Министерство Образования и Исследований Республики Молдова

Технический Университет Молдовы

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Программной Инженерии и Автоматики

**Лабораторная работа №6**

по предмету «Интернет вещи»

Выполнил: ст. гр. TI-196

Н. Шарафудинов

Проверил: А. Бырназ

Кишинев 2022

**Тема:** Конечные автоматы.

**Задание:**

Описание проблемы: Создайте приложение, которое будет реализовывать конечные автоматы следующим образом:

1. Разрабатывать приложения для конечного автомата кнопка-светодиод

2. Разрабатывать приложения для конечного автомата светофор

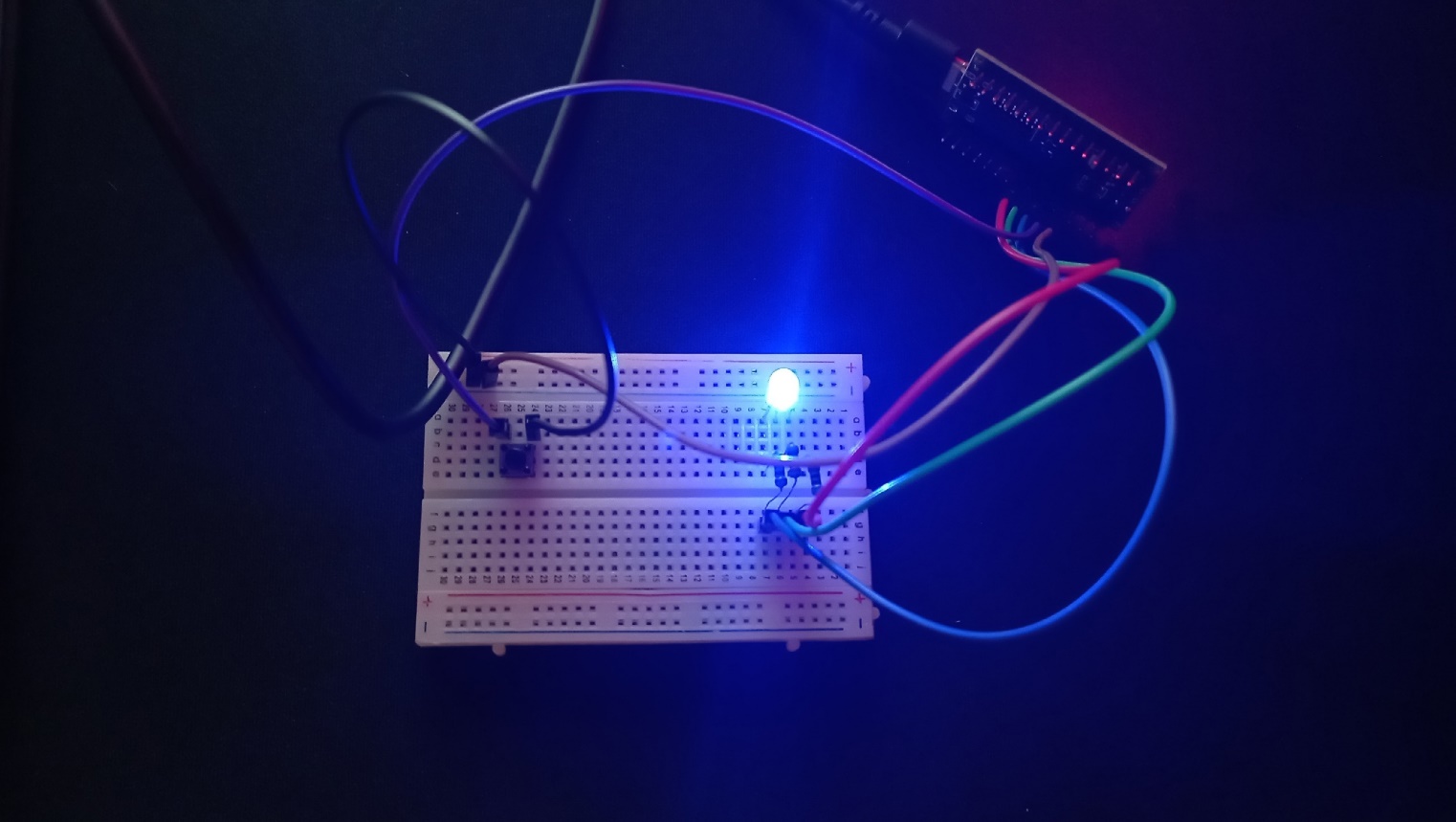
Рекомендации:

- Используйте последовательный интерфейс для отчетов о работе конечных автоматов

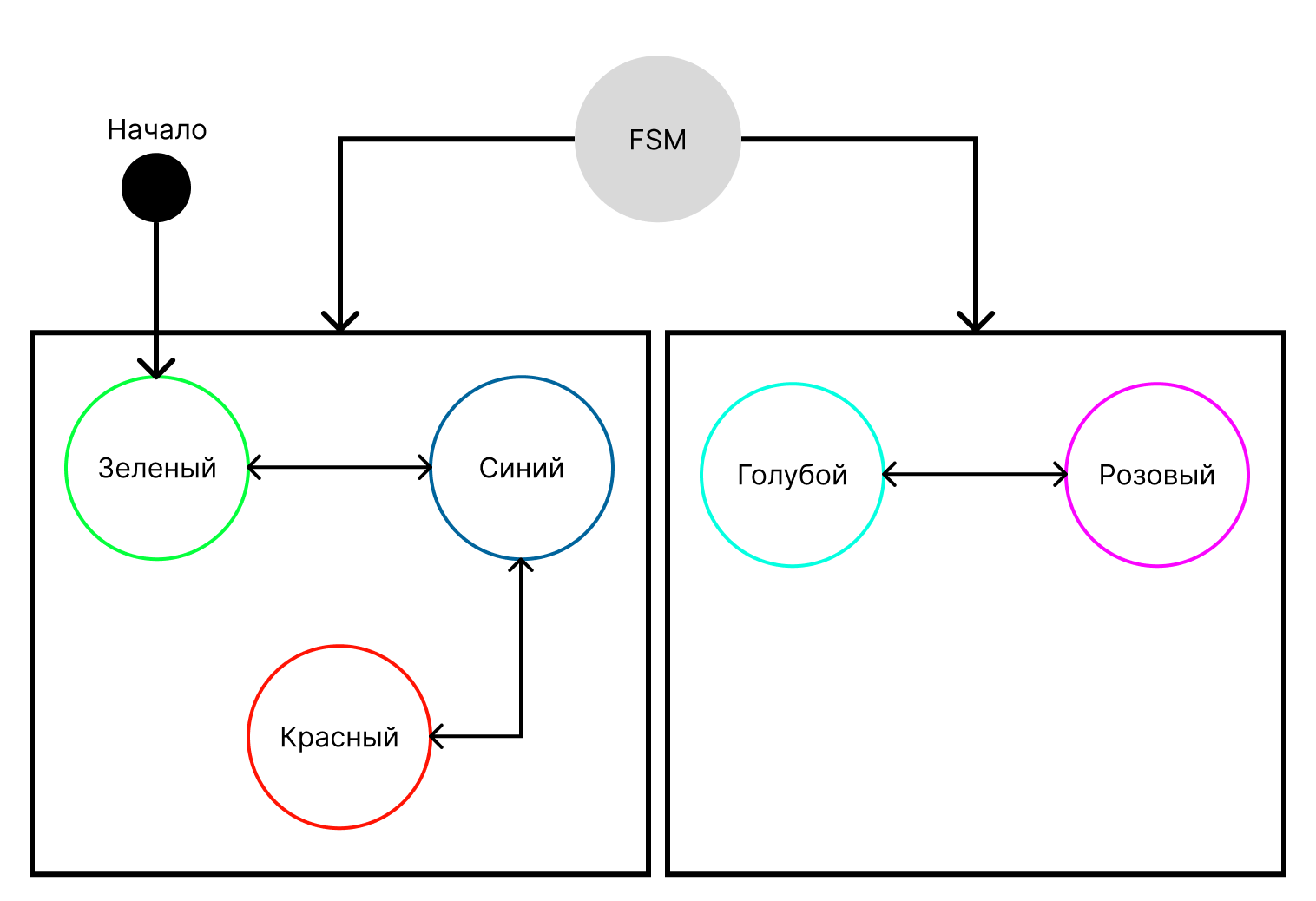
- Максимально используйте решенные, представленные в предыдущих лабораториях.

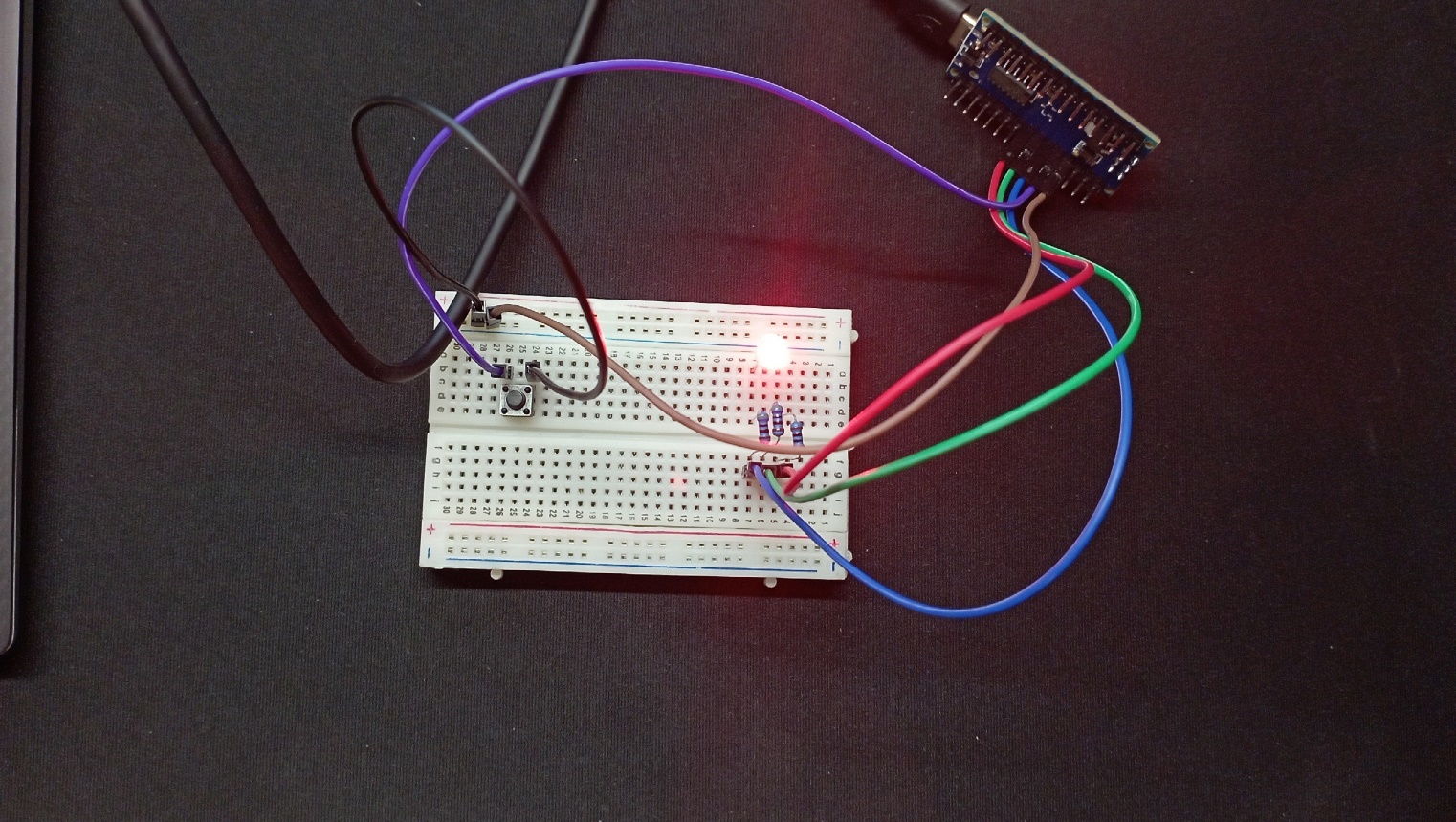
- Посмотрите ресурсы, преподаваемые на курсе

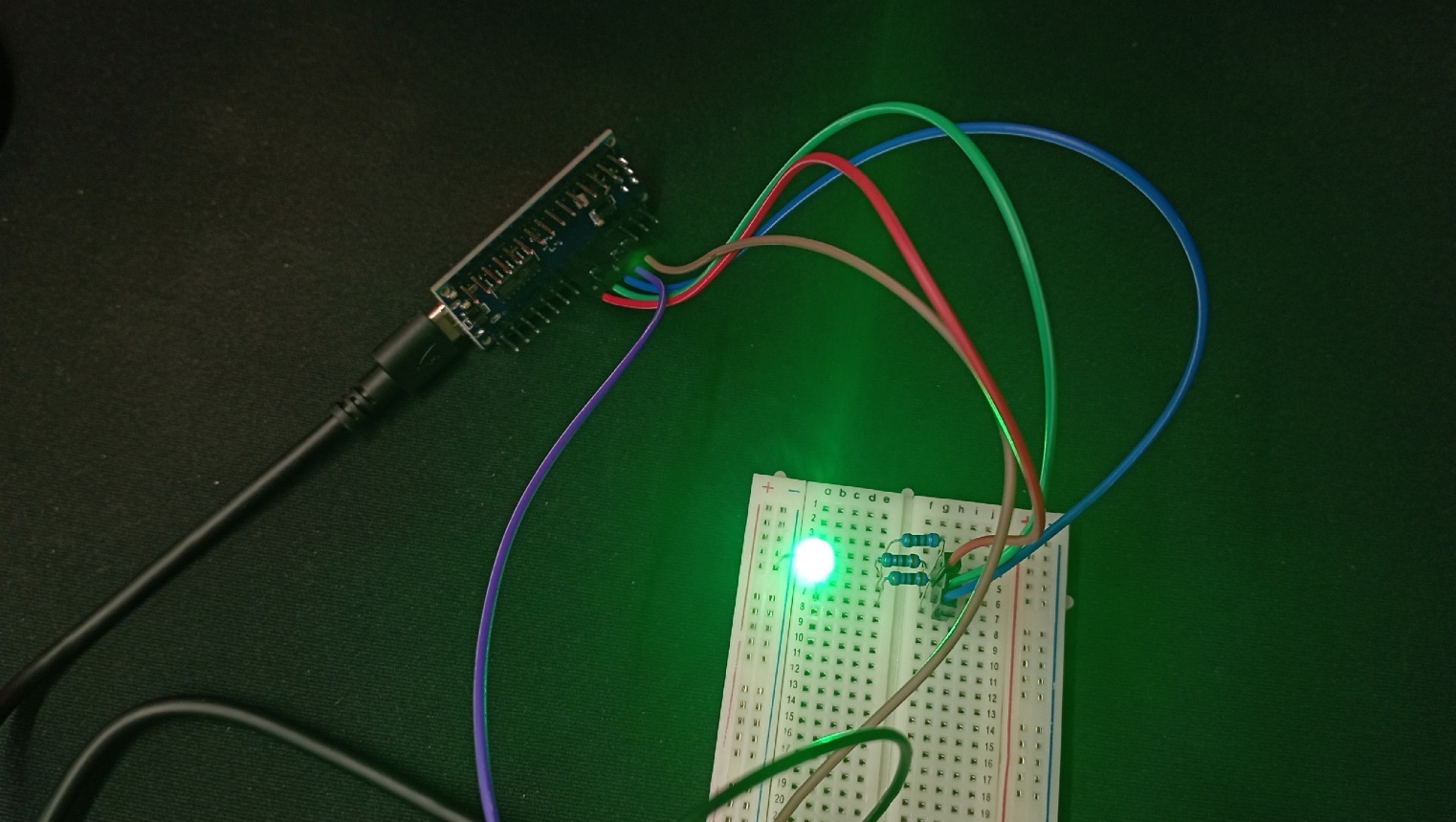
**Пункт A**



**Пункт B**







**Вывод**

В данной лабораторной работе я познакомился с работой конечных автоматов, а также опробовал как их можно использовать, создав эмуляцию светофора что помогло лучше понять работу механизмов реальных светофоров.

**Приложение**

**Пункт А**

#define led 3

#define btn 2

#define led\_off\_state 0

#define led\_on\_state 1

struct State {

 unsigned long StateID;

 unsigned long Time;

 unsigned long Next[2];

};

typedef const struct State Styp;

Styp FSM[2] = {

 {0, 10, {led\_off\_state, led\_on\_state}},

 {1, 10, {led\_on\_state, led\_off\_state}}

};

int FSM\_State = led\_off\_state;

void setup() {

 pinMode(led, OUTPUT);

 pinMode(btn, INPUT\_PULLUP);

 FSM\_State = led\_off\_state;

}

void loop() {

 int state = FSM[FSM\_State].StateID;

 digitalWrite(led, state);

 delay(FSM[FSM\_State].Time \* 10);

 int input = digitalRead(btn);

 while(digitalRead(btn)){}

 FSM\_State = FSM[FSM\_State].Next[input];

}

**Пункт B**

#include "Structures.h"

#define bLed 3

#define gLed 4

#define rLed 5

#define btn 2

Green g;

Blue b;

Red r;

LedState Normal[4] = {

  {0, 500, { r.OFF, g.ON,  b.OFF  }, 1 },

  {1, 300, { r.OFF, g.OFF, b.ON   }, 2 },

  {2, 500, { r.ON,  g.OFF, b.OFF  }, 3 },

  {3, 300, { r.OFF, g.OFF, b.ON   }, 0 }

};

LedState ForHumans[2] = {

  {0, 50, { r.OFF,  g.ON,   b.ON }, 1 },

  {1, 50, { r.ON,   g.OFF,  b.ON }, 0 },

};

Semafore FSM[2] = {

  {0, Normal, 1},

  {1, ForHumans, 0}

};

unsigned long timing;

int typeState = 0;

int fsmState = 0;

void setup() {

  Serial.begin(9600);

  pinMode(rLed, OUTPUT);

  pinMode(gLed, OUTPUT);

  pinMode(bLed, OUTPUT);

  pinMode(btn, INPUT\_PULLUP);

}

void loop() {

 smType(FSM[fsmState].type);

if(!digitalRead(btn))

  if((millis() - timing > 300)){

    timing = millis();

    fsmState = FSM[fsmState].nextState;

  }

}

void smType(LedState type[]) {

  rgb(  type[typeState].Color[0],

        type[typeState].Color[1],

        type[typeState].Color[2]

      );

  delay(type[typeState].Time \* 10);

  typeState = type[typeState].nextState;

}

void rgb(int r, int g, int b) {

  digitalWrite(rLed, r);

  digitalWrite(gLed, g);

  digitalWrite(bLed, b);

}

**Structures.h**

struct Green {

  int ON = 1;

  int OFF = 0;

};

struct Blue {

  int ON = 1;

  int OFF = 0;

};

struct Red {

  int ON = 1;

  int OFF = 0;

};

struct State {

  unsigned long State;

  unsigned long Time;

  unsigned long Color[3];

  unsigned long nextState;

};

typedef const struct State LedState;

struct SemaforState {

  unsigned long State;

  LedState\* type;

  unsigned long nextState;

};

typedef const struct SemaforState Semafore;