Министерство Образования и Исследований Республики Молдова

Технический Университет Молдовы

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Программной Инженерии и Автоматики

**Лабораторная работа №5**

по предмету «Интернет вещи»

Выполнил: ст. гр. TI-196

Н. Шарафудинов

Проверил: А. Бырназ

Кишинев 2022

**Тема:** Работа с двигателями и драйверами.

**Задание:**

Описание проблемы:

1. Создать приложение на базе микроконтроллера, которая будет реализовывать системы управления для:

a) Управление в разомкнутом контуре светодиода;

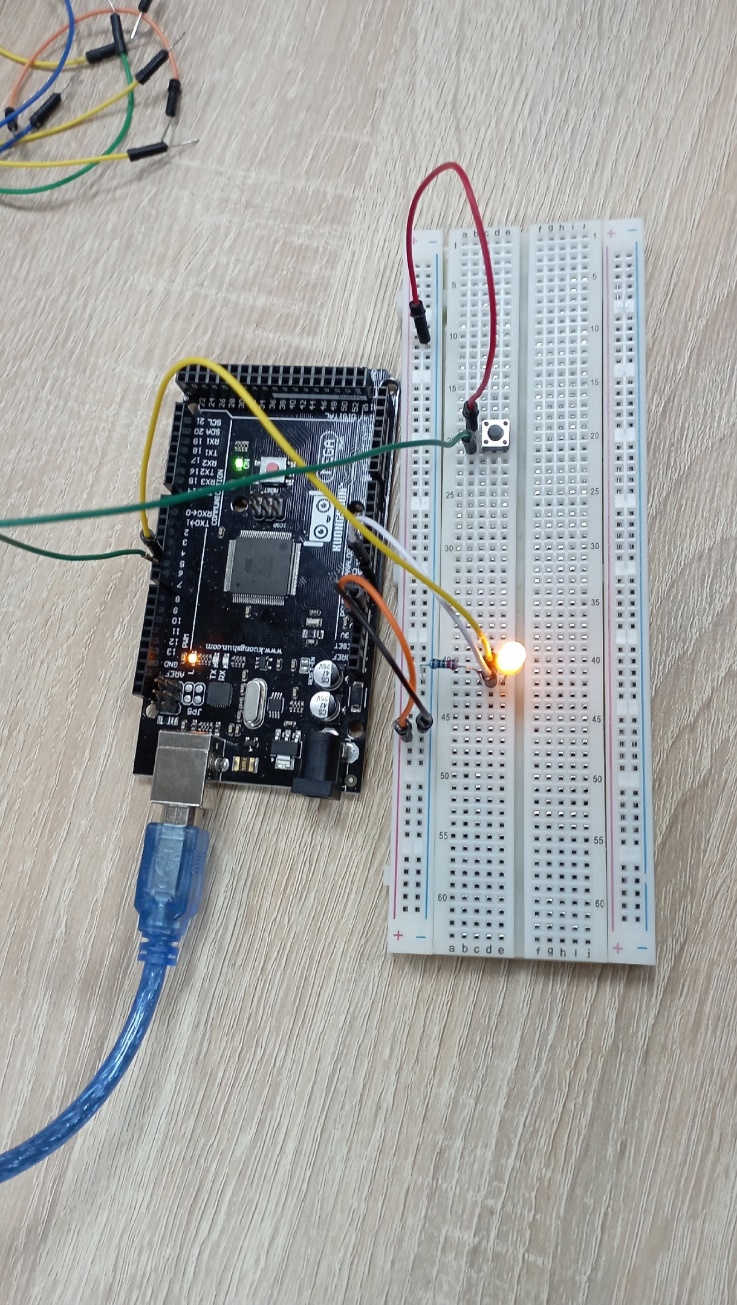
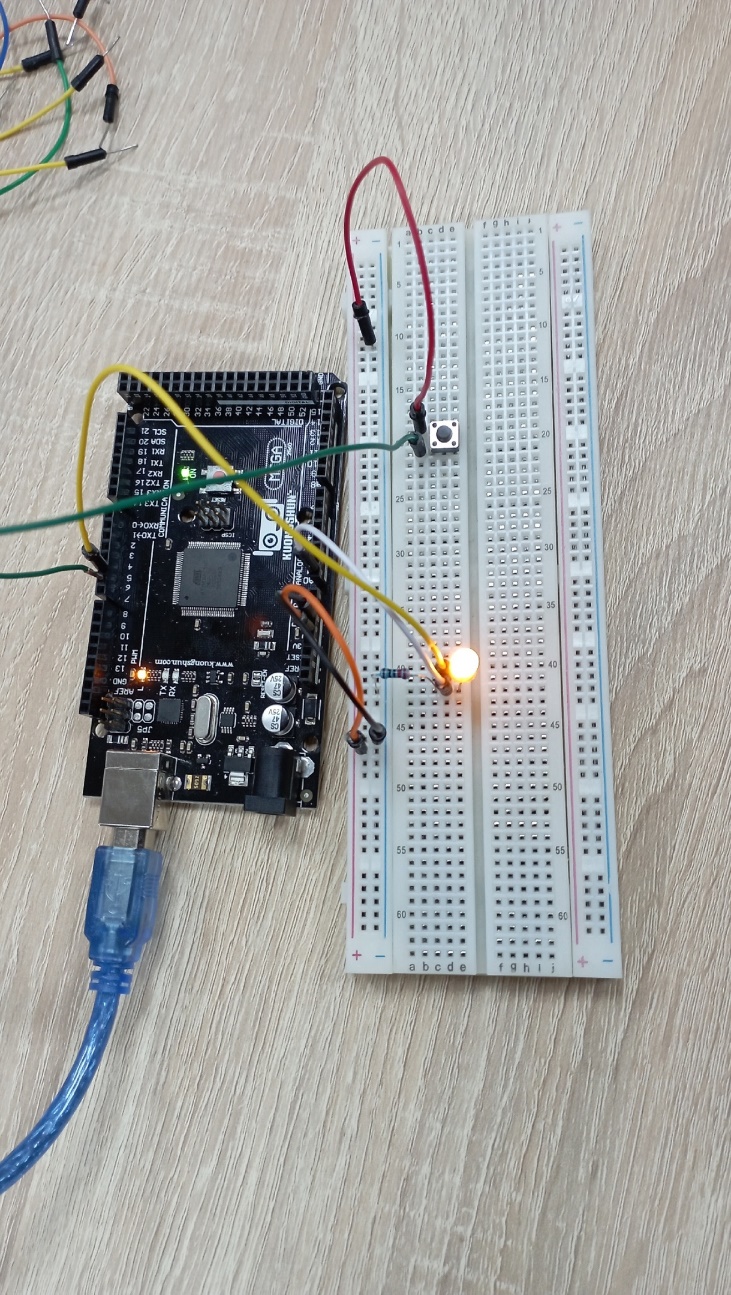
b) Контроль температуры или влажности с применением метода управления ВКЛ / ВЫКЛ с гистерезисом через релейное срабатывание;

c) Управление скоростью двигателя с применением метода ПИД, с энкодером в качестве датчика и драйвером L298 для подачи питания на двигатель. ПРИМЕЧАНИЕ: В пункте (b) вы также можете выбрать другой параметр управления приводом, с ограничением что будет иметь разрешение не менее 8 бит.

2. Заданное значение (контрольное значение) будет установлено из одного из источников по вашему выбору. - потенциометр - две кнопки вверх / вниз - энкодер - клавиатура - последовательный интерфейс

3. Заданное значение и текущее значение будут отображаться на ЖК-дисплее

**Пункт А**

**Вывод**

В данной лабораторной работе я познакомился с работой системы управления. ПИД-регулирование. Так же было использование управлением лампочкой на уровне вольтажа.

**Приложение**

**Пункт А**

#include <Arduino.h>

#define LED\_PIN 5

#define BTN\_PIN 6

void setup()

{

  Serial.begin(9600);

  pinMode(LED\_PIN, OUTPUT);

  pinMode(BTN\_PIN, INPUT\_PULLUP);

}

static int btn\_state = 1;

static int led\_state = 0;

void loop()

{

  int check = digitalRead(BTN\_PIN);

  if (!check && check != btn\_state)

  {

    led\_state = !led\_state;

    digitalWrite(LED\_PIN, led\_state);

    int vol = analogRead(A0);

    if (vol > 300)

    {

      Serial.println("Lampa gorit!!!!!!!!!!!!!!!!!!!!!!");

    }

    else

    {

      Serial.println("Lampa ne gorit !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!?");

    }

  }

  btn\_state = check;

}

**Пункт B**

#include "stdio.h"

#include "termoSensor.h"

unsigned long time;

float termoStandard = 0;

int gestStep = 0;

int enB = 3;

int in3 = 5;

int in4 = 4;

int GL = 10;

int RL = 11;

int BL = 12;

int incB = 8;

int decB = 7;

static FILE serial\_stdinout = { .buf = NULL, .unget = 0, .flags = \_FDEV\_SETUP\_RW, .size = 0, .len = 0, .put = serial\_putchar, .get = serial\_getchar, .udata = 0 };

void setup() {

  Serial.begin(9600);

  stdout = &serial\_stdinout;

  stdin = &serial\_stdinout;

  stderr = &serial\_stdinout;

  pinMode(enB, OUTPUT);

  pinMode(in3, OUTPUT);

  pinMode(in4, OUTPUT);

  pinMode(GL, OUTPUT);

  pinMode(RL, OUTPUT);

  pinMode(BL, OUTPUT);

  pinMode(decB, INPUT\_PULLUP);

  pinMode(incB, INPUT\_PULLUP);

  digitalWrite(in3, LOW);

  digitalWrite(in4, LOW);

}

void loop() {

  BtnCheck();

  TermoInfo();

}

void TermoInfo() {

  // Termo sensor data

  auto result = TermoHumiditySensor();

  // init delay

  if(!initTermo(result.\_t)) { delay(100); }

  // work just after init

  if (initTermo(result.\_t) && gestStep) {

    float gestInc = termoStandard + gestStep;

    float gestDec = termoStandard - gestStep;

    float currTerm = result.\_t;

    // Serial.print(gestInc);

    // Serial.print(", ");

    // Serial.print(termoStandard);

    // Serial.print(", ");

    // Serial.print(gestDec);

    // Serial.print(", ");

    // Serial.println(currTerm);

    printf("%d,",  (int)gestInc);

    printf("%d,",  (int)termoStandard);

    printf("%d,",  (int)gestDec);

    printf("%d\n", (int)currTerm);

    if(currTerm == termoStandard) {

      RGB(0,0,1);

    }

    if(currTerm > gestInc){

      motorStart();

    }

    if(currTerm < gestDec){

      motorEnd();

    }

  }

  else

  {

    Serial.println("Push button ! ! !");

  }

}

bool initTermo(float term) {

  return termoStandard == 0 ? termoStandard = term : termoStandard;

}

void motorStart() {

  digitalWrite(in3, HIGH);

  digitalWrite(in4, LOW);

  RGB(0,1,0);

}

void motorEnd() {

  digitalWrite(in3, LOW);

  digitalWrite(in4, LOW);

  RGB(1,0,0);

}

void RGB(int R, int G, int B) {

    digitalWrite(RL, R);

    digitalWrite(GL, G);

    digitalWrite(BL, B);

}

void BtnCheck() {

  int btnDelay = 100;

  if(!digitalRead(incB)) {

    if (millis() - time > btnDelay) {

      time = millis();

      gestStep++;

    }

  }

  if(!digitalRead(decB)) {

    if (millis() - time > btnDelay) {

      time = millis();

      if(gestStep > 1)

        gestStep--;

    }

  }

}

**TermoSensor.h**

#pragma once

#include "DHT.h"

#define DHTPIN A0

#define DHTTYPE DHT11

struct result {float \_h; float \_t; float \_f; float \_hi; char \*\_err;};

result TermoHumiditySensor();

**TermoSensor.cpp**

#include "termoSensor.h"

DHT dht(DHTPIN, DHTTYPE, 6);

unsigned long timing;

result TermoHumiditySensor(){

  dht.begin();

  float h, t, f, hi;

  char err [] ="nothing";

  while(millis() - timing <= 1000){}

  timing = millis();

  h = dht.readHumidity();

  t = dht.readTemperature();

  f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {

    for(char& c : err) {

      c='\0';

    }

    strcat(err, "Failed to read from DHT sensor!");

    return;

  }

  hi = dht.computeHeatIndex(f, h);

  return result {h, t, f, hi, err};

}

**Пункт С**