**TDS METER CODE :-**

We need to Download One Wire Library and Dallas library

#include <OneWire.h>

#include <DallasTemperature.h>

#include <LiquidCrystal.h>

// 16x2 LCD

#define rs 10

#define en 9

#define d4 6

#define d5 5

#define d6 4

#define d7 3

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

namespace pin {

const byte tds\_sensor = A1;

const byte one\_wire\_bus = 7; // Dallas Temperature Sensor

}

namespace device {

float aref = 4.3;

}

namespace sensor {

float ec = 0;

unsigned int tds = 0;

float waterTemp = 0;

float ecCalibration = 1;

}

OneWire oneWire(pin::one\_wire\_bus);

DallasTemperature dallasTemperature(&oneWire);

void setup() {

Serial.begin(115200); // Dubugging on hardware Serial 0

lcd.begin(16, 2);

dallasTemperature.begin();

}

void loop() {

readTdsQuick();

delay(1000);

}

void readTdsQuick() {

dallasTemperature.requestTemperatures();

sensor::waterTemp = dallasTemperature.getTempCByIndex(0);

float rawEc = analogRead(pin::tds\_sensor) \* device::aref / 1024.0; // read the analog value more stable by the median filtering algorithm, and convert to voltage value

float temperatureCoefficient = 1.0 + 0.02 \* (sensor::waterTemp - 25.0); // temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02\*(fTP-25.0));

sensor::ec = (rawEc / temperatureCoefficient) \* sensor::ecCalibration; // temperature and calibration compensation

sensor::tds = (133.42 \* pow(sensor::ec, 3) - 255.86 \* sensor::ec \* sensor::ec + 857.39 \* sensor::ec) \* 0.5; //convert voltage value to tds value

Serial.print(F("TDS:")); Serial.println(sensor::tds);

Serial.print(F("EC:")); Serial.println(sensor::ec, 2);

Serial.print(F("Temperature:")); Serial.println(sensor::waterTemp,2);

lcd.clear();

lcd.print("TDS EC Temp");

lcd.setCursor(0,1);

lcd.print(sensor::tds);

lcd.setCursor(5,1);

lcd.print(sensor::ec, 2);

lcd.setCursor(11,1);

lcd.print(sensor::waterTemp,2);

}

