#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Encoder.h>

#include <EEPROM.h>

LiquidCrystal\_I2C lcd(0x27, 20, 4);

Encoder enc(2, 3);

#define BUTTON\_PIN 4

#define START\_PIN 5

#define OUTPUT\_PIN A0

#define BUZZER\_PIN 6

#define VOLTAGE\_PIN A1

#define PROTECTION\_PIN A2

#define CONTACT\_PIN A3 // Новый пин для контроля замыкания электродов

#define BACKLIGHT\_PIN 9

// Для режима AUTO

bool autoTriggered = false;

unsigned long contactStartTime = 0;

int selected = 0;

int values[3] = {0, 0, 0};

float sValue = 0.1;

bool lightEnabled = true;

byte brightness = 5;

bool autoMode = true;

long lastPos = 0;

bool editMode = false;

bool buttonHeld = false;

unsigned long buttonPressTime = 0;

bool blinkState = true;

bool startButtonState = false;

unsigned long lastDebounceTime = 0;

const unsigned long debounceDelay = 5;

const float voltageMultiplier = 1;

bool dangerDisplayed = false;

unsigned long lastBeepTime = 0;

unsigned long lowVoltageStartTime = 0;

bool lowVoltageProtectionActive = false;

unsigned long protectionVoltageStartTime = 0;

bool protectionActive = false;

void drawScreen();

void drawVoltage(float voltage, bool lowVoltage);

void generatePulse();

void beep() {

digitalWrite(BUZZER\_PIN, HIGH);

delay(50);

digitalWrite(BUZZER\_PIN, LOW);

}

void testDisplay() {

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("Testing Display...");

delay(1000);

lcd.clear();

for (int row = 0; row < 4; row++) {

lcd.setCursor(0, row);

lcd.print("ABCDEFGHIJKLMNOPQRST");

}

delay(1000);

lcd.clear();

for (int row = 0; row < 4; row++) {

lcd.setCursor(0, row);

lcd.print("01234567890123456789");

}

delay(1000);

lcd.clear();

for (int row = 0; row < 4; row++) {

lcd.setCursor(0, row);

lcd.print("####################");

}

delay(1000);

lcd.clear();

}

void selfTest() {

testDisplay();

lcd.setCursor(1, 0);

lcd.print("SYSTEM TESTING...");

delay(500);

lcd.setCursor(2, 1);

lcd.print("EEPROM Check...");

for (int i = 0; i < 3; i++) {

values[i] = EEPROM.read(i);

if (values[i] > 50) values[i] = 0;

}

sValue = EEPROM.read(6) / 10.0;

if (sValue < 0.3 || sValue > 2.0) sValue = 0.3;

delay(500);

lcd.setCursor(2, 2);

lcd.print("Backlight Test...");

lcd.backlight();

delay(500);

lcd.noBacklight();

delay(500);

lcd.backlight();

delay(500);

lcd.setCursor(4, 3);

lcd.print("Beep Test...");

beep();

delay(500);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Voltage Test...");

float protectionVoltage = analogRead(PROTECTION\_PIN) \* (5.0 / 1023.0 \* 3.0);

lcd.setCursor(0, 1);

lcd.print("Voltage: ");

lcd.print(protectionVoltage, 2);

lcd.print("V");

delay(500);

if (protectionVoltage < 10.0 || protectionVoltage > 18.0) {

lcd.setCursor(0, 2);

lcd.print("ERROR: OUT OF RANGE");

lcd.setCursor(0, 3);

lcd.print("REQ: 10-18V");

for (int i = 0; i < 3; i++) {

beep();

delay(500);

}

delay(1000);

} else {

lcd.setCursor(0, 2);

lcd.print("Voltage OK");

}

delay(500);

lcd.clear();

lcd.setCursor(2, 1);

lcd.print("TEST COMPLETE!");

delay(1000);

}

void setup() {

pinMode(BUTTON\_PIN, INPUT\_PULLUP);

pinMode(START\_PIN, INPUT\_PULLUP);

pinMode(OUTPUT\_PIN, OUTPUT);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(VOLTAGE\_PIN, INPUT);

pinMode(PROTECTION\_PIN, INPUT);

pinMode(CONTACT\_PIN, INPUT); // Новый пин как вход

digitalWrite(OUTPUT\_PIN, LOW);

digitalWrite(BUZZER\_PIN, LOW);

pinMode(BACKLIGHT\_PIN, OUTPUT);

lcd.init();

lcd.backlight();

brightness = EEPROM.read(7);

if (brightness > 5) brightness = 5;

analogWrite(BACKLIGHT\_PIN, map(brightness, 0, 5, 0, 255));

lcd.setCursor(3, 0);

lcd.print("BY AKA KASYAN");

lcd.setCursor(2, 1);

lcd.print(">>MyWeld V2.0<<");

lcd.setCursor(5, 2);

lcd.print("SOFT V2.0");

lcd.setCursor(1, 3);

lcd.print("Aka Kasyan YouTube");

delay(3000);

beep();

selfTest();

lcd.clear();

drawScreen();

}

void loop() {

float protectionVoltage = analogRead(PROTECTION\_PIN) \* (5.0 / 1023.0);

bool protectionVoltageOutOfRange = (protectionVoltage > 5.0 || protectionVoltage < 3.0);

if (protectionVoltageOutOfRange) {

if (protectionVoltageStartTime == 0) {

protectionVoltageStartTime = millis();

} else if (millis() - protectionVoltageStartTime >= 1500) {

protectionActive = true;

}

} else {

protectionVoltageStartTime = 0;

protectionActive = false;

dangerDisplayed = false;

}

if (protectionActive) {

if (!dangerDisplayed) {

lcd.clear();

lcd.setCursor(2, 0);

lcd.print("DC 15V PROBLEM!");

lcd.setCursor(6, 1);

lcd.print("DANGER!");

lcd.setCursor(2, 2);

lcd.print("CONTROL VOLTAGE");

lcd.setCursor(2, 3);

lcd.print("ERROR1 <10V >18V");

dangerDisplayed = true;

}

if (millis() - lastBeepTime >= 1000) {

beep();

lastBeepTime = millis();

}

return;

}

// === Режим AUTO: контроль замыкания электродов ===

float contactVoltage = analogRead(CONTACT\_PIN) \* (5.0 / 1023.0);

bool contactDetected = contactVoltage > 2.0; // Порог чувствительности (подстрой по делителю)

if (autoMode && !lowVoltageProtectionActive) {

if (contactDetected) {

if (contactStartTime == 0) {

contactStartTime = millis();

beep(); // Сигнал о замыкании

} else if (!autoTriggered && millis() - contactStartTime >= (unsigned long)(sValue \* 1000)) {

generatePulse();

autoTriggered = true;

}

} else {

contactStartTime = 0;

autoTriggered = false;

}

}

if (digitalRead(BUTTON\_PIN) == LOW) {

if (!buttonHeld) {

buttonPressTime = millis();

buttonHeld = true;

} else if (millis() - buttonPressTime > 2000) {

editMode = !editMode;

beep();

buttonHeld = false;

}

} else {

buttonHeld = false;

}

static unsigned long lastBlinkTime = 0;

if (editMode && millis() - lastBlinkTime > 500) {

blinkState = !blinkState;

lastBlinkTime = millis();

drawScreen();

}

long newPos = enc.read() / 4;

if (newPos != lastPos) {

beep();

if (!editMode) {

if (newPos > lastPos) {

selected = (selected + 1) % (autoMode ? 6 : 5); // В MAN пропускаем S

} else {

selected = (selected + (autoMode ? 5 : 4)) % (autoMode ? 6 : 5);

}

} else {

if (selected < 3) {

if (newPos > lastPos && values[selected] < 50) values[selected]++;

if (newPos < lastPos && values[selected] > 0) values[selected]--;

EEPROM.write(selected, values[selected]);

} else if (selected == 3) {

if (newPos > lastPos && brightness < 5) brightness++;

if (newPos < lastPos && brightness > 0) brightness--;

analogWrite(BACKLIGHT\_PIN, map(brightness, 0, 5, 0, 255));

EEPROM.write(7, brightness);

} else if (selected == 4) {

autoMode = !autoMode;

EEPROM.write(5, autoMode);

if (!autoMode && selected == 5) {

selected = 0;

}

} else if (selected == 5 && autoMode) {

if (newPos > lastPos && sValue < 2.0) sValue += 0.1;

if (newPos < lastPos && sValue > 0.3) sValue -= 0.1;

EEPROM.write(6, (int)(sValue \* 10));

}

}

lastPos = newPos;

drawScreen();

}

float voltage = analogRead(VOLTAGE\_PIN) \* (14 / 1023.0) \* voltageMultiplier;

bool lowVoltage = (voltage < 4.5);

if (lowVoltage) {

if (lowVoltageStartTime == 0) {

lowVoltageStartTime = millis();

} else if (millis() - lowVoltageStartTime >= 1500) {

lowVoltageProtectionActive = true;

}

} else {

lowVoltageStartTime = 0;

lowVoltageProtectionActive = false;

}

drawVoltage(voltage, lowVoltageProtectionActive);

if (!autoMode && digitalRead(START\_PIN) == LOW && !startButtonState && !lowVoltageProtectionActive) {

startButtonState = true;

beep();

generatePulse();

}

if (digitalRead(START\_PIN) == HIGH) {

startButtonState = false;

}

} // <-- Закрытие loop()

void drawScreen() {

for (int i = 0; i < 5; i++) {

lcd.setCursor((i % 2) \* 10, i / 2);

// Стрелка перед выбранным параметром

if (i == selected) {

lcd.print("->");

} else {

lcd.print(" ");

}

// Мигание выбранного параметра

if (i == selected && editMode && blinkState) {

lcd.print(" "); // Очищаем строку при мигании

} else {

if (i < 3) {

lcd.print(i == 0 ? "P1=" : i == 1 ? "T=" : "P2=");

lcd.print(values[i]);

lcd.print("ms ");

} else if (i == 3) {

lcd.print("Light:");

lcd.print(brightness);

lcd.print(" "); // Затираем остатки текста

} else if (i == 4) {

lcd.print("Mode:");

lcd.print(autoMode ? "AUTO " : "MAN ");

}

}

}

// Если режим AUTO — рисуем S, иначе стираем строку

lcd.setCursor(10, 2);

if (autoMode) {

if (selected == 5) {

lcd.print("->");

} else {

lcd.print(" ");

}

if (selected == 5 && editMode && blinkState) {

lcd.print(" "); // Мигание S

} else {

lcd.print("S=");

lcd.print(sValue, 1);

lcd.print("s ");

}

} else {

lcd.print(" "); // Очищаем строку, если режим MAN

}

}

void drawVoltage(float voltage, bool lowVoltage) {

lcd.setCursor(0, 3);

lcd.print("Voltage: ");

lcd.print(voltage, 2);

lcd.print("V ");

if (lowVoltage) {

lcd.print("LOW ");

} else {

lcd.print(" "); // ← 5 пробелов, чтобы гарантированно затереть "LOW"

}

}

void generatePulse() {

digitalWrite(OUTPUT\_PIN, HIGH);

delay(values[0]);

digitalWrite(OUTPUT\_PIN, LOW);

delay(10);

delay(values[1]);

digitalWrite(OUTPUT\_PIN, HIGH);

delay(values[2]);

digitalWrite(OUTPUT\_PIN, LOW);

}