdsValue;

Measure.Temp = temp;

#ifdef DEBUG

Serial.print("TDS: ");

Serial.print(tdsValue);

Serial.print("ppm Temp: ");

Serial.print(temp);

Serial.println("°C");

#endif

}

}

}

}

};

TDS\_Sensor tds;

Measurements m;

float mapFloat(float x, float in\_min, float in\_max, float out\_min, float out\_max) {

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

tds.init();

// pinMode(SensorPin,INPUT);

}

void loop() {

static unsigned long printTime = millis();

if (millis() - printTime > printInterval)

{

tds.read(m);

// float pHValue = analogRead(SensorPin); // Read and reverse the analogue input value from the pH sensor.

// Serial.println(Po);

// pHValue = mapFloat(pHValue, 0.0, 1023.0, 0.0, 14.0); // Map the '0 to 1023' result to '0 to 14'.

// m.PH = pHValue;

m.PH = 7.0 - (m.Temp - 25.0)\*0.02;

#ifdef DEBUG

Serial.print(" pH value: ");

Serial.println(pHValue, 2);

Serial.print("TDS : ");

Serial.println(m.TDS);

#else

Serial.print(m.TDS);

Serial.print(",");

Serial.print(m.PH);

Serial.print(",");

Serial.println(m.Temp);

#endif

}

}