**Министерство Образования, Культуры и Исследований**

**Технический университет Республики Молдова**

**Факультет Вычислительной техники, Информатики и Микроэлектроники**

**Департамент программной инженерии и автоматики**

**Отчет**

По дисциплине: Internet of Things

По лабораторной работе №1

Тема: Взаимодействие с пользователем

**Выполнили:**

ст. гр. TI-174 Старицын Д., Изотова В., Хромцов Д., Малая А.

**Проверил:** Gh. Purci

**Кишинев – 2020**

1 Задание

1. Спроектировать приложение на базе MCU, которое бы меняло состояние LED при обнаружении нажатия кнопки.
2. Спроектировать приложение на базе MCU, которое бы откликалось на команды из терминала, используя Serial интерфейс, чтобы выставить состояние LED. Реализовать команды led on для включения и led off для выключения. Система должна отвечать текстовыми сообщениями, подтверждая команду. Для обмена сообщениями через терминал необходимо использовать библиотеку stdio.
3. Спроектировать приложение на базе MCU, которое будет обнаруживать код, введенный с клавиатуры 4х4, проверять код и отправлять сообщение на LCD. Если введен правильный код, то необходимо зажечь зеленый LED, а для неправильного – красный LED.

2 Выполнение задания

Для реализации заданий были использованы:

* Пакет программ для автоматизированного проектирования (САПР) электронных схем Proteus Design Suite.
* Arduino UNO SMD.
* IDE ARDUINO 1.8.13.

2.1 Первое задание (a)

**Схема**

Для выполнения была спроектирована схема, включающая Arduino UNO SMD, LED-GREEN, RESISTOR и BUTTON (рисунок 1).

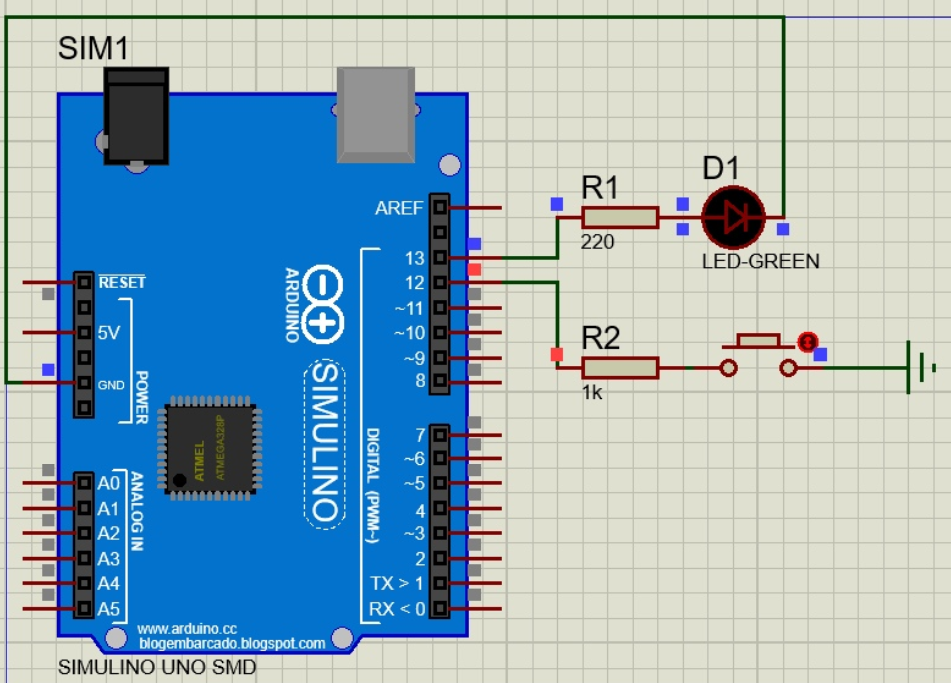


Рисунок - Схема первого задания

**Код файлов**

Код **led.h:**

void LedInit(int LED\_PIN)

{

pinMode(LED\_PIN, OUTPUT);

}

Код **button.h:**

void ButtonInit(int BUTTON\_PIN)

{

pinMode(BUTTON\_PIN, INPUT\_PULLUP);

}

Код **main:**

#include "button.h"

#include "led.h"

#define BUTTON\_PIN 12

#define LED\_PIN 13

#define BUTTON\_ON HIGH

#define BUTTON\_OFF LOW

#define LED\_ON LOW

#define LED\_OFF HIGH

void setup()

{

ButtonInit(BUTTON\_PIN);

LedInit(LED\_PIN);

}

int current\_state = LED\_OFF;

void loop()

{

if (digitalRead(BUTTON\_PIN) == BUTTON\_ON)

{

delay(200);

if (current\_state == LED\_OFF)

{

digitalRead(LED\_PIN, LED\_ON);

}

else

{

digitalRead(LED\_PIN, LED\_OFF);

}

}

}

**Скриншоты работы системы**

Пример работы системы демонстрируется на рисунках 2 и 3.

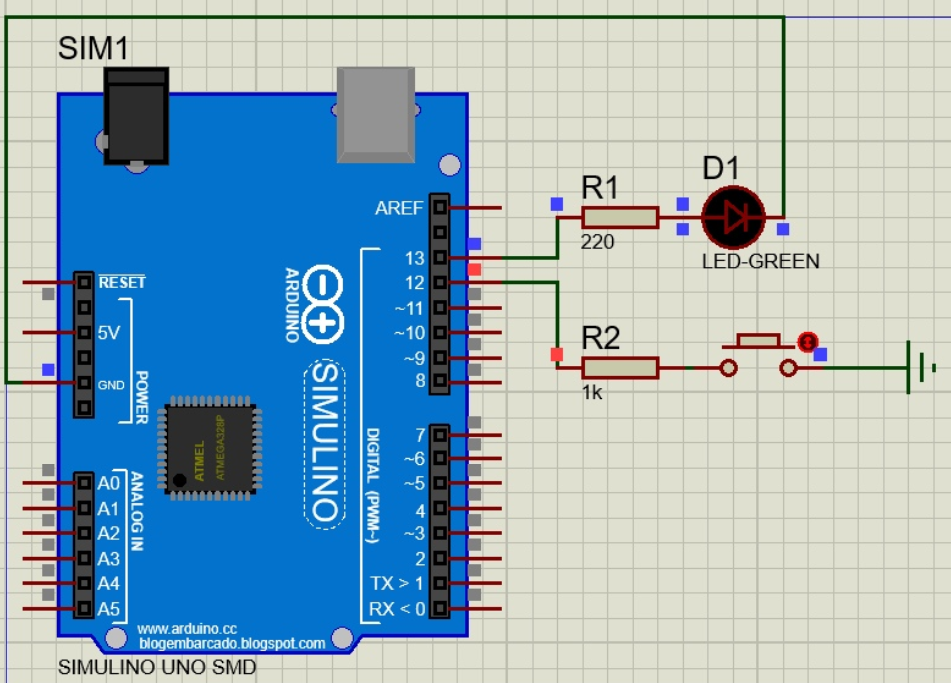


Рисунок - Кнопка нажата нечетное количество раз

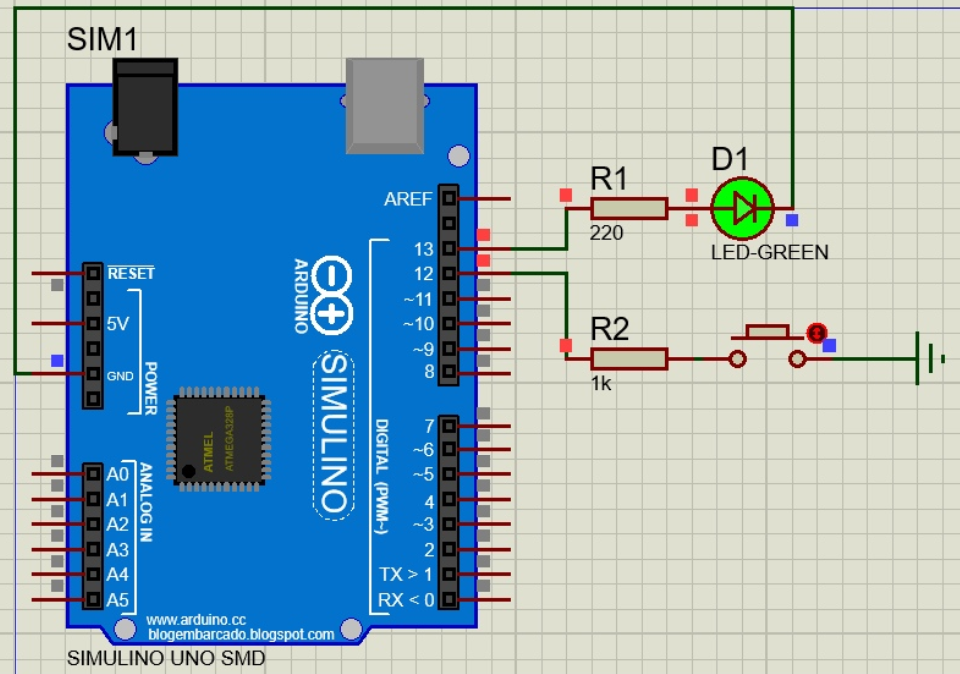


Рисунок - Кнопка нажата четное количество раз

2.2 Второе задание (b)

**Схема**

Для проектирования системы были использованы следующие компоненты (рисунок 4):

* LED BLUE;
* ARDUINO UNO SMD;
* RESISTOR;
* Virtual Terminal.

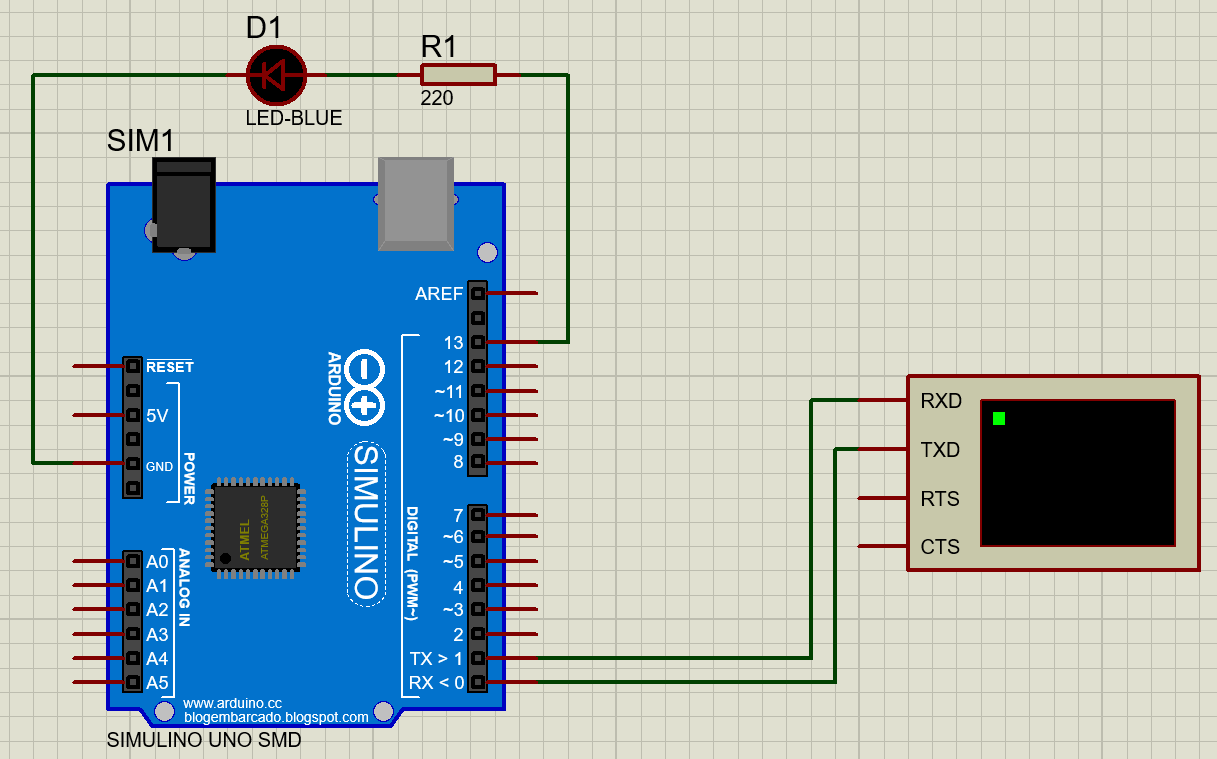


Рисунок - Схема для второго задания

**Код файлов**

Код **serial.h:** (используется для перенаправления терминала на стандартный ввод-вывод):

void my\_putChar(char c, FILE \*f)

{

Serial.write(c);

return c;

}

char my\_GetChar(FILE \*f)

{

while(!Serial.available());

return Serial.read();

}

FILE\* stdio\_init()

{

return fdevopen(my\_putChar, my\_GetChar);

}

Код **led.h**:

void LedInit(int LED\_PIN)

{

pinMode(LED\_PIN, OUTPUT);

}

Код **main**:

#include <string.h>

#include <stdio.h>

#include "led.h"

#include "stdio.h"

#define LED\_PIN 13

#define LED\_ON HIGH

#define LED\_OFF LOW

#define COMMAND\_ON "led on"

#define COMMAND\_OFF "led off"

char\* last\_word = new char[256];

int helping\_index = 0;

int current\_state = LED\_OFF;

void setup()

{

FILE \*my\_stream = stdio\_init();

stdin = stdout = my\_stream;

Serial.begin(9600);

LedInit(LED\_PIN);

}

void scan\_newline()

{

for(helping\_index = 0; helping\_index < 256; ++helping\_index)

{

last\_word[helping\_index] = '\0';

}

char c;

while(!Serial.available());

helping\_index = 0;

scanf("%c", &c);

while ((c != '\r') && (helping\_index < 256))

{

if (c == 8)

{

if (helping\_index != 0)

{

--helping\_index;

last\_word[helping\_index] = '\0';

}

scanf("%c", &c);

continue;

}

last\_word[helping\_index] = c;

++helping\_index;

scanf("%c", &c);

}

}

void beautify()

{

for (int i = 0; (i < 256) && (last\_word[i] != '\0'); ++i)

{

switch (last\_word[i])

{

case 'L': last\_word[i] = 'l'; break;

case 'E': last\_word[i] = 'e'; break;

case 'D': last\_word[i] = 'd'; break;

case 'O': last\_word[i] = 'o'; break;

case 'N': last\_word[i] = 'n'; break;

case 'F': last\_word[i] = 'f'; break;

}

}

}

void loop()

{

printf("@: ");

scan\_newline();

beautify();

if (strcmp(last\_word, COMMAND\_ON) == 0)

{

if (current\_state == LED\_ON)

{

printf("<System>: LED is already on\r");

}

else

{

digitalWrite(LED\_PIN, LED\_ON);

current\_state = LED\_ON;

printf("<System>: Turned LED on\r");

}

}

else if (strcmp(last\_word, COMMAND\_OFF) == 0)

{

if (current\_state == LED\_OFF)

{

printf("<System>: LED is already off\r");

}

else

{

digitalWrite(LED\_PIN, LED\_OFF);

current\_state = LED\_OFF;

printf("<System>: Turned LED off\r");

}

}

else

{

printf("<System>: Unknown command!\r");

}

}

**Скриншоты работы системы**

Пример работы системы изображается на рисунках 5 и 6.

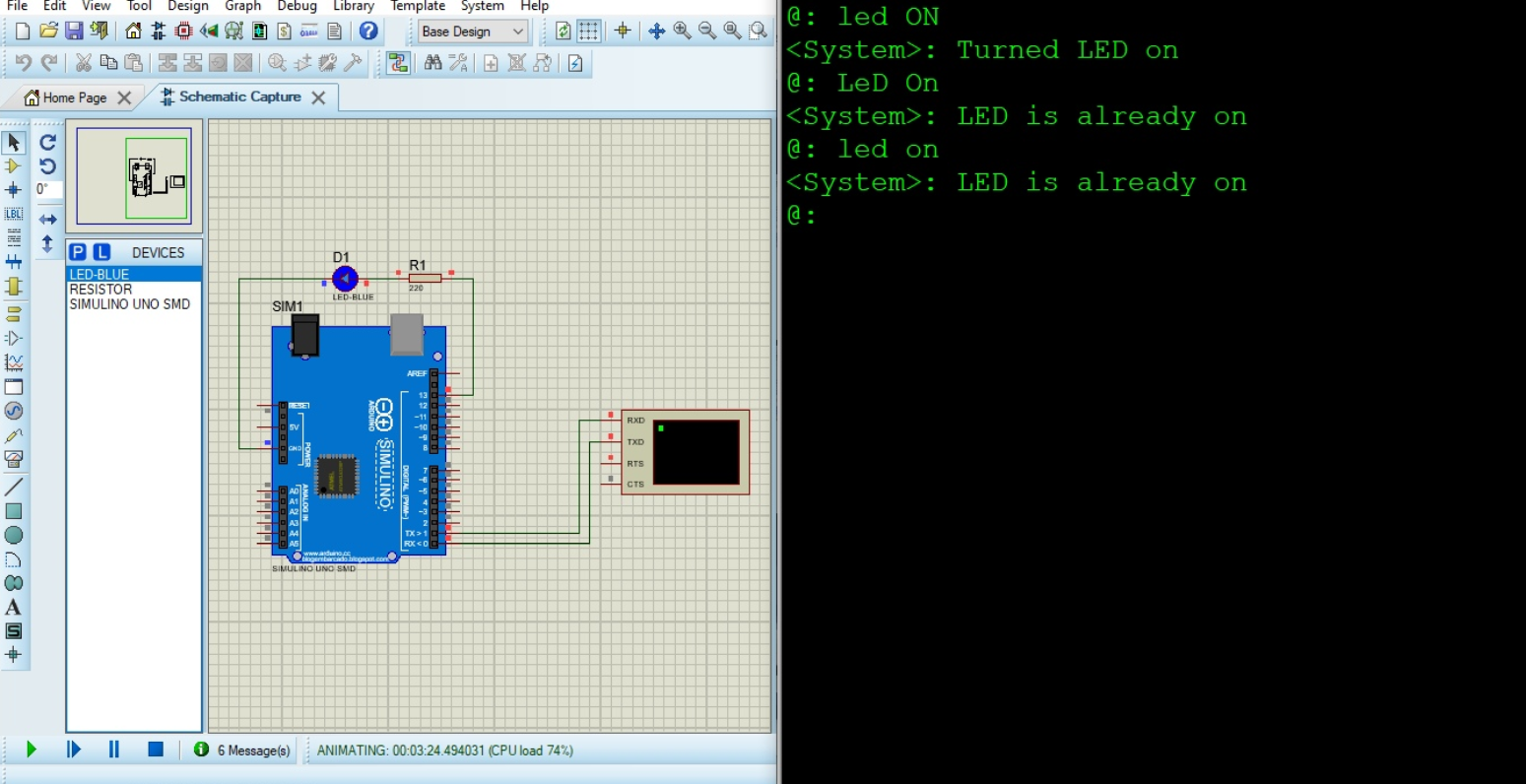


Рисунок - Включение LED командой "led on"

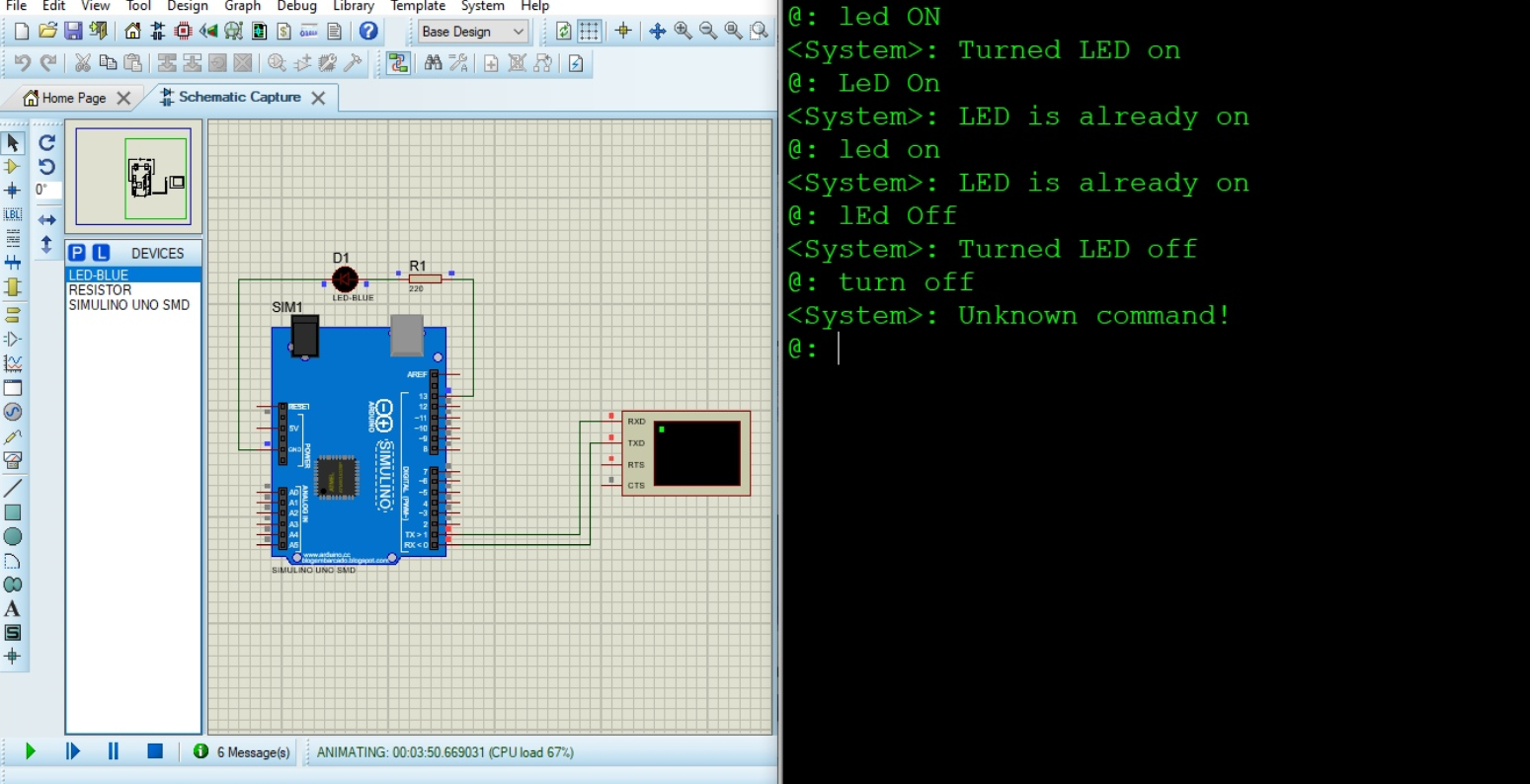


Рисунок - выключение LED командой "led off"

2.3 Третье задание (c)

**Схема**

Для проектирования системы были использованы следующие компоненты (рисунок 7):

* KEYPAD;
* ARDUINO UNO SMD;
* LCD.

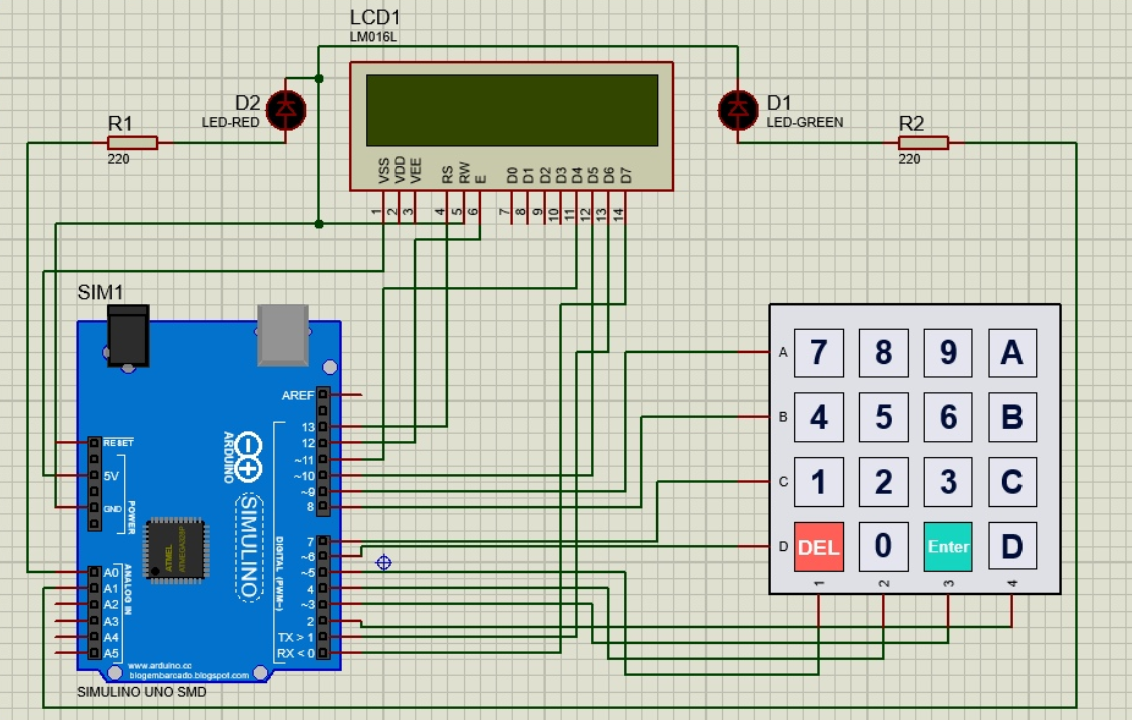


Рисунок 7 - Схема для третьего задания

**Код файлов**

Код **keypad.h:**

#include <Keypad.h>

Keypad \*keypad;

void begin\_keypad(Keypad \*k)

{

keypad = k;

}

char keypad\_get\_char(FILE \*f)

{

char key;

do

{

key = keypad->getKey();

}while (key == 0);

return key;

}

Код **led.h**:

void LedInit(int LED\_PIN)

{

pinMode(LED\_PIN, OUTPUT);

}

Код **lcd.h:**

#include <LiquidCrystal.h>

int LCD\_X;

int LCD\_Y;

LiquidCrystal \*lcd;

void begin\_lcd(LiquidCrystal \*l, int x, int y)

{

lcd = l;

lcd->begin(x, y);

LCD\_X = x;

LCD\_Y = y;

}

void clear\_lcd(const char\* fill\_char = " ")

{

for (int i = 0; i < LCD\_Y; ++i)

{

lcd->setCursor(0, i);

for (int j = 0; j < LCD\_X; ++j)

{

printf(fill\_char);

}

}

lcd->setCursor(0, 0);

}

void lcd\_put\_char(char ch, FILE \*f)

{

lcd->print(ch);

return ch;

}

Код **main**:

#include <stdio.h>

#include "lcd.h"

#include "keypad.h"

#include "led.h"

#define RED\_LED\_PIN A0

#define GREEN\_LED\_PIN A1

#define LED\_ON HIGH

#define LED\_OFF LOW

#define RS 13

#define EN 12

#define D4 11

#define D5 10

#define D6 1

#define D7 0

#define LCD\_X 16

#define LCD\_Y 2

#define ROW\_NUM 4

#define COLUMN\_NUM 4

#define CODE "1337"

#define STRING\_LEN LCD\_X\*LCD\_Y

byte pin\_rows[ROW\_NUM] = {9, 8, 7, 6};

byte pin\_column[COLUMN\_NUM] = {5, 4, 3, 2};

char keys[ROW\_NUM][COLUMN\_NUM] = {

{'7','8','9', 'A'},

{'4','5','6', 'B'},

{'1','2','3', 'C'},

{'\*','0','#', 'D'}

};

char\* string = new char[STRING\_LEN];

int pos\_x = 0;

int pos\_y = 0;

unsigned long time\_del = 0;

void clear\_string()

{

for (int i = 0; i < STRING\_LEN; ++i)

{

string[i] = '\0';

}

}

void reset()

{

clear\_lcd();

pos\_x = 0;

pos\_y = 0;

clear\_string();

}

bool check\_answer()

{

return (strcmp(string, CODE) == 0);

}

void blink\_led(int led\_pin)

{

digitalWrite(led\_pin, LED\_ON);

delay(750);

digitalWrite(led\_pin, LED\_OFF);

}

void right\_answer()

{

blink\_led(GREEN\_LED\_PIN);

}

void wrong\_answer()

{

blink\_led(RED\_LED\_PIN);

}

void keypadEvent(KeypadEvent key);

void setup()

{

keypad = new Keypad(makeKeymap(keys), pin\_rows, pin\_column, ROW\_NUM, COLUMN\_NUM);

begin\_keypad(keypad);

lcd = new LiquidCrystal(RS, EN, D4, D5, D6, D7);

begin\_lcd(lcd, 16, 2);

FILE\* stream = fdevopen(lcd\_put\_char, keypad\_get\_char);

stderr = stdout = stdin = stream;

LedInit(RED\_LED\_PIN);

LedInit(GREEN\_LED\_PIN);

reset();

keypad->addEventListener(keypadEvent);

}

bool is\_hold(char c = '\*', int delay\_millisec = 1500)

{

for(int i = 0; i < 10; ++i)

{

if (!keypad->isPressed(&c))

{

return false;

}

delay(delay\_millisec / 10);

}

return true;

}

void delete\_char()

{

if (pos\_x == 0 && pos\_y == 0)

{

return;

}

string[((pos\_y) \* LCD\_X) + pos\_x - 1] = '\0';

--pos\_x;

if (pos\_x < 0)

{

pos\_x = LCD\_X - 1;

--pos\_y;

}

lcd->setCursor(pos\_x, pos\_y);

printf(" ");

lcd->setCursor(pos\_x, pos\_y);

}

void keypadEvent(KeypadEvent key)

{

if (key == '\*' && keypad->getState() == IDLE)

{

if (millis() - time\_del < 1000)

{

delete\_char();

}

else

{

reset();

}

}

}

void loop()

{

char c;

scanf("%c", &c);

if (c == '\*') // Стирание строки

{

time\_del = millis();

}

else if (c == '#') // Проверка кода

{

if (check\_answer())

{

right\_answer();

}

else

{

wrong\_answer();

}

}

else if ((pos\_x + 1) \* (pos\_y + 1) < STRING\_LEN + 1) // Добавление символа (если есть свободное место)

{

if (pos\_x >= LCD\_X)

{

pos\_x = 0;

pos\_y = pos\_y + 1;

lcd->setCursor(pos\_x, pos\_y);

}

string[((pos\_y) \* LCD\_Y) + pos\_x] = c;

++pos\_x;

printf("%c", c);

}

}

**Скриншоты работы системы**

Пример работы системы изображается на рисунках 8 и 9.

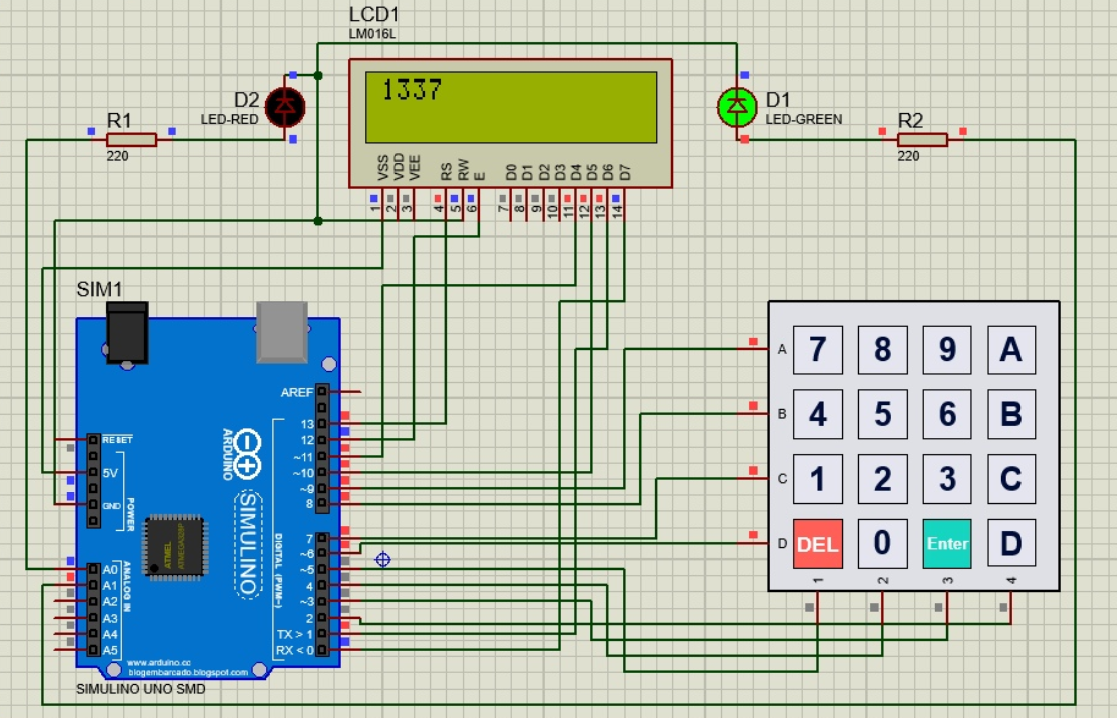


Рисунок 8 - Включение LED-GREEN при правильно введенном коде «1337»

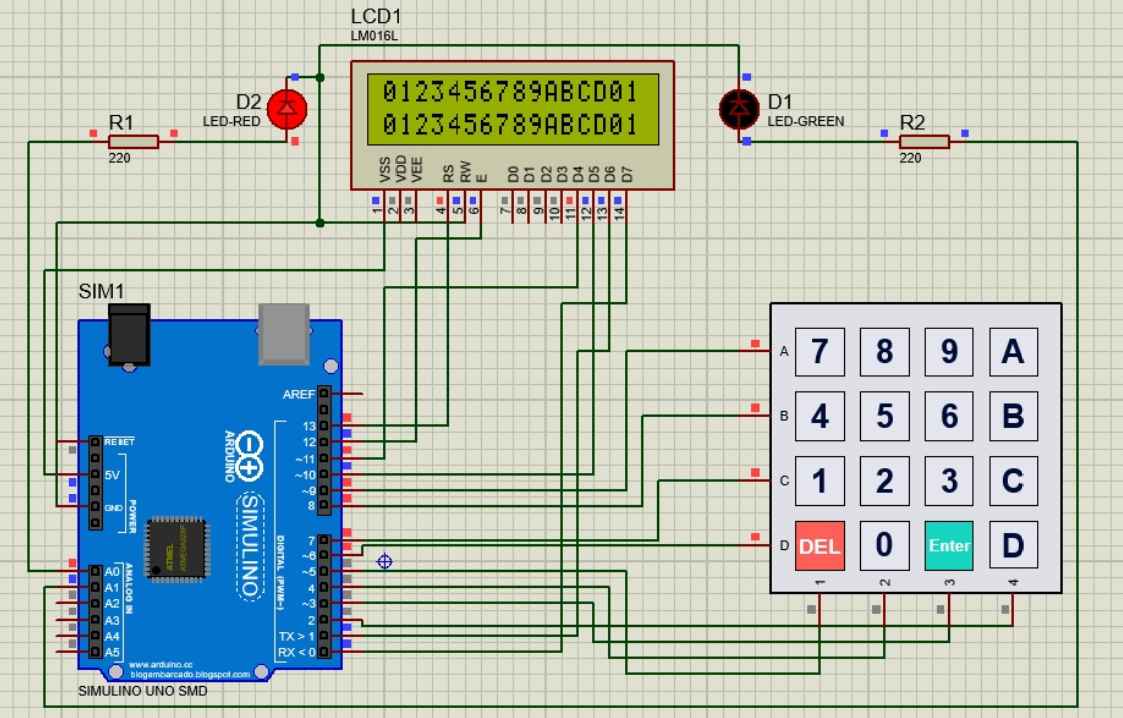


Рисунок 9 - Включение LED-RED при неправильно введенном коде

Заключение

В ходе выполнения лабораторной работы мы изучили систему автоматизированного проектирования Proteus, которая позволяет эмулировать процесс работы Arduino UNO SMD. Мы разработали несколько систем, которые позволяют пользователю взаимодействовать с системами. Таким образом, способов взаимодействия пользователя и различных систем IoT огромное множество и их реализация требует определенного опыта и знаний.