

Załącznik nr 1

Kod źródłowy szkicu `thermostat.ino`

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include "proj.h"

//wyświetlacz
LiquidCrystal lcd(2,3,4,5,6,7);

//Timer
Timer tempMeasureTimer;
Timer timer2;
Timer timer3;

//przyciski
Button buttonPlus(8,PULLUP);
Button buttonMinus(9,PULLUP);
Button buttonAccept(10,PULLUP);

//temperatura
Temperature tValue;
Temperature tDesired(25); //domyślna ustawiona temperatura = 25

// zmienne potrzebne do obliczenia temperatury z ostatnich 5 sekund (10 pomiarów)
const int sum_l = 10;
const float sum_length = 10.00;
float tArray[sum_l];
float tAverage = 0;
int iterator = 0;
float sumT = 0;

//regulator
int k_p = 50;
int k_i = 0.5;
int k_d = 20;
RegPID reg(k_p, k_i, k_d);

//inne
byte menuOption = 1;
byte submenuOption = 1;
Button *recentlyPressed;

//RS485
const byte rxPin = 0;
const byte txPin = 1;
const byte receiveEnablePin = 11;
const byte dataEnablePin = 12;
SoftwareSerial rs485(rxPin, txPin);
int psAdress = 0; //adres zasilacza
String psAdressString = "000";

//Termopara z MAX6675
const byte soPin = A5;
const byte csPin = A4;
const byte sckPin = A3;
MAX6675 thermocouple(sckPin, csPin, soPin);
```

```

//Zasilacz Manson2405
Manson2405 powerSupply;
int voltage = 0, current = 40;
int maxVoltage = 60; //wartości napięcia w woltach = voltage/10; wartości prądu w amperach = current/100;

void setup() {
    timer2.threshold = 100;

    //ustawienia rs485
    rs485.begin(9600);
    pinMode(receiveEnablePin,OUTPUT);
    pinMode(dataEnablePin,OUTPUT);
    digitalWrite(receiveEnablePin,LOW);
    digitalWrite(dataEnablePin,HIGH);

    //rozpoczęcie pracy ekranu
    lcd.begin(16,2);

    lcd.clear();
    delay(500);

    // Zbieranie pomiarów temperatury przez ok. 5 sekund
    for (int i = sum_length - 1 ; i >= 0 ; i--){
        lcd.setCursor(0,0); lcd.print("Czekaj...");
        lcd.setCursor(0,1); lcd.print(i/2);
        tArray[i] = tValue.getTempValue(thermocouple);
        lcd.setCursor(10,0);
        lcd.print(tArray[i]);
        delay(500);
        sumT += tArray[i];
    }
    lcd.setCursor(10,0);
    lcd.print(sumT/sum_length);
    delay(2000);
    lcd.clear();
}

void loop()
{
    if(timer3.stepTimer(500)){
        tArray[iterator] = tValue.getTempValue(thermocouple);
        iterator++;
    }
    if(iterator == sum_length) {
        iterator = 0;
    }
    sumT = 0;
    for(int i = 0; i < sum_length; i++){
        sumT += tArray[i];
    }
    tAverage = sumT/sum_length;

    boolean sTimer = tempMeasureTimer.stepTimer(500);
    if(sTimer) {
        //BŁĄD - nie ma podłączonej termopary
        while(thermocouple.readCelsius() == 0 || isnan(thermocouple.readCelsius()) ) {
            lcd.setCursor(0,0);
            lcd.print("BLAD! Sprawdź");
            lcd.setCursor(0,1);
            lcd.print("termopare");
            delay(2000);
            lcd.clear();
        }
    }
}

```

```

if(buttonAccept.uniquePress()) {
    menuOption++;
    if(menuOption > 3) menuOption = 1;
    lcd.clear();
}
switch(menuOption) {
    case 1: { //ustawienia temperatury
        lcd.setCursor(0,0);
        lcd.print("USTAW");
        if(buttonPlus.uniquePress()) {
            displayTemp(tDesired.set( tDesired.value() + 0.5 ), lcd, 1);
            recentlyPressed = &buttonPlus;
        }

        if (buttonMinus.uniquePress()) {
            displayTemp(tDesired.set( tDesired.value() - 0.5 ), lcd, 1);
            recentlyPressed = &buttonMinus;
        }
        unsigned long timeP = recentlyPressed -> timePressed();

        if( timeP > 1000 && timeP % 200 < 10) {
            if(recentlyPressed == &buttonPlus) tDesired.set( tDesired.value() + 0.5 );
            if(recentlyPressed == &buttonMinus) tDesired.set( tDesired.value() - 0.5 );
            displayTemp(tDesired.value(), lcd, 1);
        }
        lcd.setCursor(0,1);
        if(sTimer) {
            lcd.print((String)(tAverage)+"    -/+");
            displayTemp(tDesired.value(),lcd,1);
        }
        break;
    }
    case 2: { // ustawienie zasilacza
        if(timer2.stepTimer(1000)) {
            lcd.clear();
            lcd.print("Opcje");
            lcd.setCursor(0,1);
            lcd.print("zasilacza (+)");
        }
        if(buttonPlus.uniquePress()) { // submenu
            int endLoop = 137;
            lcd.clear();
            lcd.print("SUBMENU");
            int submenuOption = 0;
            while(submenuOption != endLoop) {
                //przełączanie między submenu
                if(buttonAccept.isPressed()) {
                    submenuOption++;
                    if(submenuOption>2) submenuOption = 1;
                    lcd.clear();
                    switch(submenuOption) {
                        case 1: // rozpoczęcie sesji
                            lcd.clear();
                            lcdPrint(lcd,"Start sesji",0);
                            lcdPrint(lcd,"(+)", 1);
                            break;

                        case 2:
                            lcd.setCursor(0,0);
                            lcd.print("Koniec sesji");
                            lcdPrint(lcd,"(+)",1);
                            break;
                        default: break;
                    }
                }
                delay(500);
            }
        }
    }
}

```

```

//dzialanie submenu
switch(submenuOption) {
    case 1: { //rozpoczęcie sesji
        if(buttonPlus.isPressed()){ //submenu wybor adresu
            lcd.clear();
            lcdPrint(lcd,"Adres: ",0);
            while(!buttonAccept.uniquePress()) {
                //konwersja na string
                String psAdressString = (String)psAdress;
                if(psAdress<10) psAdressString = "0" + psAdressString;
                //wyswietlanie adresu
                lcdPrint(lcd,psAdressString,1);

                if(buttonPlus.uniquePress()) {
                    psAdress++;
                    if(psAdress>31) psAdress = 31;
                }
                if(buttonMinus.uniquePress()) {
                    psAdress--;
                    if(psAdress<0) psAdress = 0;
                }
            }

            powerSupply.startSession(rs485, psAdress);
        }
        break;
    }

    case 2: { //koniec sesji
        if(buttonPlus.uniquePress())
            powerSupply.endSession(rs485, psAdress);
    }

    default: break;
}

if(buttonAccept.timePressed() > 1000) submenuOption = endLoop;
if(menuOption == endLoop) break;
}
}
break;
}

case 3: { // regulacja temperatury
    psAdressString = (String)psAdress;
    int endLoop = 137;
    if(psAdress<10) psAdressString = "0" + psAdressString;
    //wyswietlenie komunikatu
    if(timer2.stepTimer(1000)) lcd.clear();
    lcdPrint(lcd,"START/STOP",0);
    lcdPrint(lcd,"(+)",1);

    if(buttonPlus.uniquePress()) { //przycisk "plus" - rozpoczęcie pracy
        lcd.clear();
        submenuOption++;
        if(submenuOption > 2) submenuOption = 1;
        switch(submenuOption) {
            case 1: { //grzanie wyłączone
                lcd.clear();
                lcdPrint(lcd,"START/STOP",0);
                lcdPrint(lcd,"(+)",1);
            }
            break;

            case 2:{ //włączone

```

```

boolean safetyCondition = false; //warunek bezpieczeństwa
lcd.setCursor(0,0);
lcd.print("V:");
lcd.setCursor(10,0);
lcd.print("U:"+(String)(tDesired.value()));

lcd.setCursor(10,1);
lcd.print("T:");

//przekształcenie wartości prądu na String
String currString;
if(current < 10) currString = "00" + (String)current;
else if(current < 100) currString = "0" + (String)current;
else if(current < 1000) currString = (String)current;
else currString = "0";

//wysłanie komendy
powerSupply.sendCommand(rs485, "CURR"+psAdressString+currString);
Timer timerReg;
while(!buttonAccept.isPressed()) {
    if(timerReg.stepTimer(500)) {

        tArray[iterator] = tValue.getTempValue(thermocouple);
        iterator++;
        if(iterator == sum_length) {
            iterator = 0;
        }
        sumT = 0;
        for(int i = 0; i < sum_length; i++){
            sumT += tArray[i];
        }
        tAverage = sumT/sum_length;

        float temperature = tAverage;
        displayTemp(temperature, lcd, 1);

        //obliczone napięcie
        voltage = maxVoltage*(reg.regulator(tDesired.value(), temperature))/100.00;
        lcd.setCursor(2,0);
        lcd.print("000");
        String voltageString;
        if(voltage < 10) {
            lcd.setCursor(4,0);
            voltageString = "00" + (String)voltage;
        }
        else if(voltage < 100) {
            lcd.setCursor(3,0);
            voltageString = "0" + (String)voltage;
        } else {
            lcd.setCursor(2,0);
            voltageString = (String)voltage;
        }
        lcd.print(voltage);
        powerSupply.sendCommand(rs485, "VOLT"+psAdressString+voltageString);

        //warunek bezpieczeństwa
        if(temperature > tDesired.value() + 10) safetyCondition = true;

        //zabezpieczenie
        while( !isnan(thermocouple.readCelsius()) || safetyCondition) {
            lcd.setCursor(0,0);
            lcd.print("BLAD!!! Sprawdz");
            lcd.setCursor(0,1);
            if(!safetyCondition)
                lcd.print("termopare");
            else lcd.print("temperature");
        }
    }
}

```

```

        delay(2000);
        lcd.clear();
        powerSupply.sendCommand(rs485, "VOLT"+psAdressString+"000");
        delay(10);
        powerSupply.sendCommand(rs485, "CURR"+psAdressString+"000");
        if(tValue.getTempValue(thermocouple) < tDesired.value() + 5 ) safetyCondition = false;
    }
}
}
powerSupply.sendCommand(rs485, "VOLT"+psAdressString+"000");
delay(10);
powerSupply.sendCommand(rs485, "CURR"+psAdressString+"000");
}
break;

default: break;
}
}
break;
}
default: break;
}
}

```