Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

**ПРАКТИЧЕСКАЯ РАБОТА №7**

**«МДК 01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»**

**Специальность: «09.02.07 Информационные системы и программирование*»***

Квалификация: программист

**Тема: Мониторинг системы. Вывод данных на LCD-дисплей**

|  |  |  |
| --- | --- | --- |
| Проверил преподаватель: | Выполнил: | Выполнил: |
| Чернышев И.В. \_\_\_\_\_\_\_\_ | Студент группы П50-4-21 | Студент группы П50-1-18 |
| «\_\_\_» \_\_\_\_\_\_\_\_ 2023 год | \_\_\_\_\_\_\_\_\_\_\_\_ Игошев Р.В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_И.И. Иванов |
| «\_\_\_» \_\_\_\_\_\_\_\_\_2023 год | | «\_\_\_» \_\_\_\_\_\_\_\_\_2020 год | |

Москва 2023

Цель работы: Собрать схему из LCD-дисплея и I2C-модуля, подключить их к Arduino Uno.

Ссылка на тинкеркад:

<https://www.tinkercad.com/things/45q7uLvpyOJ-pw7?sharecode=x1cApUw4osSBjKcp4NBXajtc8cggIYBy4jH4KJvKC-E>

Начать следует с создания макета. Здесь нужно соединить LCD-экран некоторыми проводами.

Выглядеть макет будет следующим образом:

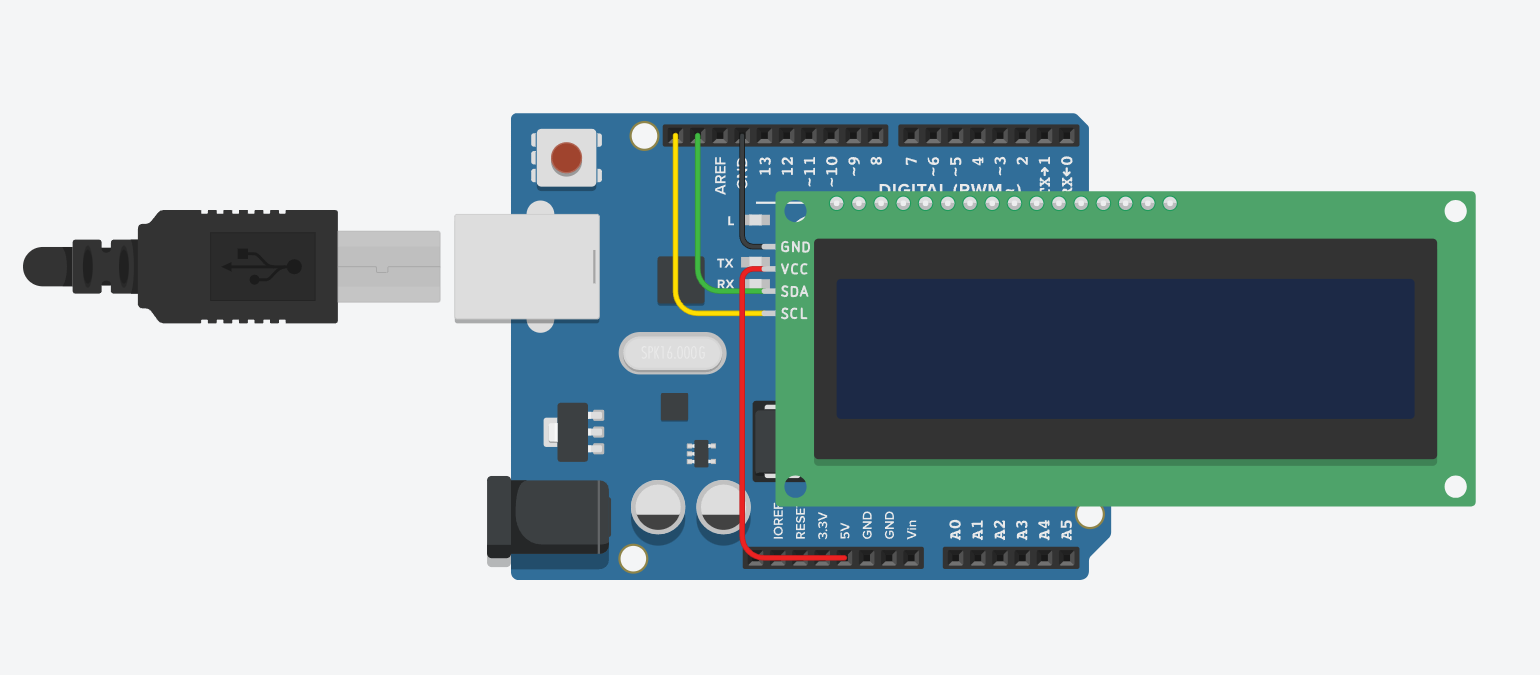


Рисунок 1 – Скриншот из Tinkercad

Снизу же видно код, в котором описана логика.

byte speedMIN = 10, speedMAX = 90, tempMIN = 30, tempMAX = 70;

#define DRIVER\_VERSION 1 // 0 - маркировка драйвера кончается на 4АТ, 1 - на 4Т

#define COLOR\_ALGORITM 0 // 0 или 1 - разные алгоритмы изменения цвета (строка 222)

#define ERROR\_DUTY 90

#define FAN\_PIN 9 // на мосфет вентиляторов

int RECV\_PIN = 11;

#include <string.h> // библиотека расширенной работы со строками

#include <Wire.h> // библиотека для соединения

#include <LiquidCrystal\_I2C.h> // библтотека дислея

#if (DRIVER\_VERSION)

LiquidCrystal\_I2C lcd(0x27, 16, 2);

#else

LiquidCrystal\_I2C lcd(0x3f, 16, 2);

#endif

#define printByte(args) write(args);

#define TEMPERATURE\_PRECISION 9]

byte degree[8] = {0b11100, 0b10100, 0b11100, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000};

byte right\_empty[8] = {0b11111, 0b00001, 0b00001, 0b00001, 0b00001, 0b00001, 0b00001, 0b11111};

byte left\_empty[8] = {0b11111, 0b10000, 0b10000, 0b10000, 0b10000, 0b10000, 0b10000, 0b11111};

byte center\_empty[8] = {0b11111, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111};

byte row8[8] = {0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};

byte row7[8] = {0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};

byte row6[8] = {0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};

byte row5[8] = {0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};

byte row4[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111};

byte row3[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111};

byte row2[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111};

byte row1[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111};

char inData[108]; // массив входных значений (СИМВОЛЫ)

int PCdata[26]; // массив численных значений показаний с компьютера

byte blocks, halfs;

byte index = 0;

int display\_mode = 0;

String string\_convert;

unsigned long timeout, blink\_timer, plot\_timer;

boolean lightState, reDraw\_flag = 1, updateDisplay\_flag, updateTemp\_flag, timeOut\_flag = 1;

int duty, LEDcolor;

int k, b, R, G, B, Rf, Gf, Bf;

byte mainTemp;

byte lines[] = {4, 5, 7, 6};

String perc;

unsigned long sec;

unsigned int mins, hrs;

byte temp1, temp2;

boolean btn1\_sig, btn2\_sig, btn1\_flag, btn2\_flag;

void setup() {

Serial.begin(9600);

lcd.init();

lcd.backlight();

lcd.clear(); // очистить дисплей

PCdata[8] = speedMAX;

PCdata[9] = speedMIN;

PCdata[10] = tempMAX;

PCdata[11] = tempMIN;

}

void loop() {

parsing(); // парсим строки с компьютера

updateDisplay();

}

void parsing() {

while (Serial.available() > 0) {

char aChar = Serial.read();

if (aChar != 'E') {

inData[index] = aChar;

index++;

inData[index] = '\0';

} else

{

char \*p = inData;

char \*str;

index = 0;

String value = "";

while ((str = strtok\_r(p, ";", &p)) != NULL) {

string\_convert = str;

PCdata[index] = string\_convert.toInt();

index++;

}

index = 0;

updateDisplay\_flag = 1;

updateTemp\_flag = 1;

}

timeout = millis();

timeOut\_flag = 1;

}

}

void updateDisplay() {

if (updateDisplay\_flag) {

if (reDraw\_flag) {

lcd.clear();

draw\_labels\_11();

reDraw\_flag = 0;

}

switch (display\_mode) {

case 0: draw\_stats\_11();

break;

case 1: draw\_stats\_12();

break;

case 2: draw\_stats\_21();

break;

case 3: draw\_stats\_22();

break;

case 50: debug();

break;

}

updateDisplay\_flag = 0;

}

}

void draw\_stats\_11() {

lcd.setCursor(4, 0); lcd.print(PCdata[0]); lcd.write(223);

lcd.setCursor(13, 0); lcd.print(PCdata[4]);

if (PCdata[4] < 10) perc = "% ";

else if (PCdata[4] < 100) perc = "%";

else perc = ""; lcd.print(perc);

lcd.setCursor(4, 1); lcd.print(PCdata[1]); lcd.write(223);

lcd.setCursor(13, 1); lcd.print(PCdata[5]);

if (PCdata[5] < 10) perc = "% ";

else if (PCdata[5] < 100) perc = "%";

else perc = ""; lcd.print(perc);

for (int i = 0; i < 2; i++) {

byte line = ceil(PCdata[lines[i]] / 16);

lcd.setCursor(7, i);

if (line == 0) lcd.printByte(1)

else lcd.printByte(4);

for (int n = 1; n < 5; n++) {

if (n < line) lcd.printByte(4);

if (n >= line) lcd.printByte(2);

}

if (line == 6) lcd.printByte(4)

else lcd.printByte(3);

}

}

void draw\_stats\_12() {

lcd.setCursor(13, 0); lcd.print(PCdata[7]);

if (PCdata[7] < 10) perc = "% ";

else if (PCdata[7] < 100) perc = "%";

else perc = ""; lcd.print(perc);

lcd.setCursor(13, 1); lcd.print(PCdata[6]);

if (PCdata[6] < 10) perc = "% ";

else if (PCdata[6] < 100) perc = "%";

else perc = ""; lcd.print(perc);

for (int i = 0; i < 2; i++) {

byte line = ceil(PCdata[lines[i + 2]] / 16);

lcd.setCursor(7, i);

if (line == 0) lcd.printByte(1)

else lcd.printByte(4);

for (int n = 1; n < 5; n++) {

if (n < line) lcd.printByte(4);

if (n >= line) lcd.printByte(2);

}

if (line == 6) lcd.printByte(4)

else lcd.printByte(3);

}

}

void draw\_stats\_21() {

lcd.setCursor(13, 0); lcd.print(duty);

if ((duty) < 10) perc = "% ";

else if ((duty) < 100) perc = "%";

else perc = ""; lcd.print(perc);

lcd.setCursor(3, 1); lcd.print(temp1); lcd.write(223);

lcd.setCursor(11, 1); lcd.print(temp2); lcd.write(223);

byte line = ceil(duty / 16);

lcd.setCursor(6, 0);

if (line == 0) lcd.printByte(1)

else lcd.printByte(4);

for (int n = 1; n < 5; n++) {

if (n < line) lcd.printByte(4);

if (n >= line) lcd.printByte(2);

}

if (line == 6) lcd.printByte(4)

else lcd.printByte(3);

}

void draw\_stats\_22() {

lcd.setCursor(2, 0); lcd.print(PCdata[2]); lcd.write(223);

lcd.setCursor(10, 0); lcd.print(PCdata[3]); lcd.write(223);

char lcd\_time[10];

snprintf(lcd\_time, sizeof(lcd\_time), "%02d:%02d",PCdata[19],PCdata[20]);

char lcd\_date[12];

snprintf(lcd\_date, sizeof(lcd\_date), "%02d/%02d/%04d",PCdata[22],PCdata[23],PCdata[24]);

lcd.setCursor(0, 1);

lcd.print(lcd\_time);

lcd.setCursor(6, 1);

lcd.print(lcd\_date);

}

void draw\_labels\_11() {

lcd.createChar(0, degree);

lcd.createChar(1, left\_empty);

lcd.createChar(2, center\_empty);

lcd.createChar(3, right\_empty);

lcd.createChar(4, row8);

lcd.setCursor(0, 0);

lcd.print("CPU:");

lcd.setCursor(0, 1);

lcd.print("GPU:");

}

void debug() {

lcd.clear();

lcd.setCursor(0, 0);

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

lcd.print(PCdata[j]); lcd.print(" ");

}

lcd.setCursor(0, 1);

for (int j = 6; j < 10; j++) {

lcd.print(PCdata[j]); lcd.print(" ");

}

lcd.setCursor(0, 2);

for (int j = 10; j < 15; j++) {

lcd.print(PCdata[j]); lcd.print(" ");

}

lcd.setCursor(0, 3);

for (int j = 15; j < 18; j++) {

lcd.print(PCdata[j]); lcd.print(" ");

}

}

Вывод: Собрали схему из LCD-дисплея и I2C-модуля, подключили их к Arduino Uno.