FULL SOURCE CODE FOR IOT AUTOMATIC IRRIGATION SYSTEM

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#include <Arduino.h>

#include <U8g2lib.h>

#include <Wire.h>

#include <math.h>

#include <EEPROM.h>

#include <Keypad.h>

#include "TANK1.h"

#include "temp.h"

#include "hum.h"

#include "Alarm.h"

#include "Well.h"

#include "network.h"

#include "PUMP.h"

#define BLYNK\_PRINT Serial

// #define BLYNK\_TEMPLATE\_ID "TMPL2lLYCUD3H"

// #define BLYNK\_TEMPLATE\_NAME "JET A1 FUEL LEVEL MONITOR"

// #define BLYNK\_AUTH\_TOKEN "i-D2U16Tykbk71DF6wCXIZXMzGenErCp"

#define BLYNK\_TEMPLATE\_ID "TMPL2moGk8JZI"

#define BLYNK\_TEMPLATE\_NAME "IoT Smart Irrigation system "

#define BLYNK\_AUTH\_TOKEN "HbVQTH3VrASvxWoHMqhEiw5qSwzAAEgG"

//-----------------------------------------------------------------------

// Select your modem:

#define TINY\_GSM\_MODEM\_SIM800

//-----------------------------------------------------------------------

#include <TinyGsmClient.h>

#include <BlynkSimpleTinyGSM.h>

//-----------------------------------------------------------------------

char auth[] = BLYNK\_AUTH\_TOKEN;

//-----------------------------------------------------------------------

char apn[] = "web.gprs.mtnnigeria.net";

char user[] = "";

char pass[] = "";

//-----------------------------------------------------------------------

// or Software Serial on Uno, Nano

#define rxPin 15

#define txPin 2

HardwareSerial sim800(1);

TinyGsm modem(sim800);

unsigned long previousMillis = 1500;

const long interval = 500;

unsigned long previousMillis\_tank = 1000;

const long interval\_tank = 1500;

#define pushbutton 34

const int MANUAL\_START\_STOP = 32; // the number of the pushbutton pin

int RELAYState = HIGH; // the current state of the output pin

int buttonState; // the current reading from the input pin

int lastButtonState = HIGH; // the previous reading from the input pin

// milliseconds, will quickly become a bigger number than can be stored in an int.

unsigned long lastDebounceTime = 0; // the last time the output pin was toggRELAY

unsigned long debounceDelay = 50; // the debounce time; increase if the output flickers

RF24 radio(21, 22);

const byte address[][32] = { "00001", "00002" };

U8G2\_ST7920\_128X64\_F\_SW\_SPI u8g2(U8G2\_R0, /\* clock=\*/5, /\* data=\*/17, /\* CS=\*/4, /\* reset=\*/16);

const byte ROWS = 4; //four rows

const byte COLS = 3; //three columns

char keys[ROWS][COLS] = {

{ '1', '2', '3' },

{ '4', '5', '6' },

{ '7', '8', '9' },

{ '\*', '0', '#' }

};

byte rowPins[ROWS] = { 33, 25, 26, 27 }; //connect to the row pinouts of the keypad

byte colPins[COLS] = { 14, 12, 13 }; //connect to the column pinouts of the keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

long b = 0;

int c = 0;

float i = 0; // reading from the sensor

int H2O\_percent\_1 = 0;

unsigned long Vcm;

unsigned short int Water\_Vol;

//int tank\_diameter = 0;

long int Num1, Num2, Number;

unsigned int addr\_1 = 0;

unsigned int addr\_2 = 50;

unsigned int addr\_3 = 100;

unsigned int addr\_4 = 150;

unsigned int height; //double

unsigned int diameter;

unsigned int fillable;

unsigned int autostart;

double tank\_height; // Thank Height

double tank\_diameter; // Tank Diameter

double tank\_fillable;

double pump\_autostart;

long int Tank\_volume;

int WH\_1;

//unsigned int radious = tank\_diameter;

float fill;

float empty;

//const int // buzzer = 32;

// // buzzer

const int BatteryPin = 36;

float battery\_voltage = 0;

int set = 0;

int flag1 = 0, flag2 = 0;

int hum;

int temp;

int distance;

int RX\_STARTSTOP\_STATE;

int soilmoisturepercent;

int pressure;

long altitude;

int EC\_PERC;

int TDS\_PERC;

struct Data\_Package {

byte hum;

byte temp;

byte distance;

byte pressure;

long altitude;

byte RX\_STARTSTOP\_STATE;

byte soilmoisturepercent;

byte TDS\_PERC;

byte EC\_PERC;

};

Data\_Package data;

//---------------------------------------------------------------------

BlynkTimer timer;

#define INTERVAL 500L

//-----------------------------------------------------------------------

void IOT\_Data() {

display();

//-----------------------------------------------------------------------

//Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

Blynk.virtualWrite(V0, distance);

Blynk.virtualWrite(V1, WH\_1);

Blynk.virtualWrite(V2, H2O\_percent\_1);

Blynk.virtualWrite(V3, pressure);

Blynk.virtualWrite(V4, altitude);

Blynk.virtualWrite(V5, soilmoisturepercent);

Blynk.virtualWrite(V6, EC\_PERC);

Blynk.virtualWrite(V7, TDS\_PERC);

Blynk.virtualWrite(V8, hum);

Blynk.virtualWrite(V9, temp);

Blynk.virtualWrite(V10, Water\_Vol);

Blynk.virtualWrite(V11, RELAYState);

// Blynk.virtualWrite(V12, AS);

// Blynk.virtualWrite(V13, F);

// Blynk.virtualWrite(V14, D);

// Blynk.virtualWrite(V15, v);

//-----------------------------------------------------------------------

}

void setup() {

EEPROM.begin(500);

Serial.begin(115200);

u8g2.begin();

radio.begin();

radio.openWritingPipe(address[0]); //Setting the address at which we will send the data

radio.openReadingPipe(1, address[1]); //Setting the address at which we will receive the data

radio.setAutoAck(false);

radio.setDataRate(RF24\_250KBPS);

radio.setPALevel(RF24\_PA\_MAX);

//pinMode( buzzer, OUTPUT);

pinMode(pushbutton, INPUT);

pinMode(MANUAL\_START\_STOP, INPUT\_PULLUP);

u8g2.clearBuffer();

u8g2.setFont(u8g2\_font\_helvB08\_tf);

u8g2.setCursor(15, 9);

u8g2.print(" IoT AUTOMATIC"); //18

u8g2.setCursor(3, 24);

u8g2.print(" IRRIGATION SYSTEM");

u8g2.setCursor(60, 37);

u8g2.print("BY");

u8g2.setCursor(0, 50);

u8g2.print("Ayuba Ridwan");

u8g2.setCursor(0, 63);

u8g2.print("Chigbu Ikechuku");

u8g2.setFont(u8g2\_font\_4x6\_tf);

u8g2.setCursor(89, 50);

u8g2.print("18/30GC025");

u8g2.setCursor(89, 63);

u8g2.print("18/30GC029");

u8g2.sendBuffer();

delay(5000);

u8g2.clearBuffer();

u8g2.setFont(u8g2\_font\_helvB08\_tf);

u8g2.setCursor(35, 24);

u8g2.print("SUPERVISED");

u8g2.setCursor(60, 37);

u8g2.print("BY");

u8g2.setCursor(10, 50);

u8g2.print(" Dr A.I Abdullateef");

u8g2.sendBuffer();

delay(5000);

sim800.begin(115200, SERIAL\_8N1, rxPin, txPin);

Serial.println("SIM800L serial initialize");

// delay(3000);

Serial.println("Initializing modem...");

modem.restart();

//-----------------------------------------------------------

Blynk.begin(auth, modem, apn, user, pass);

//-----------------------------------------------------------

// Setup a function to be called every second

timer.setInterval(INTERVAL, IOT\_Data);

//-----------------------------------------------------------

}

void display() {

// delay(5);

u8g2.clearBuffer();

radio.startListening(); //This sets the module as receiver

if (radio.available()) {

radio.read(&data, sizeof(Data\_Package)); // Read the whole data and store it into the 'data' structure

u8g2.drawXBMP(47, 23, u8g2\_network\_width, u8g2\_network\_height, u8g2\_hum\_network);

}

hum = data.hum;

temp = data.temp;

distance = data.distance;

soilmoisturepercent = data.soilmoisturepercent;

pressure = data.pressure \* 11.0F; //100.0F

altitude = data.altitude;

EC\_PERC = data.EC\_PERC;

TDS\_PERC = data.TDS\_PERC;

RX\_STARTSTOP\_STATE = data.RX\_STARTSTOP\_STATE;

char key = keypad.getKey();

if ((key) && (c < 5)) {

if (key == '1') //If Button 1 is pressed

{

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 1;

else

Number = (Number \* 10) + 1; //Pressed twice

}

if (key == '2') //Button 2 is Pressed

{

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 2;

else

Number = (Number \* 10) + 2; //Pressed twice

}

if (key == '3') {

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 3;

else

Number = (Number \* 10) + 3; //Pressed twice

}

if (key == '4') //If Button 4 is pressed

{

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 4;

else

Number = (Number \* 10) + 4; //Pressed twice

}

if (key == '5') {

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 5;

else

Number = (Number \* 10) + 5; //Pressed twice

}

if (key == '6') {

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 6;

else

Number = (Number \* 10) + 6; //Pressed twice

}

if (key == '7') //If Button 7 is pressed

{

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 7;

else

Number = (Number \* 10) + 7; //Pressed twice

}

if (key == '8') {

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 8;

else

Number = (Number \* 10) + 8; //Pressed twice

}

if (key == '9') {

//tone( buzzer, 50, 20);

if (Number == 0)

Number = 9;

else

Number = (Number \* 10) + 9; //Pressed twice

}

if (key == '0') {

//tone( buzzer, 50, 20);

if (Number == 0) {

Number = 0;

c = c - 1;

} else

Number = (Number \* 10) + 0; //Pressed twice

}

c = c + 1;

}

if (key == '\*') //If cancel Button is pressed

{

//tone( buzzer, 50, 20);

Number = 0;

c = 0;

}

// well();

/\*===============SCROLL TO SET THE THANK HEIGHT AND TANK DIAMETER AND FILLABLE SPACE=================\*/

// Scroll to the left

if (digitalRead(pushbutton) == 0) {

if (flag1 == 0 && flag2 == 0) {

flag1 = 1;

set = set + 1;

if (set > 4) {

set = 0;

}

delay(50);

}

}

else {

flag1 = 0;

}

/\*========= RETREAVING SETTING SAVED TO EEPROM =====\*/

tank\_height = EEPROM.get(addr\_1, height);

tank\_diameter = EEPROM.get(addr\_2, diameter);

pump\_autostart = EEPROM.get(addr\_3, autostart);

tank\_fillable = EEPROM.get(addr\_4, fillable);

i = distance; // tank\_height FROM SENSOR TRANSMITTER

u8g2.setFontMode(1); /\* activate transparent font mode \*/

u8g2.setDrawColor(1); /\* color 1 for the box \*/

u8g2.drawBox(78, 0, 128, 32);

u8g2.setDrawColor(2);

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setFontDirection(0);

/\*AUTOSTART\*/

u8g2.setCursor(79, 31);

int AS = (autostart); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("AS=");

u8g2.print(AS);

u8g2.print("%");

/\*TANK fillable space\*/

u8g2.setCursor(79, 23);

int F = (fillable); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("FS=");

u8g2.print(F);

u8g2.print("%");

/\*THANKS HEIGHT\*/

u8g2.setCursor(79, 7);

int v = (tank\_height); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("TH=");

u8g2.print(v);

u8g2.print("cm");

/\*THANKS DIAMETER\*/

u8g2.setCursor(79, 15);

int D = (tank\_diameter); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("TR=");

u8g2.print(D);

u8g2.print("cm");

/\*SENSOR READING \*/

u8g2.setFont(u8g2\_font\_4x6\_tf);

u8g2.setCursor(0, 62);

u8g2.print("SH=");

u8g2.print(distance);

u8g2.print("cm");

/\*========= FIRST PAGE SETTING =====\*/

if (set == 1) {

u8g2.setFontMode(1); /\* activate transparent font mode \*/

u8g2.setDrawColor(1); /\* color 1 for the box \*/

u8g2.drawBox(78, 0, 128, 32);

u8g2.setDrawColor(2);

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setFontDirection(0);

u8g2.setCursor(94, 24);

u8g2.print(Number);

/\*THANKS HEIGHT\*/

u8g2.setCursor(79, 7);

int v = (height); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("TH=");

u8g2.print(v);

u8g2.print("cm");

height = Number;

if ((key == '#') && height < 301) {

//tone( buzzer, 120, 20);

EEPROM.put(addr\_1, height);

EEPROM.commit();

}

if ((key == '#') && height > 301) //||

{

height = 0;

//tone( buzzer, 2000, 3000);

}

}

/\*========= SECOUND PAGE SETTING =====\*/

if (set == 2) {

u8g2.setFontMode(1); /\* activate transparent font mode \*/

u8g2.setDrawColor(1); /\* color 1 for the box \*/

u8g2.drawBox(78, 0, 128, 32);

u8g2.setDrawColor(2);

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setFontDirection(0);

u8g2.setCursor(94, 24);

u8g2.print(Number);

/\*THANKS DIAMETER\*/

u8g2.setCursor(79, 15);

int D = (diameter); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("TR=");

u8g2.print(D);

u8g2.print("cm");

diameter = Number;

if ((key == '#') && diameter < 301) {

//tone( buzzer, 120, 20);

EEPROM.put(addr\_2, diameter);

EEPROM.commit();

}

if ((key == '#') && diameter > 301) //||

{

diameter = 0;

//tone( buzzer, 2000, 3000);

}

}

/\*========= THIRD PAGE SETTING =====\*/

if (set == 3) {

u8g2.setFontMode(1); /\* activate transparent font mode \*/

u8g2.setDrawColor(1); /\* color 1 for the box \*/

u8g2.drawBox(78, 0, 128, 32);

u8g2.setDrawColor(2);

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setFontDirection(0);

u8g2.setCursor(94, 24);

u8g2.print(Number);

/\*THANKS fillable space\*/

u8g2.setCursor(79, 15);

int F = (fillable); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("FS=");

u8g2.print(F);

u8g2.print("%");

fillable = Number;

if ((key == '#') && fillable < 98) {

//tone( buzzer, 120, 20);

EEPROM.put(addr\_4, fillable);

EEPROM.commit();

}

if ((key == '#') && fillable > 98) //||

{

fillable = 0;

//tone( buzzer, 2000, 3000);

}

}

/\*========= FORTH PAGE SETTING =====\*/

if (set == 4) {

u8g2.setFontMode(1); /\* activate transparent font mode \*/

u8g2.setDrawColor(1); /\* color 1 for the box \*/

u8g2.drawBox(78, 0, 128, 32);

u8g2.setDrawColor(2);

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setFontDirection(0);

u8g2.setCursor(94, 24);

u8g2.print(Number);

/\*THANKS fillable space\*/

u8g2.setCursor(79, 15);

int AS = (autostart); // my tank\_height cylinder weight dat was enter on keypad in cm

u8g2.print("AS=");

u8g2.print(AS);

u8g2.print("%");

autostart = Number;

if ((key == '#') && autostart < 70) {

//tone( buzzer, 120, 20);

EEPROM.put(addr\_3, autostart);

EEPROM.commit();

}

if ((key == '#') && autostart > 70) //||

{

fillable = 0;

//tone( buzzer, 2000, 3000);

}

}

fill = tank\_fillable / 100;

empty = pump\_autostart / 100;

//===============Water HEIGHT=======================//

int Tank\_1\_Fillable\_space = tank\_height \* fill;

int Tank\_1\_empty\_space = tank\_height - Tank\_1\_Fillable\_space;

WH\_1 = tank\_height - distance;

//=============Water PERCENTAGE=================//

H2O\_percent\_1 = map(distance, Tank\_1\_empty\_space, tank\_height, 100, 0);

if (H2O\_percent\_1 >= 100) {

H2O\_percent\_1 = 100;

}

if (H2O\_percent\_1 <= 0) {

H2O\_percent\_1 = 0;

}

if (height <= 0) {

height = 0;

}

//==========Water VOLUME===============//

Vcm = PI \* tank\_diameter \* tank\_diameter \* WH\_1;

Water\_Vol = Vcm / 1000;

//=====================Tanke\_volume====================//

long int Tank\_size = PI \* tank\_diameter \* tank\_diameter \* tank\_height;

Tank\_volume = Tank\_size / 1000;

if (WH\_1 < 0) {

WH\_1 = 0;

}

if (Water\_Vol <= 2) {

Water\_Vol = 0;

}

if (Water\_Vol > Tank\_volume) {

Water\_Vol = Tank\_volume;

}

u8g2.drawBox(71, 48, 128, 64);

u8g2.setFont(u8g2\_font\_5x7\_tf);

u8g2.setCursor(31, 55);

u8g2.print("TV=");

u8g2.print(Tank\_volume);

u8g2.print("L");

u8g2.setCursor(31, 63);

u8g2.print("WV=");

u8g2.print(Water\_Vol); //k, 1

u8g2.print("L");

u8g2.setFont(u8g2\_font\_4x6\_tf);

u8g2.setCursor(72, 54);

u8g2.print("Height Level");

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setCursor(71, 63);

u8g2.print(WH\_1); //l / 1000, 1

u8g2.setFont(u8g2\_font\_5x7\_tf);

u8g2.print("cm");

u8g2.setFont(u8g2\_font\_courR08\_tf);

u8g2.setCursor(103, 63);

u8g2.print(H2O\_percent\_1);

u8g2.print("%");

if (H2O\_percent\_1 < 20) {

//digitalWrite(buzzer, HIGH);

} else if (H2O\_percent\_1 > 20) {

// digitalWrite(buzzer, LOW);

}

//EMPTY

if (H2O\_percent\_1 <= 0) {

H2O\_percent\_1 = 0;

unsigned long currentMillis\_tank = millis();

if (currentMillis\_tank - previousMillis\_tank >= interval\_tank) {

previousMillis\_tank = currentMillis\_tank;

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_0\_25\_bits);

}

}

else if ((H2O\_percent\_1 > 0) && (H2O\_percent\_1 <= 2)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_2\_25\_bits);

}

else if ((H2O\_percent\_1 > 2) && (H2O\_percent\_1 <= 4)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_4\_25\_bits);

}

else if ((H2O\_percent\_1 > 4) && (H2O\_percent\_1 <= 6)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_6\_25\_bits);

}

else if ((H2O\_percent\_1 > 6) && (H2O\_percent\_1 <= 8)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_8\_25\_bits);

}

else if ((H2O\_percent\_1 > 8) && (H2O\_percent\_1 <= 10)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_10\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 10) && (H2O\_percent\_1 <= 12)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_12\_25\_bits);

}

else if ((H2O\_percent\_1 > 12) && (H2O\_percent\_1 <= 14)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_14\_25\_bits);

}

else if ((H2O\_percent\_1 > 14) && (H2O\_percent\_1 <= 16)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_16\_25\_bits);

} else if ((H2O\_percent\_1 > 16) && (H2O\_percent\_1 <= 18)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_18\_25\_bits);

}

else if ((H2O\_percent\_1 > 18) && (H2O\_percent\_1 <= 20)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_20\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 20) && (H2O\_percent\_1 <= 22)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_22\_25\_bits);

}

else if ((H2O\_percent\_1 > 22) && (H2O\_percent\_1 <= 24)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_24\_25\_bits);

}

else if ((H2O\_percent\_1 > 24) && (H2O\_percent\_1 <= 26)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_26\_25\_bits);

}

else if ((H2O\_percent\_1 > 26) && (H2O\_percent\_1 <= 28)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_28\_25\_bits);

}

else if ((H2O\_percent\_1 > 28) && (H2O\_percent\_1 <= 30)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_30\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 30) && (H2O\_percent\_1 <= 32)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_32\_25\_bits);

}

else if ((H2O\_percent\_1 > 32) && (H2O\_percent\_1 <= 34)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_34\_25\_bits);

}

else if ((H2O\_percent\_1 > 34) && (H2O\_percent\_1 <= 36)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_36\_25\_bits);

}

else if ((H2O\_percent\_1 > 36) && (H2O\_percent\_1 <= 38)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_38\_25\_bits);

}

else if ((H2O\_percent\_1 > 38) && (H2O\_percent\_1 <= 40)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_40\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 40) && (H2O\_percent\_1 <= 42)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_42\_25\_bits);

}

else if ((H2O\_percent\_1 > 42) && (H2O\_percent\_1 <= 44)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_44\_25\_bits);

}

else if ((H2O\_percent\_1 > 44) && (H2O\_percent\_1 <= 46)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_46\_25\_bits);

} else if ((H2O\_percent\_1 > 46) && (H2O\_percent\_1 <= 48)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_48\_25\_bits);

}

else if ((H2O\_percent\_1 > 48) && (H2O\_percent\_1 <= 50)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_50\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 50) && (H2O\_percent\_1 <= 52)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_52\_25\_bits);

}

else if ((H2O\_percent\_1 > 52) && (H2O\_percent\_1 <= 54)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_54\_25\_bits);

}

else if ((H2O\_percent\_1 > 54) && (H2O\_percent\_1 <= 56)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_56\_25\_bits);

}

else if ((H2O\_percent\_1 > 56) && (H2O\_percent\_1 <= 58)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_58\_25\_bits);

}

else if ((H2O\_percent\_1 > 58) && (H2O\_percent\_1 <= 60)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_60\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 60) && (H2O\_percent\_1 <= 62)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_62\_25\_bits);

}

else if ((H2O\_percent\_1 > 62) && (H2O\_percent\_1 <= 64)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_64\_25\_bits);

}

else if ((H2O\_percent\_1 > 64) && (H2O\_percent\_1 <= 66)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_66\_25\_bits);

}

else if ((H2O\_percent\_1 > 66) && (H2O\_percent\_1 <= 68)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_68\_25\_bits);

}

else if ((H2O\_percent\_1 > 68) && (H2O\_percent\_1 <= 70)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_70\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 70) && (H2O\_percent\_1 <= 72)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_72\_25\_bits);

}

else if ((H2O\_percent\_1 > 72) && (H2O\_percent\_1 <= 74)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_74\_25\_bits);

}

else if ((H2O\_percent\_1 > 74) && (H2O\_percent\_1 <= 76)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_76\_25\_bits);

} else if ((H2O\_percent\_1 > 76) && (H2O\_percent\_1 <= 78)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_78\_25\_bits);

}

else if ((H2O\_percent\_1 > 78) && (H2O\_percent\_1 <= 80)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_80\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 80) && (H2O\_percent\_1 <= 82)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_82\_25\_bits);

}

else if ((H2O\_percent\_1 > 82) && (H2O\_percent\_1 <= 84)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_84\_25\_bits);

}

else if ((H2O\_percent\_1 > 84) && (H2O\_percent\_1 <= 86)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_86\_25\_bits);

}

else if ((H2O\_percent\_1 > 86) && (H2O\_percent\_1 <= 88)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_88\_25\_bits);

}

else if ((H2O\_percent\_1 > 88) && (H2O\_percent\_1 <= 90)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_90\_25\_bits);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////

else if ((H2O\_percent\_1 > 90) && (H2O\_percent\_1 <= 92)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_92\_25\_bits);

}

else if ((H2O\_percent\_1 > 92) && (H2O\_percent\_1 <= 94)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_94\_25\_bits);

}

else if ((H2O\_percent\_1 > 94) && (H2O\_percent\_1 <= 96)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_96\_25\_bits);

}

else if ((H2O\_percent\_1 > 96) && (H2O\_percent\_1 <= 98)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_98\_25\_bits);

}

else if ((H2O\_percent\_1 > 98) && (H2O\_percent\_1 <= 100)) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_100\_25\_bits);

}

else if (H2O\_percent\_1 >= 100) {

u8g2.drawXBMP(2, 0, u8g2\_tank1\_0\_25\_width, u8g2\_tank1\_0\_25\_height, u8g2\_tank1\_100\_25\_bits);

}

//Tempreture logo//

u8g2.drawXBMP(30, 1, u8g2\_temp\_width, u8g2\_temp\_height, u8g2\_temp\_bits);

//Humidity logo//

u8g2.drawXBMP(30, 27, u8g2\_hum\_width, u8g2\_hum\_height, u8g2\_hum\_bits);

u8g2.setCursor(43, 14);

u8g2.print(temp);

u8g2.setCursor(43, 43);

u8g2.print(hum);

u8g2.setFont(u8g2\_font\_4x6\_tf);

u8g2.setCursor(62, 6);

u8g2.print("TDS");

u8g2.setCursor(60, 12);

u8g2.print(TDS\_PERC);

u8g2.print("%");

u8g2.setCursor(62, 22);

u8g2.print("EC");

u8g2.setCursor(60, 28);

u8g2.print(EC\_PERC);

u8g2.print("%");

u8g2.setCursor(62, 38);

u8g2.print("SM");

u8g2.setCursor(60, 44);

u8g2.print(soilmoisturepercent);

u8g2.print("%");

u8g2.setCursor(76, 38);

u8g2.print("ALT");

u8g2.setCursor(76, 45);

u8g2.print(altitude);

u8g2.print("M");

u8g2.setCursor(97, 38);

u8g2.print("PRE");

u8g2.setCursor(96, 45);

u8g2.print(pressure);

u8g2.print("p");

if (RELAYState == 1) {

u8g2.drawXBMP(116, 34, u8g2\_ON\_width, u8g2\_ON\_height, u8g2\_ON\_bits);

}

else if (RELAYState == 0) {

u8g2.drawXBMP(116, 34, u8g2\_OFF\_width, u8g2\_OFF\_height, u8g2\_OFF\_bits);

}

//=============PUMP CONTROL STATUS=======================//

if (RELAYState == 1 && H2O\_percent\_1 == 100) {

RELAYState = 0;

}

else if (RELAYState == 1 && H2O\_percent\_1 < 100) {

RELAYState = 1;

}

else if (RELAYState == 0 && H2O\_percent\_1 < pump\_autostart) {

RELAYState = 1;

}

u8g2.sendBuffer();

/\*

Serial.print("dis ");

Serial.print(distance);

Serial.print(" TEMP ");

Serial.print(temp);

Serial.print(" HUM ");

Serial.print(hum);

Serial.print(" PRE ");

Serial.print(pressure);

Serial.print(" ALT ");

Serial.print(altitude);

Serial.print(" SM ");

Serial.print(soilmoisturepercent);

Serial.print(" EC\_PERC ");

Serial.print(EC\_PERC);

Serial.print(" TDS\_PERC ");

Serial.print(TDS\_PERC);

\*/

}

void loop() {

display();

TX\_MODE();

Blynk.run();

timer.run();

}

void TX\_MODE() {

radio.stopListening();

// read the state of the switch into a local variable:

int reading = digitalRead(MANUAL\_START\_STOP);

// If the switch changed, due to noise or pressing:

if (reading != lastButtonState) {

// reset the debouncing timer

lastDebounceTime = millis();

}

if ((millis() - lastDebounceTime) > debounceDelay) {

if (reading != buttonState) {

buttonState = reading;

// only toggle the RELAY if the new button state is HIGH

if (buttonState == HIGH) {

RELAYState = !RELAYState;

}

}

}

lastButtonState = reading;

data.RX\_STARTSTOP\_STATE = RELAYState;

radio.write(&data, sizeof(Data\_Package));

Serial.print(" TX\_state= ");

Serial.println(RELAYState);

}