|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Мытищинский филиал**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Космический

КАФЕДРА «Прикладная математика, информатика и вычислительная техника» К3-МФ»

**Лабораторная работа**

*ПО ДИСЦИПЛИНЕ:*

***Программирование микроконтроллеров***

***НА ТЕМУ:***

***Создание дистанционно управляемого\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_законченного устройства\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_К3-53Б\_\_  **\_\_\_\_\_\_\_\_\_\_** Цветков Юрий Алексеевич

(Группа) (Подпись, дата) (И.О.Фамилия)

Студент \_\_К3-53Б\_\_  **\_\_\_\_\_\_\_\_\_\_** Наумов Сергей Алексеевич

(Группа) (Подпись, дата) (И.О.Фамилия)

#### Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ Чернышов Александр Викторович

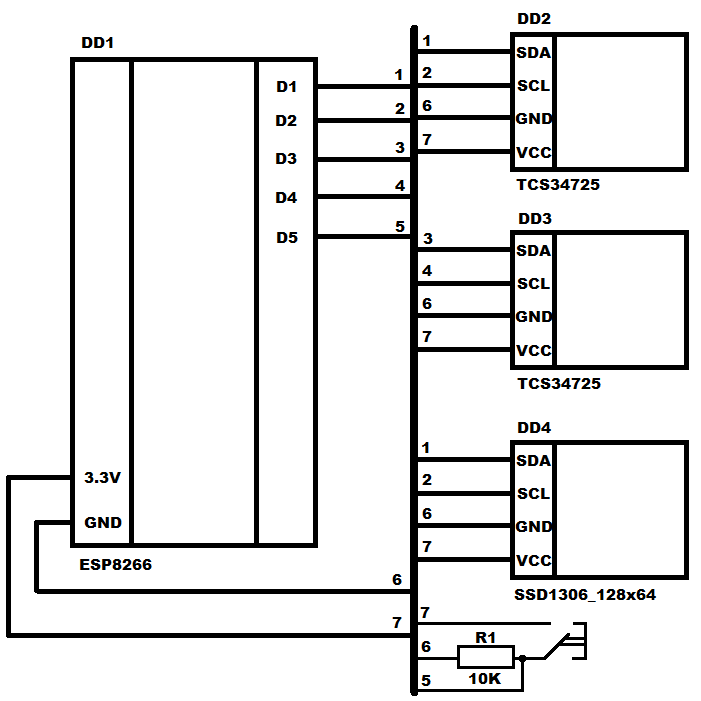
(Подпись, дата) (И.О.Фамилия)

*2021 г.*

***Задание***

Разработать прошивку микроконтроллера мобильного прибора, который измеряет параметры падающего и отражённого света.

***Схема***



***Код программы для ESP8266***

#include <Adafruit\_TCS34725.h>

#include <GyverOLED.h>

#include <Wire.h>

#include <ESP8266WiFi.h>

#define SDA1 4

#define SCL1 5

#define SDA2 0

#define SCL2 2

#define buttonPin 14

const char\* ssid = "iPhone12";

const char\* password = "12345678";

bool flag = false;

uint32\_t timer = 0;

int colorTemp1 = 0;

int lux1 = 0;

int colorTemp2 = 0;

int lux2 = 0;

Adafruit\_TCS34725 tcs(TCS34725\_INTEGRATIONTIME\_300MS, TCS34725\_GAIN\_1X);

GyverOLED<SSD1306\_128x64, OLED\_NO\_BUFFER> oled;

IPAddress ip(172, 20, 10, 11); //IP

IPAddress gateway(255, 20, 1, 17);

IPAddress subnet(255, 255, 255, 0);

WiFiServer server(80);

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

void draw()

{

oled.setScale(2);

oled.setCursor(5, 0);

oled.print("Клоп-L(01)");

oled.line(0, 19, 128, 19);

oled.line(0, 35, 128, 35);

oled.line(64, 19, 64, 64);

oled.setScale(1);

oled.setCursor(18, 3);

oled.print("TCS1");

oled.setCursor(84, 3);

oled.print("TCS2");

}

void clearNum()

{

oled.clear(0, 40, 60, 60);

oled.clear(68, 40, 128, 60);

}

void drawNum()

{

oled.setCursor(1,5);

oled.setCursor(5, 5);

oled.print(String(colorTemp1) + " K");

oled.setCursor(5, 6);

oled.print(String(lux1) + " lux");

oled.setCursor(69, 5);

oled.print(String(colorTemp2) + " K");

oled.setCursor(69, 6);

oled.print(String(lux2) + " lux");

}

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

void test\_TCS()

{

Wire.begin(SDA1,SCL1);

if (tcs.begin())

Serial.println("Found sensor 1");

else

{

Serial.println("No TCS34725\_1 found");

while (1);

}

Wire.begin(SDA2, SCL2);

if (tcs.begin())

Serial.println("Found sensor 2");

else

{

Serial.println("No TCS34725\_2 found");

while (1);

}

}

void updateTCS(int SDA, int SCL, int &temp, int &lux)

{

uint16\_t r, g, b, c;

Wire.begin(SDA,SCL);

tcs.begin();

tcs.getRawData(&r, &g, &b, &c);

temp = tcs.calculateColorTemperature\_dn40(r, g, b, c);

lux = tcs.calculateLux(r, g, b);

}

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

bool button(bool &flag, uint32\_t &timer)

{

bool btState = !digitalRead(buttonPin);

if(btState && !flag && millis() - timer > 100)

{

flag = true;

timer = millis();

}

if(!btState && flag && millis() - timer > 100)

{

flag = false;

timer = millis();

return true;

}

return false;

}

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

void getData()

{

updateTCS(SDA2, SCL2, colorTemp2, lux2);

updateTCS(SDA1, SCL1, colorTemp1, lux1);

}

void updateOled()

{

clearNum();

drawNum();

}

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

void setup() {

Serial.begin(115200);

pinMode(buttonPin, INPUT\_PULLUP);

oled.init();

oled.clear();

draw();

delay(1000);

test\_TCS();

//WIFI CONNECT

WiFi.begin(ssid, password);

WiFi.config(ip, gateway, subnet);

server.begin();

Serial.print("\nUse this URL : http://");

Serial.print(ip);

Serial.println("/");

}

void loop()

{

if(button(flag, timer))

{

getData();

updateOled();

}

WiFiClient client = server.available();

if (client)

{

String request = client.readStringUntil('\r');

client.flush();

requests(request, client);

createHTML(client);

}

}

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

void requests(String request, WiFiClient client)

{

if (request.indexOf("GET /get") >= 0)

{

getData();

}

else if(request.indexOf("GET /update") >= 0)

{

getData();

updateOled();

}

}

void createHTML(WiFiClient client)

{

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("");

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

// CSS

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println(".button { background-colsor: #195B6A; border: none; color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #77878A;}</style></head>");

// Web Page Heading

client.println("<body><h1>KLOP-L(01)</h1>");

client.println("<h2>TCS1</h2>");

client.println("<h4>Color temp: "+ String(colorTemp1) + " K</h4>");

client.println("<h4>Lux: "+ String(lux1) + "</h4>");

client.println("<h2>TCS2</h2>");

client.println("<h4>Color temp: "+ String(colorTemp2) + " K</h4>");

client.println("<h4>Lux: "+ String(lux2) + "</h4>");

client.println("<p><a href=\"/get\"><button class=\"button\">get data</button></a></p>");

client.println("<p><a href=\"/update\"><button class=\"button\">get data and update</button></a></p>");

client.println("</body></html>");

client.println();

}

***Описание разработанного прибора***

Данный прибор служит для измерения показаний падающего и отражённого света.

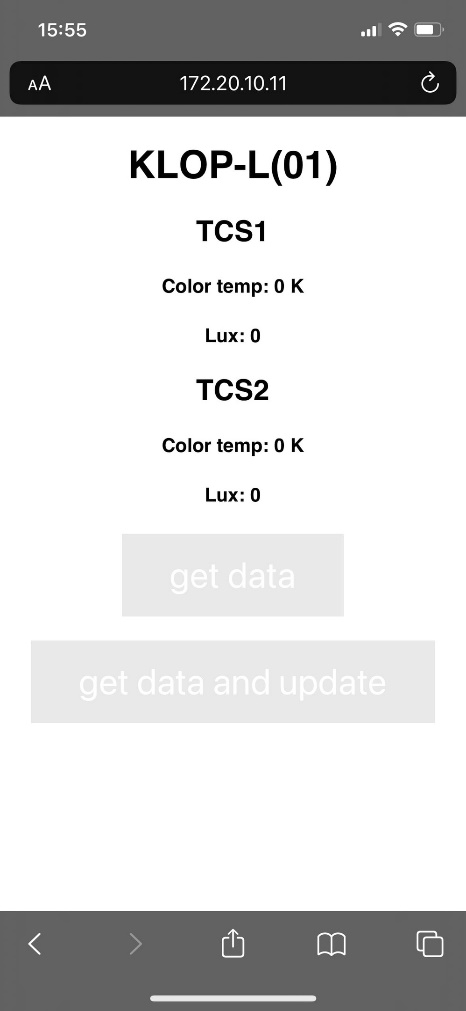
Он даёт возможность узнать цветовую температуру и освещённость по двум датчикам TCS34725.

Реализовано 2 способа получения данных с прибора:

1. Нажатием на кнопку. При этом обновляется информация на OLED экране.
2. Через специально разработанный wifi-клиент с двумя кнопками:

А) get data – получает данные с прибора, НЕ обновляя данные на экране самого прибора.

Б) get data and update - получает данные с прибора И обновляет данные на экране самого прибора.

****

Информация отображается на экране в двух колонках, 1 колонка отвечает за показания падающего света, 2 за показания отражённого.



***Результат***

Разработана прошивка микроконтроллера мобильного прибора, который измеряет параметры падающего и отражённого света.