**Tema 2-Mita Alexandra, Paraschiv Raluca**

#include <LiquidCrystal.h>

#define TP36\_SENSOR\_CHANNEL 0 // Setare pin senzor de temp A0

#define ADC\_REF\_VOLTAGE 5.0 // Tensiune de referinta 5V

// Setare pini LCD

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

// Declarare LCD

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

// Pentru timp

volatile unsigned long secunde = 0;

unsigned int ore = 0;

unsigned int minute = 0;

unsigned int sec = 0;

// Variabilă pentru temperatura afișată

float lastTemp = -1000; // Temperatură inițială imposibilă pentru a forța prima afișare

// Functie initializare ADC

void init\_adc() {

// Set the ADC voltage reference to AVCC

ADMUX |= (1 << REFS0);

// Set the ADC clock prescaler to 128

ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);

// Enable the ADC

ADCSRA |= (1 << ADEN);

}

// Functie citire date din ADC

float read\_temperature() {

ADMUX &= 0xF0; // Setarea pinilor din ADMUX pentru citire

ADMUX |= TP36\_SENSOR\_CHANNEL; // Setarea pinului la care este legat senzorul de temperatura (A0)

// Start the ADC conversion

ADCSRA |= (1 << ADSC);

// Wait for the conversion to complete

while (ADCSRA & (1 << ADSC)) {}

// Get the ADC result and calculate the temperature

uint16\_t adc\_value = ADC;

// Transformare valoare analogica in tensiune

float voltage = (float) adc\_value \* ADC\_REF\_VOLTAGE / 1024.0;

// Transformare din tensiune in grade celsius (pentru TMP36, 10mV/°C)

float temperature = (voltage - 0.5) \* 100.0;

return temperature;

}

// Functie initializare Timer1

void init\_timer1() {

// Setăm Timer1 în modul CTC (Clear Timer on Compare Match)

TCCR1A = 0;

TCCR1B = (1 << WGM12) | (1 << CS12) | (1 << CS10); // Prescaler 1024

OCR1A = 15624; // 16MH/1024

TIMSK1 = (1 << OCIE1A); // Activăm întreruperea la compararea cu OCR1A

}

// Interrupție Timer1 la fiecare secundă

ISR(TIMER1\_COMPA\_vect) {

// Incrementăm secundele la fiecare întrerupere de 1 secundă

secunde++;

sec = secunde % 60;

minute = (secunde / 60) % 60;

ore = (secunde / 3600) % 24;

}

void setup() {

lcd.begin(16, 2); // Inițializăm LCD-ul

init\_adc(); // Inițializăm ADC-ul

init\_timer1(); // Inițializăm Timer1

Serial.begin(9600); // Inițializăm portul serial

sei(); // Activăm întreruperile globale

// Afișăm structura inițială pe LCD

lcd.setCursor(0, 0);

lcd.print("Temperatura= ");

lcd.setCursor(0, 1);

lcd.print("00:00:00");

}

void loop() {

// Citim valoarea temperaturii

float tmp = read\_temperature();

// Actualizăm temperatura doar dacă s-a schimbat semnificativ

if (abs(tmp - lastTemp) > 0.1) {

lcd.setCursor(13, 0);

lcd.print(" "); // Ștergem temperatura anterioară

lcd.setCursor(13, 0);

lcd.print(tmp);

lastTemp = tmp;

}

// Afișăm ora curentă pe LCD

lcd.setCursor(0, 1);

if (ore < 10) lcd.print("0");

lcd.print(ore);

lcd.print(":");

if (minute < 10) lcd.print("0");

lcd.print(minute);

lcd.print(":");

if (sec < 10) lcd.print("0");

lcd.print(sec);

delay(200); // Mică întârziere pentru stabilitate

}

<https://www.tinkercad.com/things/dVYca7i99He-proiectul-meu/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard&sharecode=JgYJsWEsDiMvdAUazAG8J-oOk5p6z6i-pIWZHusFYVo>