|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт искусственного интеллекта

Кафедра проблем управления

**ПРАКТИЧЕСКАЯ РАБОТА №1**

по курсу элитной подготовки **Предиктивные технологии в умном производстве**

**Тема практической работы: «**Светофор для автомобилей с пешеходной фазой по кнопке»

|  |  |
| --- | --- |
| **Студент группы:** КРБО-03-23 | Грачев А.В. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Преподаватель:** | К.т.н. доцент Благовещенский В.Г. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |  |
| --- | --- |
| Работа представлена к защите: | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

Москва 2025

# **1. Цель работы**

Освоить работу с цифровыми входами/выходами и таймингами на STM32 в среде Arduino/STM32Duino, организовать базовый конечный автомат светофора, реализовать запрос пешеходного перехода по кнопке, задать длительность фазы потенциометром и управлять зуммером/индикацией.

# **2. Задача работы**

1. Собрать схему подключения светофора в Wokwi согласно заданию

2. Реализовать циклическую работу автомобильного светофора (зелёный → жёлтый → красный)

3. Организовать обработку запроса пешеходного перехода по кнопке

4. Реализовать пешеходную фазу с длительностью, задаваемой потенциометром

5. Добавить звуковую индикацию и мигание зелёного сигнала в конце пешеходной фазы

6. В качестве дополнительного задания реализовать светофор для велосипедистов и шлагбаумы

# **3. Теоретические сведения**

Светофор для автомобилей с пешеходной фазой представляет собой систему управления дорожным движением, реализованную на микроконтроллере STM32 Nucleo-C031C6. Система использует конечный автомат для управления состояниями светофоров.

Основные компоненты системы:

- Три светодиода автомобильного светофора (красный, жёлтый, зелёный)

- Два светодиода пешеходного светофора (красный, зелёный)

- Два светодиода светофора для велосипедистов (красный, зелёный)

- Кнопка для запроса пешеходного перехода

- Потенциометр для регулировки длительности пешеходной фазы

- Два сервопривода для управления шлагбаумами

Принцип работы:

Автомобильный светофор работает в циклическом режиме: зелёный → жёлтый → красный. При нажатии кнопки формируется заявка на пешеходный переход. После безопасного завершения автомобильной фазы (после жёлтого сигнала) автомобилям включается красный, а пешеходам - зелёный. Длительность пешеходной фазы регулируется потенциометром.

Особенности реализации:

- Антидребезг кнопки для надежного считывания нажатий

- Мигание зелёного сигнала пешеходам в последние 2 секунды фазы

- Плавное возвращение к автомобильному режиму через жёлтый сигнал

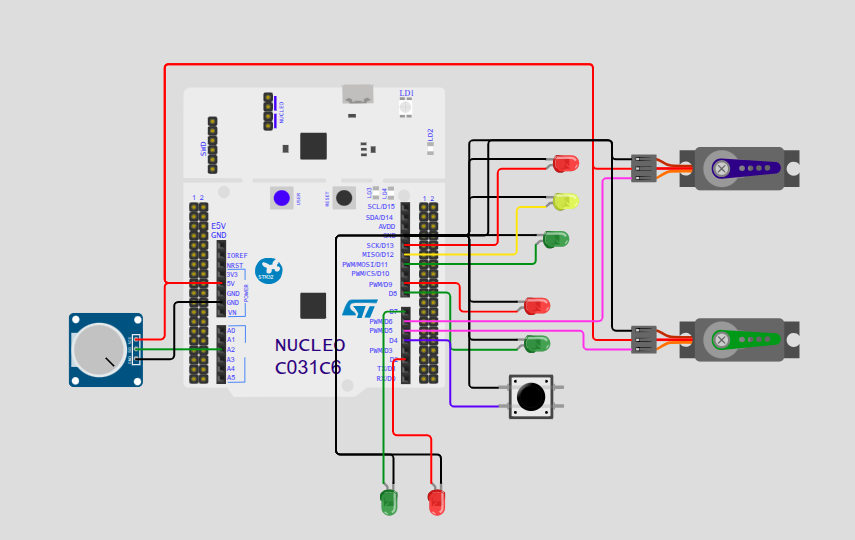
- Дополнительная фаза для велосипедистов с управлением шлагбаумами

Лабораторная установка: среда разработки Wokwi, плата STM32 Nucleo-C031C6, светодиоды, кнопка, потенциометр, сервоприводы.

# **4. Расчетно-графическая часть**

4.1. Схема подключения компонентов

Схема подключения всех компонентов к плате STM32 Nucleo-C031C6 представлена на Рисунке 1.



*Рисунок 1 – Схема подключения компонентов*

Описание подключения компонентов:

Автомобильный светофор:

- Красный: пин D13

- Жёлтый: пин D12

- Зелёный: пин D11

Пешеходный светофор:

- Красный: пин D9

- Зелёный: пин D8

Светофор для велосипедистов:

- Красный: пин D2

- Зелёный: пин D7

Управление:

- Кнопка: пин D4 (INPUT\_PULLUP)

- Потенциометр: пин A2

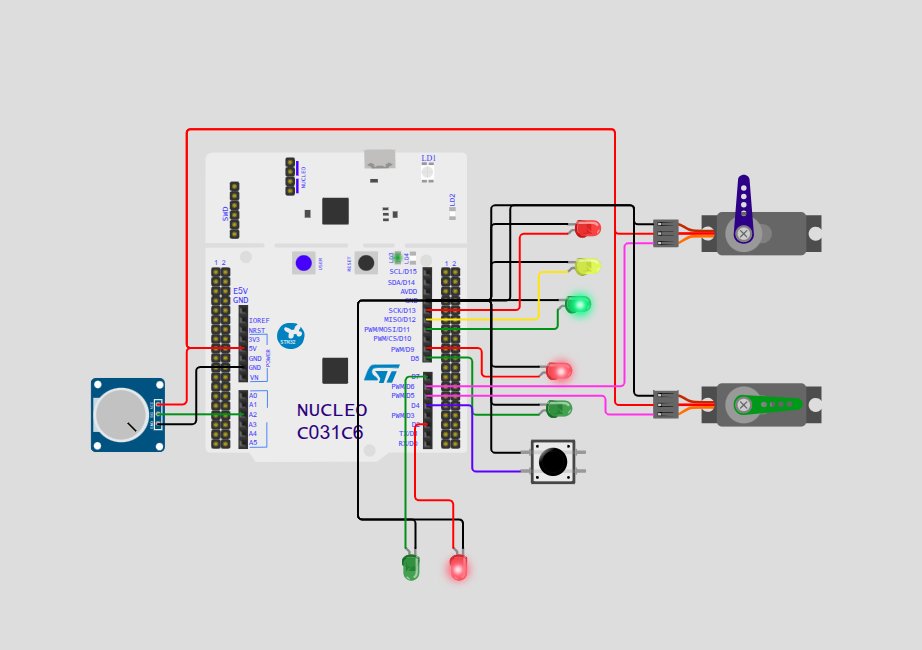
- Сервопривод автомобильного шлагбаума: пин D6

- Сервопривод пешеходного шлагбаума: пин D5

4.2. Алгоритм работы системы

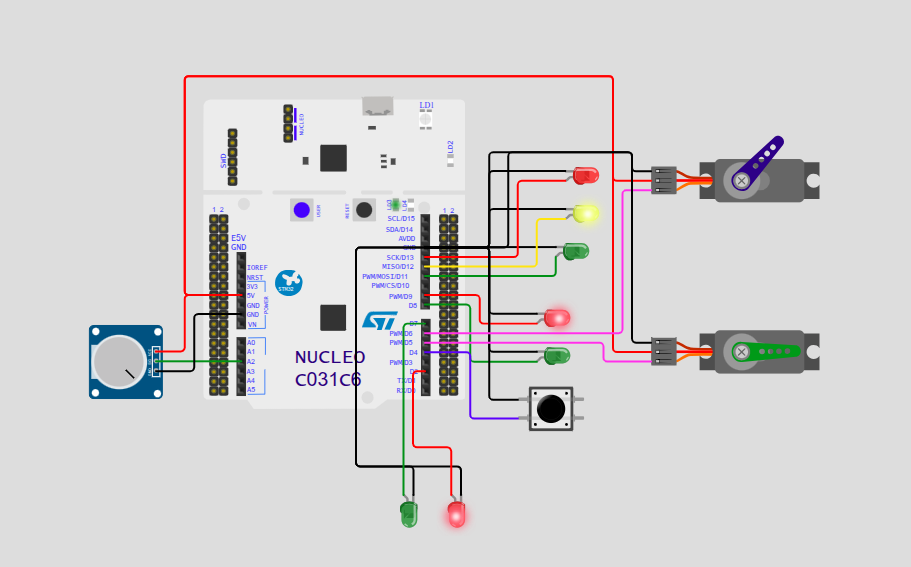
Основной алгоритм реализован как конечный автомат со следующими состояниями:

1. Автомобильный режим (зелёный): Машинам горит зелёный, пешеходам и велосипедистам - красный. Шлагбаум для машин поднят. Длительность: 5 секунд с опросом кнопки.



*Рисунок 2 – Автомобильный режим (зелёный)*

2. Подготовка к переходу (жёлтый): Машинам включается жёлтый сигнал. Шлагбаум для машин опускается под 45°. Длительность: 1 секунда.



*Рисунок 3 – Подготовка к переходу (жёлтый)*

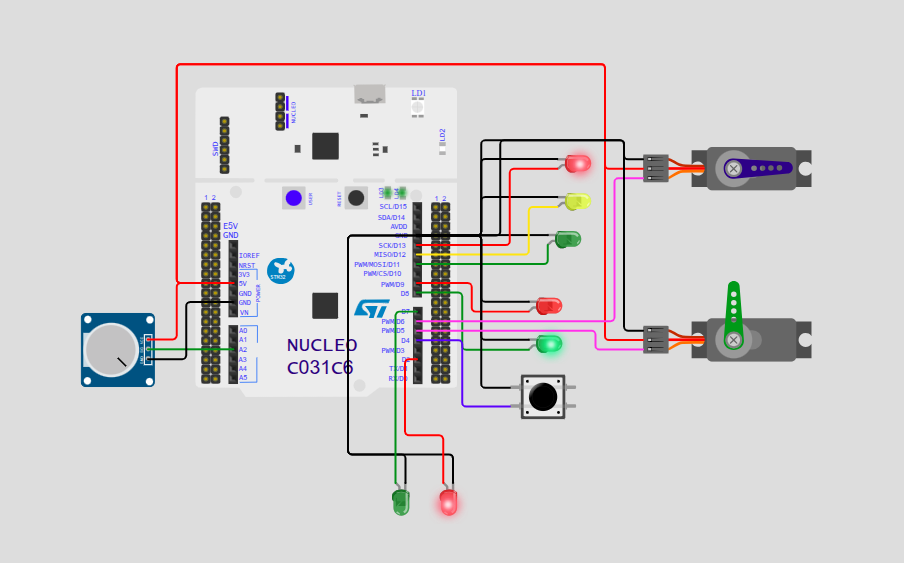
3. Пешеходная фаза: Если была нажата кнопка, выполняется пешеходная фаза:

- Машинам: красный (шлагбаум опущен)

- Пешеходам: зелёный (шлагбаум поднят)

- Длительность: 3-10 секунд (регулируется потенциометром)

- В последние 2 секунды зелёный мигает



*Рисунок 5 – Пешеходная фаза (зелёный для пешеходов)*

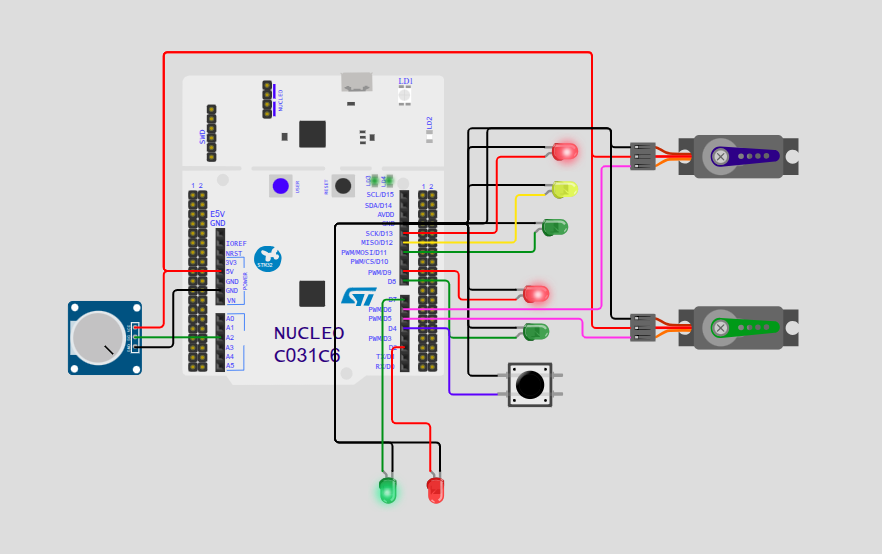
4. Фаза для велосипедистов: После пешеходной фазы выполняется фаза для велосипедистов:

- Велосипедистам: зелёный

- Все шлагбаумы опущены для безопасности

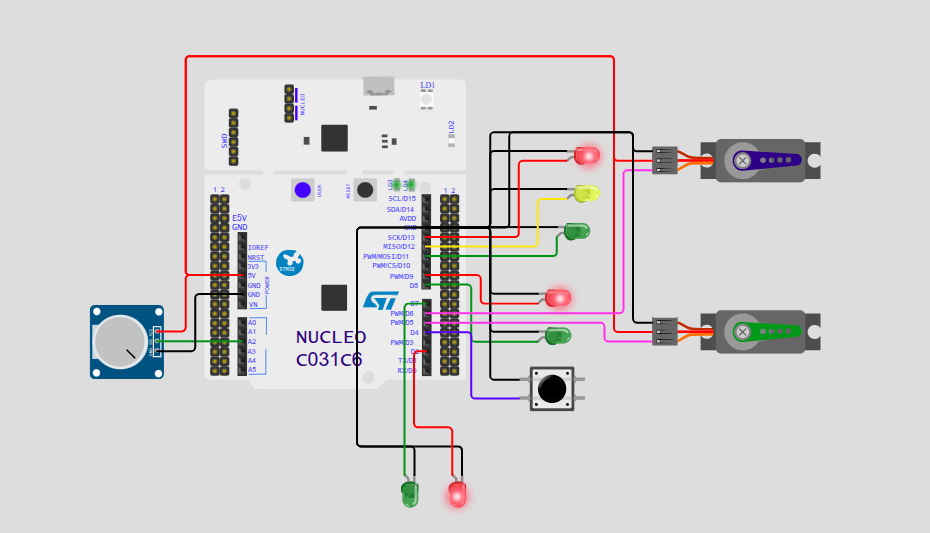
- Длительность: 3-10 секунд

- В последние 2 секунды зелёный мигает



*Рисунок 6 – Фаза для велосипедистов*

5. Возврат к автомобильному режиму: Короткий жёлтый сигнал (0.8 секунды) и переход к состоянию 1.



*Рисунок 4 – Пешеходная фаза (красный для машин)*

4.3. Особенности программной реализации

В коде реализованы следующие ключевые функции:

- cars(bool r, bool y, bool g) - управление автомобильным светофором и шлагбаумом

- peds(bool r, bool g) - управление пешеходным светофором и шлагбаумом

- vels(bool r, bool g) - управление светофором для велосипедистов

- waitWithButton(uint32\_t ms) - ожидание с опросом кнопки и антидребезгом

- pedestrianPhase(uint32\_t cross\_ms) - выполнение пешеходной фазы с миганием

- velsPhase(uint32\_t cross\_ms) - выполнение фазы для велосипедистов

Длительность пешеходной фазы рассчитывается по формуле:

`cross\_ms = map(analogRead(POT), 0, 1023, 3000, 10000)`

что соответствует диапазону от 3 до 10 секунд.

4.4. Анализ работы системы

Система успешно реализует все поставленные задачи:

- Циклическая работа автомобильного светофора работает стабильно

- Обработка кнопки с антидребезгом обеспечивает надежное считывание запросов

- Регулировка длительности пешеходной фазы потенциометром функционирует корректно

- Мигание зелёного сигнала предупреждает об окончании фазы

- Дополнительная функциональность (велосипедисты, шлагбаумы) расширяет возможности системы

Система демонстрирует эффективное использование конечного автомата для управления сложным устройством с несколькими взаимодействующими компонентами.

# **5. Выводы по работе**

В ходе практической работы была успешно реализована система светофора для автомобилей с пешеходной фазой и дополнительной функциональностью для велосипедистов.

Основные достижения:

1. Освоена работа с цифровыми входами/выходами микроконтроллера STM32

2. Реализован конечный автомат для управления состояниями светофора

3. Организована обработка запросов пешеходного перехода с антидребезгом кнопки

4. Реализована регулировка длительности фазы с помощью потенциометра

5. Добавлена звуковая и световая индикация окончания фаз

6. В качестве дополнительного задания успешно реализованы светофор для велосипедистов и система шлагбаумов

Система работает стабильно, все компоненты взаимодействуют корректно. Цель работы достигнута в полном объеме.

**ПРИЛОЖЕНИЕ 1**  
Код-описание диаграммы подключений.  
{

"version": 1,

"author": "Uri Shaked",

"editor": "wokwi",

"parts": [

{

"type": "board-st-nucleo-c031c6",

"id": "nucleo",

"top": -56.77,

"left": -260.18,

"attrs": {}

},

{

"type": "wokwi-led",

"id": "led1",

"top": -1.2,

"left": 102.6,

"rotate": 90,

"attrs": { "color": "red" }

},

{

"type": "wokwi-led",

"id": "led2",

"top": 37.2,

"left": 102.6,

"rotate": 90,

"attrs": { "color": "yellow" }

},

{

"type": "wokwi-led",

"id": "led3",

"top": 75.6,

"left": 93,

"rotate": 90,

"attrs": { "color": "green" }

},

{

"type": "wokwi-led",

"id": "led4",

"top": 142.8,

"left": 73.8,

"rotate": 90,

"attrs": { "color": "red" }

},

{

"type": "wokwi-led",

"id": "led5",

"top": 181.2,

"left": 73.8,

"rotate": 90,

"attrs": { "color": "green" }

},

{

"type": "wokwi-pushbutton",

"id": "btn1",

"top": 236.6,

"left": 57.6,

"attrs": { "color": "black", "xray": "1" }

},

{

"type": "wokwi-potentiometer",

"id": "pot1",

"top": 173.8,

"left": -375.7,

"rotate": 270,

"attrs": {}

},

{

"type": "wokwi-servo",

"id": "servo1",

"top": 142,

"left": 192,

"attrs": { "hornColor": "#008800" }

},

{

"type": "wokwi-servo",

"id": "servo2",

"top": -30.8,

"left": 192,

"attrs": { "hornColor": "#000088" }

},

{

"type": "wokwi-led",

"id": "led6",

"top": 337.6,

"left": -72.6,

"rotate": 180,

"attrs": { "color": "green" }

},

{

"type": "wokwi-led",

"id": "led7",

"top": 337.6,

"left": -24.6,

"rotate": 180,

"attrs": { "color": "red" }

}

],

"connections": [

[ "$serialMonitor:TX", "nucleo:PA3", "", [] ],

[ "$serialMonitor:RX", "nucleo:PA2", "", [] ],

[ "pot1:SIG", "nucleo:A2", "green", [ "h0" ] ],

[ "pot1:GND", "nucleo:GND.6", "black", [ "h38.4", "v-57.6" ] ],

[ "pot1:VCC", "nucleo:5V.2", "red", [ "h28.8", "v-56.8" ] ],

[ "led3:A", "nucleo:D11", "green", [ "v19.2", "h-132.05" ] ],

[ "led1:C", "nucleo:GND.9", "black", [ "h-76.8", "v77.2" ] ],

[ "led2:C", "nucleo:GND.9", "black", [ "h-76.8", "v0.4" ] ],

[ "led3:C", "nucleo:GND.9", "black", [ "h0" ] ],

[ "led2:A", "nucleo:D12", "gold", [ "h-28.8", "v48" ] ],

[ "led1:A", "nucleo:D13", "red", [ "h-48", "v76.8" ] ],

[ "led4:C", "nucleo:GND.9", "black", [ "h-48", "v-66.8" ] ],

[ "led5:C", "nucleo:GND.9", "black", [ "h-48", "v-95.6" ] ],

[ "led4:A", "nucleo:D9", "red", [ "h-57.6", "v-28.8" ] ],

[ "led5:A", "nucleo:D8", "green", [ "h-67.2", "v-48" ] ],

[ "btn1:1.l", "nucleo:GND.9", "black", [ "h-28.8", "v-153.6" ] ],

[ "btn1:2.l", "nucleo:D4", "blue", [ "h-48", "v-67" ] ],

[ "servo2:V+", "nucleo:5V.2", "red", [ "h-38.4", "v-105.5", "h-432", "v220.8" ] ],

[ "servo1:V+", "nucleo:5V.2", "red", [ "h-38.4", "v-278.3", "h-432", "v220.8" ] ],

[ "servo2:PWM", "nucleo:D6", "violet", [ "h-28.8", "v144.2" ] ],

[ "servo2:GND", "nucleo:GND.9", "black", [ "h-19.2", "v-19.2", "h-124.8", "v96" ] ],

[ "servo1:PWM", "nucleo:D5", "violet", [ "h-48", "v-19" ] ],

[ "servo1:GND", "nucleo:GND.9", "black", [ "h-19.2", "v-192", "h-144", "v96" ] ],

[ "nucleo:GND.9", "led6:C", "black", [ "h-69.55", "v220.8", "h57.6" ] ],

[ "led7:C", "nucleo:GND.9", "black", [ "v-28.8", "h-106", "v-220.8" ] ],

[ "led6:A", "nucleo:D7", "green", [ "v0" ] ],

[ "led7:A", "nucleo:D2", "red", [ "v-48", "h-38.4", "v-76.8" ] ]

],

"dependencies": {}

}

**ПРИЛОЖЕНИЕ 2**

Код основной программы.

// NUCLEO-C031C6: светофор + пешеходный по кнопке

// Пины из задания:

#define LED\_R 13 // машины: красный

#define LED\_Y 12 // машины: жёлтый

#define LED\_G 11 // машины: зелёный

#define LED\_R\_P 9 // пешеходы: красный

#define LED\_G\_P 8 // пешеходы: зелёный

#define LED\_R\_V 2 // пешеходы: красный

#define LED\_G\_V 7 // пешеходы: зелёный

#define BUT 4 // кнопка (INPUT\_PULLUP)

#define POT A2 // потенциометр (длительность пешеходной фазы)

#define SERVO\_CAR 6

#define SERVO\_PED 5

const int minPulseWidth = 500; // Minimum pulse width (in microseconds)

const int maxPulseWidth = 2400; // Maximum pulse width (in microseconds)

// --- сервисные функции --

void cars(bool r, bool y, bool g) {

digitalWrite(LED\_R, r);

digitalWrite(LED\_Y, y);

digitalWrite(LED\_G, g);

if (r) {servoAngle(90, SERVO\_CAR);}

if (y) {servoAngle(45, SERVO\_CAR);}

if (g) {servoAngle(0, SERVO\_CAR);}

}

void vels(bool r, bool g) {

digitalWrite(LED\_R\_V, r);

digitalWrite(LED\_G\_V, g);

if (g) {servoAngle(90, SERVO\_PED);}

if (g) {servoAngle(90, SERVO\_CAR);}

}

void peds(bool r, bool g) {

digitalWrite(LED\_R\_P, r);

digitalWrite(LED\_G\_P, g);

if (r) {servoAngle(90, SERVO\_PED);}

if (g) {servoAngle(0, SERVO\_PED);}

}

void servoAngle(int angle, int pin) {

// Calculate the pulse width for the given angle

int pulseWidth = map(angle, 0, 180, minPulseWidth, maxPulseWidth);

//map() is a built-in function in Arduino environment.

//It maps a number from one range to another.

// Generate the PWM signal (1 ms to 2.4 ms pulse width)

digitalWrite(pin, HIGH);

delayMicroseconds(pulseWidth);

digitalWrite(pin, LOW);

delay(20 - pulseWidth / 1000); // Rest of the 20ms period

}

// ожидание с опросом кнопки (для дебаунса и «заказа» перехода)

bool requestCross = false;

void waitWithButton(uint32\_t ms) {

const uint16\_t step = 20; // мс

uint32\_t t0 = millis();

while (millis() - t0 < ms) {

if (digitalRead(BUT) == LOW) { // нажали (кнопка к GND)

delay(20); // дебаунс

if (digitalRead(BUT) == LOW) {

requestCross = true; // запоминаем запрос

while (digitalRead(BUT) == LOW) {} // ждём отпускание

}

}

delay(step);

}

}

// мигание зелёного пешеходам в последние N миллисекунд

void pedestrianPhase(uint32\_t cross\_ms) {

// машины стоят, пешеходы идут

cars(true, false, false);

peds(false, true);

vels(true, false);

uint32\_t t0 = millis();

while (millis() - t0 < cross\_ms) {

uint32\_t elapsed = millis() - t0;

uint32\_t remain = (cross\_ms > elapsed) ? (cross\_ms - elapsed) : 0;

// в последние 2 секунды мигаем зелёным для пешеходов

if (remain <= 2000) {

digitalWrite(LED\_G\_P, (millis() / 200) % 2); // 5 Гц/2=~2.5 Гц

} else {

digitalWrite(LED\_G\_P, HIGH);

}

delay(20);

}

// пешеходам - красный, готовим возврат к движению машин peds(true, false);

peds(true, false);

}

void velsPhase(uint32\_t cross\_ms) {

cars(true, false, false);

peds(true, false);

vels(false, true);

uint32\_t t0 = millis();

while (millis() - t0 < cross\_ms) {

uint32\_t elapsed = millis() - t0;

uint32\_t remain = (cross\_ms > elapsed) ? (cross\_ms - elapsed) : 0;

// в последние 2 секунды мигаем зелёным для великов

if (remain <= 2000) {

digitalWrite(LED\_G\_V, (millis() / 200) % 2); // 5 Гц/2=~2.5 Гц

} else {

digitalWrite(LED\_G\_V, HIGH);

}

delay(20);

}

vels(true, false);

}

// -- настройка -

void setup() {

pinMode(LED\_R, OUTPUT);

pinMode(LED\_Y, OUTPUT);

pinMode(LED\_G, OUTPUT);

pinMode(LED\_R\_P, OUTPUT);

pinMode(LED\_G\_P, OUTPUT);

pinMode(LED\_R\_V, OUTPUT);

pinMode(LED\_G\_V, OUTPUT);

pinMode(SERVO\_CAR, OUTPUT);

pinMode(SERVO\_PED, OUTPUT);

pinMode(BUT, INPUT\_PULLUP);

pinMode(POT, INPUT);

// стартовое состояние: машинам зелёный, пешеходам красный

cars(false, false, true);

peds(true, false);

}

// -- основной цикл -

void loop() {

// Длительность пешеходной фазы с потенциометра: 3…10 с

uint32\_t cross\_ms = map(analogRead(POT), 0, 1023, 3000, 10000);

// 1) Машинам зелёный

cars(false, false, true);

peds(true, false);

vels(true, false);

waitWithButton(5000); // «обычный» проезд ~5 с

// 2) Жёлтый для машин

cars(false, true, false);

delay(1000);

// 3) Если просили переход - включаем пешеходную фазу

cars(true, false, false); // красный для машин

delay(300); // небольшая пауза безопасности

if (requestCross) {

pedestrianPhase(cross\_ms);

velsPhase(cross\_ms);

requestCross = false;

} else {

// без пешеходов подержим красный 2 с

delay(2000);

}

peds(true, false);

vels(true, false);

// 4) Короткий жёлтый и снова зелёный для машин

cars(false, true, false);

delay(800);

cars(false, false, true);

// цикл повторяется

}