

Практическое занятие №

Тема: «Программирование Шаговых двигателей»

Цель работы: приобрести практические навыки по программированию шаговых двигателей эффектов на платформе Arduino.

Последовательность выполнения работы:

- Собрать схемы на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно заданию в примере.
- Выполнить задание для самостоятельной работы.

Содержание отчета:

- Название практического занятия, его цель.
- Фото или скриншоты собранной схемы.
- Написанный программный код вставить текстом, Courier New, 12 кегль, одинарный отступ без абзацев.
- Вывод о проделанной работе.
- Файл Fritzing с принципиальной и монтажной схемой.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Шаговый двигатель (stepper motor) предназначен для точного позиционирования или перемещения объекта на заданное количество шагов вала. Плата Arduino может управлять шаговым двигателем с помощью драйвера и библиотеки stepper.h или accelstepper.h. Рассмотрим принцип работы и схему подключения шагового двигателя к Arduino Uno / Nano, а также разберем скетч для управления шаговым мотором.

Принцип работы шагового двигателя

В зависимости от конструкции, сегодня применяются три вида шаговых двигателей: с постоянным магнитом, с переменным магнитным сопротивлением и гибридные двигатели. У двигателей с постоянным магнитом число шагов на один оборот вала доходит до 48, то есть один шаг соответствует повороту вала на $7,5^\circ$. Гибридные двигатели обеспечивают не меньше 400 шагов на один оборот (угол шага $0,9^\circ$).

Устройство шагового мотора

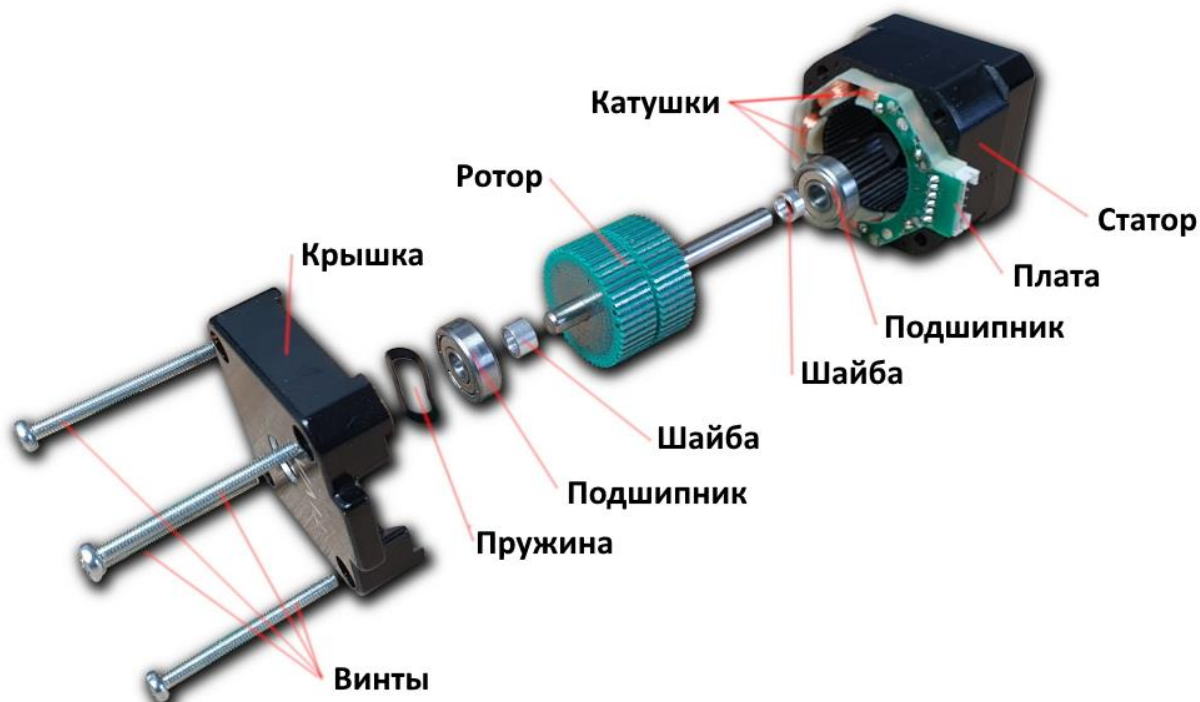


Рисунок 1 – Устройство шагового мотора в разрезе

Подсчитав количество сделанных шагов, можно определить точный угол поворота ротора. Таким образом, шаговый двигатель является сегодня идеальным приводом в 3D принтерах, станках с ЧПУ и в другом промышленном оборудовании. Это лишь краткий обзор устройства и принципа работы stepper motor, нас больше интересует, как осуществляется управление шаговым двигателем с помощью Ардуино.

Драйвер шагового двигателя

Шаговый двигатель — это бесколлекторный синхронный двигатель, как и все двигатели, он преобразует электрическую энергию в механическую. В отличие от двигателя постоянного тока в которых происходит вращение вала, вал шаговых двигателей совершает дискретные перемещения, то есть вращается не постоянно, а шагами. Каждый шаг вала (ротора) представляет собой часть полного оборота.

Драйвера для управления шаговым двигателем

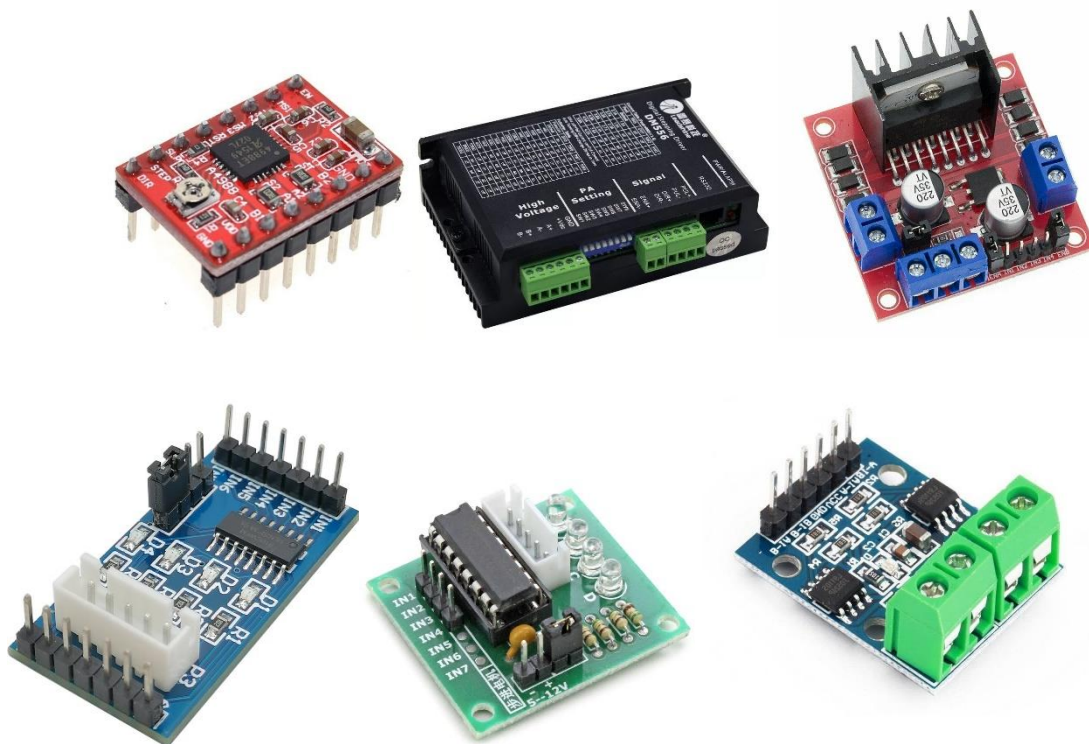


Рисунок 2 – Виды драйверов для управления шаговым двигателем

Вращение вала двигателя осуществляется с помощью сигнала, который управляет магнитным полем катушек в статоре драйвера. Сигнал генерирует драйвер шагового двигателя. Магнитное поле, возникающее при прохождении электрического тока в обмотках статора, заставляет вращаться вал, на котором установлены магниты. Количество шагов задаются в программе с помощью библиотеки Arduino IDE.

Схема подключения шагового двигателя 28BYJ-48 к Arduino Uno через драйвер ULN2003 изображена на рисунке ниже. Основные характеристики мотора 28BYJ-48: питание от 5 или 12 Вольт, 4-х фазный двигатель, угол шага $5,625^\circ$. Порты драйвера IN1 — IN4 подключаются к любым цифровым выводам платы Arduino Mega или Nano. Светодиоды на модуле служат для индикации включения катушек двигателя.

Как подключить шаговый двигатель к Ардуино

Для этого потребуется:

- Arduino Uno / Arduino Nano / Arduino Mega;
- драйвер шагового двигателя ULN2003;
- шаговый двигатель 28BYJ-48;
- провода «папа-мама».

Схема подключения шагового двигателя к Arduino UNO:

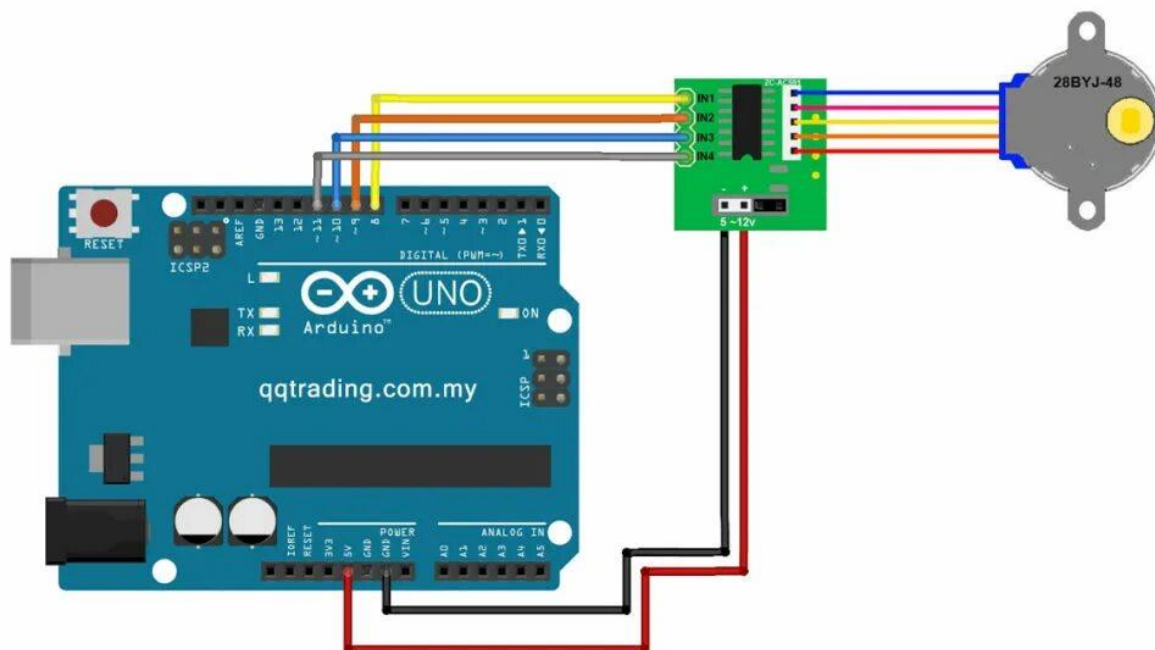


Рисунок 3 – Схема подключения шагового двигателя к Arduino UNO

Управление шаговым двигателем через Ардуино производится путем подачи импульсов на обмотки мотора в определенной последовательности. Для облегчения управления шаговым мотором созданы специальные библиотеки `stepper.h` и `accelstepper.h`, но можно вращать вал мотора без стандартных библиотек. Подключите шаговый мотор к микроконтроллеру, как на схеме выше и загрузите следующий скетч.

Скетч для управления шаговым двигателем:

```
// порты для подключения модуля ULN2003 к Arduino
#define in1 8
#define in2 9
#define in3 10
#define in4 11

int d1 = 5; // время задержки между импульсами

void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {
  digitalWrite(in1, HIGH);
```

```
digitalWrite(in2, LOW);  
digitalWrite(in3, LOW);  
digitalWrite(in4, HIGH);  
delay(d1);
```

```
digitalWrite(in1, HIGH);  
digitalWrite(in2, HIGH);  
digitalWrite(in3, LOW);  
digitalWrite(in4, LOW);  
delay(d1);
```

```
digitalWrite(in1, LOW);  
digitalWrite(in2, HIGH);  
digitalWrite(in3, HIGH);  
digitalWrite(in4, LOW);  
delay(d1);
```

```
digitalWrite(in1, LOW);  
digitalWrite(in2, LOW);  
digitalWrite(in3, HIGH);  
digitalWrite(in4, HIGH);  
delay(d1);
```

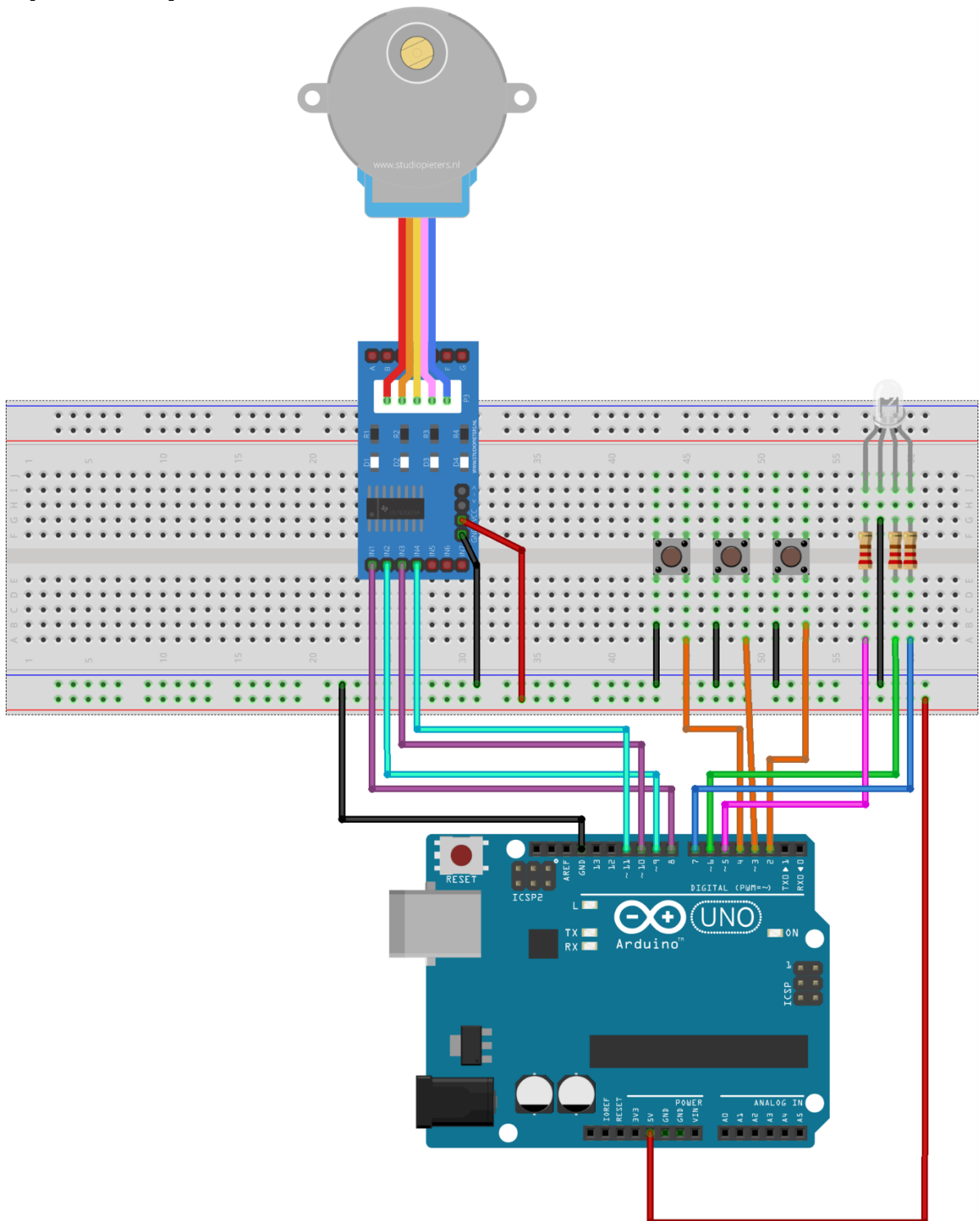
```
}
```

ЗАДАНИЕ

Пояснения к программе:

Вместо портов 8,9,10,11 можно использовать любые цифровые порты; время задержки в миллисекундах $\text{int dl} = 5$; можно изменять, чем меньше задержка в программе, тем быстрее будет вращаться вал мотора; алгоритм работы программы, представлен на следующей картинке.

Собрать схему:



Пояснения к программе:

Логика работы лифта:

- лифт может перемещаться между этажами от -3 до 20
- кнопки "Вверх" и "Вниз" выбирают целевой этаж
- кнопка "Пуск" запускает движение к выбранному этажу
- при достижении предела этажей загорается красный светодиод

Управление двигателем:

- используется полношаговый режим (8 шагов на цикл)
- за один «этаж» двигатель делает 500 полных циклов
- двигатель останавливается после завершения движения

Индикация:

- синий: нажата кнопка выбора этажа
- зеленый: лифт в движении
- красный: достигнут предел этажей
- Моргание зеленым: прибытие на этаж

Написать программу управления:

```
// =====  
// НАСТРОЙКИ ДВИГАТЕЛЯ И ЛИФТА  
// =====  
  
// Пины подключения драйвера ULN2003  
#define IN1 8  
#define IN2 9  
#define IN3 10  
#define IN4 11  
  
// Пины кнопок  
#define BUTTON_UP 2  
#define BUTTON_DOWN 3  
#define BUTTON_START 4  
  
// Пины RGB светодиода  
#define RED_PIN 5  
#define GREEN_PIN 6  
#define BLUE_PIN 7  
  
// Параметры лифта  
#define MIN_FLOOR -3 // Минимальный этаж (подвал)  
#define MAX_FLOOR 20 // Максимальный этаж  
#define FLOOR_HEIGHT 500 // Количество шагов на один этаж
```

```

// Настройки шагового двигателя
#define STEPS_PER_REVOLUTION 2048 // Полных шагов за оборот
для 28BYJ-48

// Задержка между шагами (в микросекундах)
#define STEP_DELAY 3000

// =====
// ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ
// =====

int currentFloor = 0; // Текущий этаж (стартуем с
0)
int targetFloor = 0; // Целевой этаж
bool moving = false; // Флаг движения лифта
bool limitReached = false; // Флаг достижения предела
этажей

// Исправленный массив последовательности шагов для
двигателя (полношаговый режим)
byte stepSequence[8][4] = { // Изменено с [4][4] на [8][4]
    {1, 0, 0, 1},
    {1, 0, 0, 0},
    {1, 1, 0, 0},
    {0, 1, 0, 0},
    {0, 1, 1, 0},
    {0, 0, 1, 0},
    {0, 0, 1, 1},
    {0, 0, 0, 1}
};

// =====
// ФУНКЦИИ УПРАВЛЕНИЯ СВЕТОДИОДОМ
// =====

void setLED(int red, int green, int blue) {
    // Управление RGB светодиодом (общий катод)
    digitalWrite(REDA_PIN, red);
    digitalWrite(GREEN_PIN, green);
    digitalWrite(BLUE_PIN, blue);
}

```

```

void showStatusLED() {
    // Отображение текущего статуса лифта
    if (moving) {
        // Движение - зеленый
        setLED(LOW, HIGH, LOW);
    } else if (limitReached) {
        // Достигнут предел - красный
        setLED(HIGH, LOW, LOW);
    } else if (digitalRead(BUTTON_UP) == LOW ||
digitalRead(BUTTON_DOWN) == LOW) {
        // Кнопка нажата - синий
        setLED(LOW, LOW, HIGH);
    } else {
        // Ожидание - выключен
        setLED(LOW, LOW, LOW);
    }
}

// =====
// ФУНКЦИИ УПРАВЛЕНИЯ ДВИГАТЕЛЕМ
// =====

void setMotorPins(int a, int b, int c, int d) {
    // Установка состояния пинов двигателя
    digitalWrite(IN1, a);
    digitalWrite(IN2, b);
    digitalWrite(IN3, c);
    digitalWrite(IN4, d);
}

void performStep(int step) {
    // Выполнение одного шага
    // Проверка границ массива
    if (step >= 0 && step < 8) {
        setMotorPins(
            stepSequence[step][0],
            stepSequence[step][1],
            stepSequence[step][2],
            stepSequence[step][3]
        );
        delayMicroseconds(STEP_DELAY);
    }
}
}

```

```

void moveOneFloor(bool up) {
    // Перемещение на один этаж вверх или вниз
    int steps = FLOOR_HEIGHT;

    if (up) {
        // Движение вверх
        for (int i = 0; i < steps; i++) {
            for (int step = 0; step < 8; step++) {
                performStep(step);
            }
        }
        currentFloor++;
    } else {
        // Движение вниз
        for (int i = 0; i < steps; i++) {
            for (int step = 7; step >= 0; step--) {
                performStep(step);
            }
        }
        currentFloor--;
    }

    // Вывод информации о текущем этаже
    Serial.print("Текущий этаж: ");
    Serial.println(currentFloor);
}

void moveToTargetFloor() {
    // Перемещение на целевой этаж
    Serial.print("Начинаем движение с этажа ");
    Serial.print(currentFloor);
    Serial.print(" на этаж ");
    Serial.println(targetFloor);

    moving = true;
    limitReached = false;

    // Определяем направление движения
    bool moveUp = (targetFloor > currentFloor);
    int floorsToMove = abs(targetFloor - currentFloor);

    // Выполняем перемещение по этажам
    for (int i = 0; i < floorsToMove; i++) {
        moveOneFloor(moveUp);
    }
}

```

```

    // Проверяем, не достигли ли целевого этажа
    if ((moveUp && currentFloor >= targetFloor) ||
        (!moveUp && currentFloor <= targetFloor)) {
        break;
    }
}

// Достигли целевого этажа
Serial.print("Прибыли на этаж: ");
Serial.println(currentFloor);

// Обновляем целевой этаж
targetFloor = currentFloor;
moving = false;

// Моргаем зеленым светодиодом 3 раза
for (int i = 0; i < 3; i++) {
    setLED(LOW, HIGH, LOW);
    delay(300);
    setLED(LOW, LOW, LOW);
    delay(300);
}
}

// =====
// ФУНКЦИИ ОБРАБОТКИ КНОПОК
// =====

void handleButtons() {
    // Обработка нажатий кнопок
    static bool lastUpState = HIGH;
    static bool lastDownState = HIGH;
    static bool lastStartState = HIGH;

    bool currentUpState = digitalRead(BUTTON_UP);
    bool currentDownState = digitalRead(BUTTON_DOWN);
    bool currentStartState = digitalRead(BUTTON_START);
}

```

```

// Кнопка "Вверх"
if (lastUpState == HIGH && currentUpState == LOW) {
    if (!moving && targetFloor < MAX_FLOOR) {
        targetFloor++;
        limitReached = false;
        Serial.print("Целевой этаж: ");
        Serial.println(targetFloor);
    } else if (targetFloor >= MAX_FLOOR) {
        limitReached = true;
        Serial.println("Достигнут максимальный этаж!");
    }
    delay(50); // Защита отдребезга
}

// Кнопка "Вниз"
if (lastDownState == HIGH && currentDownState == LOW) {
    if (!moving && targetFloor > MIN_FLOOR) {
        targetFloor--;
        limitReached = false;
        Serial.print("Целевой этаж: ");
        Serial.println(targetFloor);
    } else if (targetFloor <= MIN_FLOOR) {
        limitReached = true;
        Serial.println("Достигнут минимальный этаж!");
    }
    delay(50); // Защита отдребезга
}

// Кнопка "Пуск"
if (lastStartState == HIGH && currentStartState == LOW) {
    if (!moving && targetFloor != currentFloor) {
        Serial.println("Запуск лифта...");
        moveToTargetFloor();
    }
    delay(50); // Защита отдребезга
}

// Сохраняем текущие состояния для следующего цикла
lastUpState = currentUpState;
lastDownState = currentDownState;
lastStartState = currentStartState;
}

```

```

// =====
// ОСНОВНЫЕ ФУНКЦИИ ARDUINO
// =====

void setup() {
  // Инициализация последовательного порта
  Serial.begin(9600);
  Serial.println("Система лифта инициализирована");
  Serial.println("=====");

  // Настройка пинов двигателя
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  // Настройка пинов кнопок
  pinMode(BUTTON_UP, INPUT_PULLUP);
  pinMode(BUTTON_DOWN, INPUT_PULLUP);
  pinMode(BUTTON_START, INPUT_PULLUP);

  // Настройка пинов светодиода
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);

  // Выключить двигатель
  setMotorPins(0, 0, 0, 0);

  // Выключить светодиод
  setLED(LOW, LOW, LOW);

  // Вывод начальной информации
  Serial.println("Система готова к работе");
  Serial.print("Диапазон этажей: от ");
  Serial.print(MIN_FLOOR);
  Serial.print(" до ");
  Serial.println(MAX_FLOOR);
  Serial.print("Текущий этаж: ");
  Serial.println(currentFloor);
  Serial.println("Используйте кнопки для выбора этажа");
  Serial.println("=====");
}

```

```
void loop() {  
  // Обработка кнопок  
  handleButtons();  
  
  // Отображение статуса на светодиоде  
  showStatusLED();  
  
  // Небольшая задержка для стабильности  
  delay(10);  
}
```