

Управление силовой нагрузкой

Павел Пронин
C++ разработчик



Проверка связи



Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти



Поставьте в чат:

-  если меня видно и слышно
-  если нет

Павел Пронин

О спикере:

- Разработчик на C++ более 8-ми лет
- Опыт в разработке беспилотных автомобилей
- С 2022 года разработчик в компании разработки мобильных игр Playrix (компания разрабатывает такие игры как homescapes и gardegscapes)



Вспоминаем прошрое занятие

Вопрос: что измеряет акселерометр?



Вспоминаем прошрое занятие

Вопрос: что измеряет акселерометр?

Ответ: акселерометр измеряет ускорение
(проекцию ускорения на измерительную ось)



Вспоминаем прошрое занятие

Вопрос: сколько разрядов выделяется на адрес ведомого устройства в интерфейсе I2C?



Вспоминаем прошрое занятие

Вопрос: сколько разрядов выделяется на адрес ведомого устройства в интерфейсе I2C?

Ответ: на адрес выделяется 7 разрядов



Вспоминаем прошрое занятие

Вопрос: какая должна быть
последовательность действий при чтении
данных из модуля MPU6050?



Вспоминаем прошрое занятие

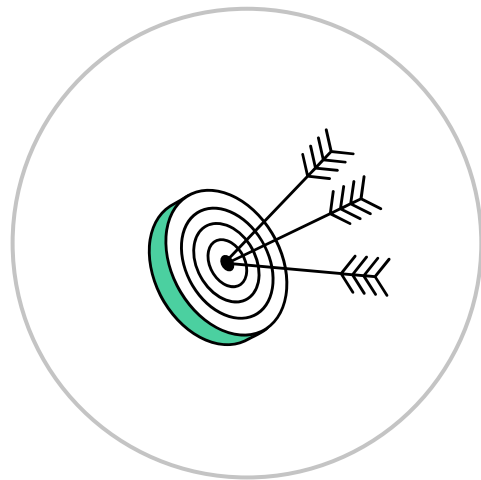
Вопрос: какая должна быть
последовательность действий при чтении
данных из модуля MPU6050?

Ответ: установка адреса на нужный регистр с
данными (т.е. запись), затем чтение (минимум
2-х байт, т.к. данные 16-разрядные)



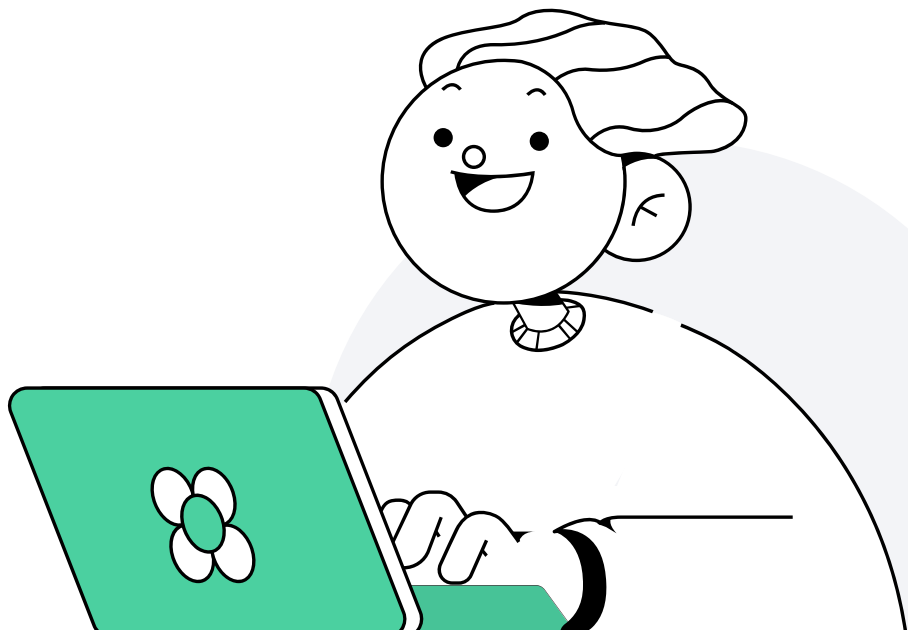
Цели занятия

- Узнаем, как подключать электромагнитное реле и управлять им
- Узнаем, как подключать шаговый двигатель и управлять им с помощью библиотеки
- Научимся подключать шаговый двигатель через специальный драйвер
- Узнаем, как подключать и использовать сервопривод



План занятия

- 1 Как подключить электромагнитное реле
- 2 Как подключить шаговый двигатель
- 3 Как подключить сервопривод
- 4 Итоги



Как управлять электромагнитным реле



1

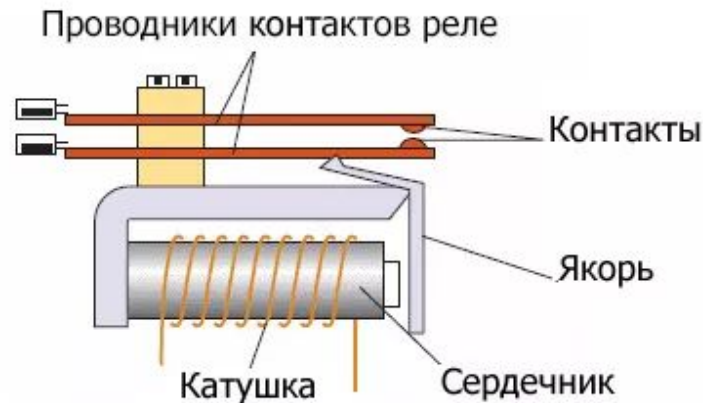
Назначение электромагнитного реле

Замыкает или размыкает
электрическую цепь при протекании
электрического тока через
управляющую обмотку



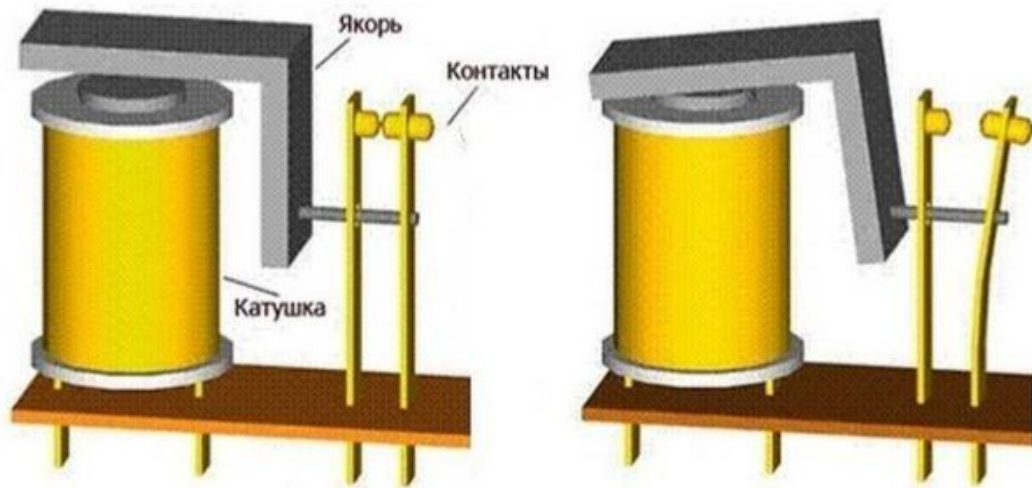
Внутреннее устройство электромагнитного реле

Работа электромагнитных реле основана на использовании электромагнитных сил, возникающих в металлическом сердечнике при прохождении тока по виткам его катушки



Нормально замкнутые контакты реле

Нормально замкнутые контакты (обозначение НЗ или NC) в исходном состоянии замкнуты, т.е. через них протекает ток. При срабатывании реле контакты размыкаются, цепь разрывается



Нормально разомкнутые контакты реле

Нормально разомкнутые контакты (обозначение НО или NO) в исходном состоянии разомкнуты, т.е. через них не протекает ток. При срабатывании реле контакты замыкаются, формируется электрическая цепь



Переключающие контакты реле

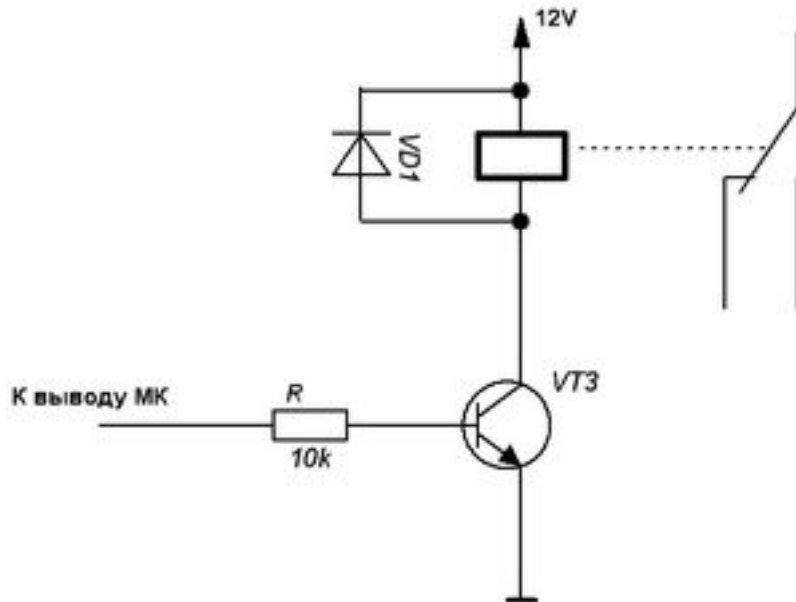
Состоит из трех пластин и объединяет в себе функции нормально замкнутых и нормально разомкнутых контактов



Подключение электромагнитного реле

Порт микроконтроллера не может обеспечить ток через катушку электромагнитного реле, поэтому необходимо использовать ключ на транзисторе.

Диод - для защиты от выброса напряжение при размыкании цепи катушки.



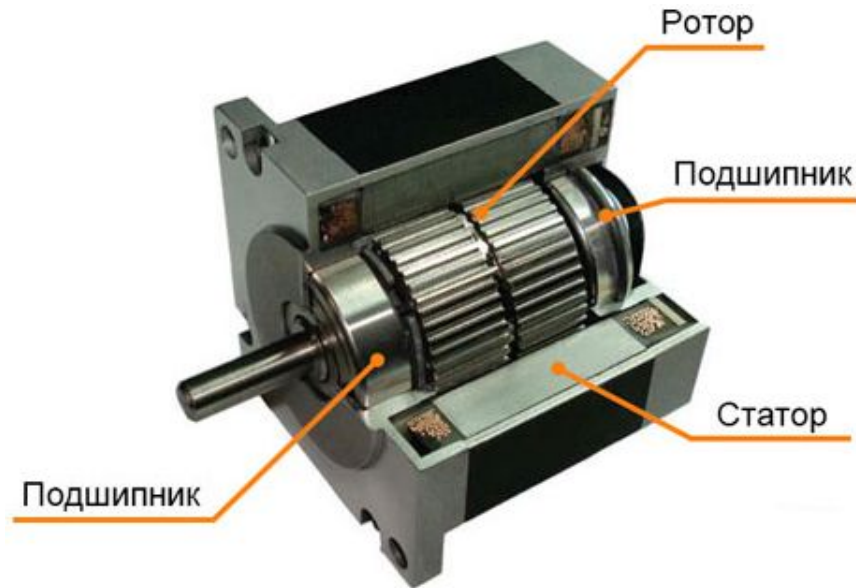
Как подключить шаговый двигатель



2

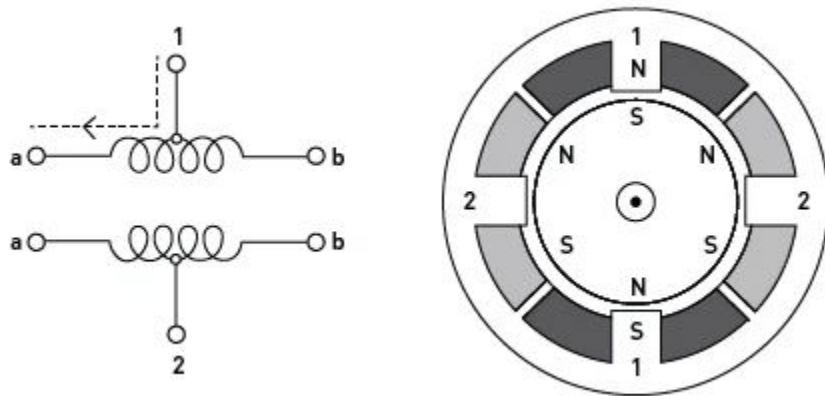
Конструкция шагового двигателя

Шаговый двигатель, как и любой вращающийся электродвигатель, состоит из ротора и статора. Статор - неподвижная часть, ротор - вращающаяся часть



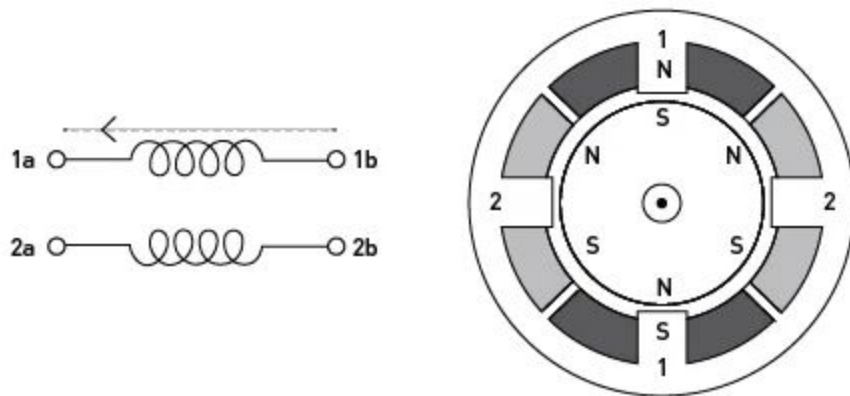
Шаговые двигатели с постоянными магнитами

Униполярный шаговый двигатель с постоянными магнитами имеет одну обмотку на фазу с ответвлением в центре. Каждая секция обмотки включается отдельно. Обычно центральное ответвление каждой фазы делается общим.



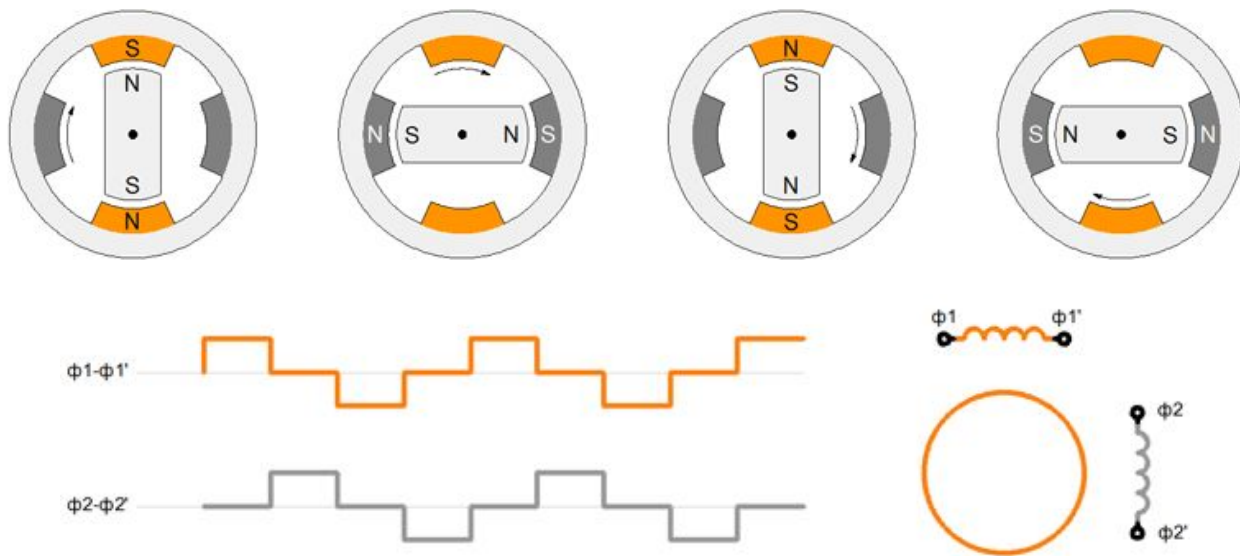
Шаговые двигатели с постоянными магнитами

Двухполярные двигатели имеют одну обмотку на фазу. Для того чтобы изменить магнитную полярность полюсов необходимо изменить направление тока в обмотке, для этого схема управления должна быть более сложной, обычно с H-мостом.



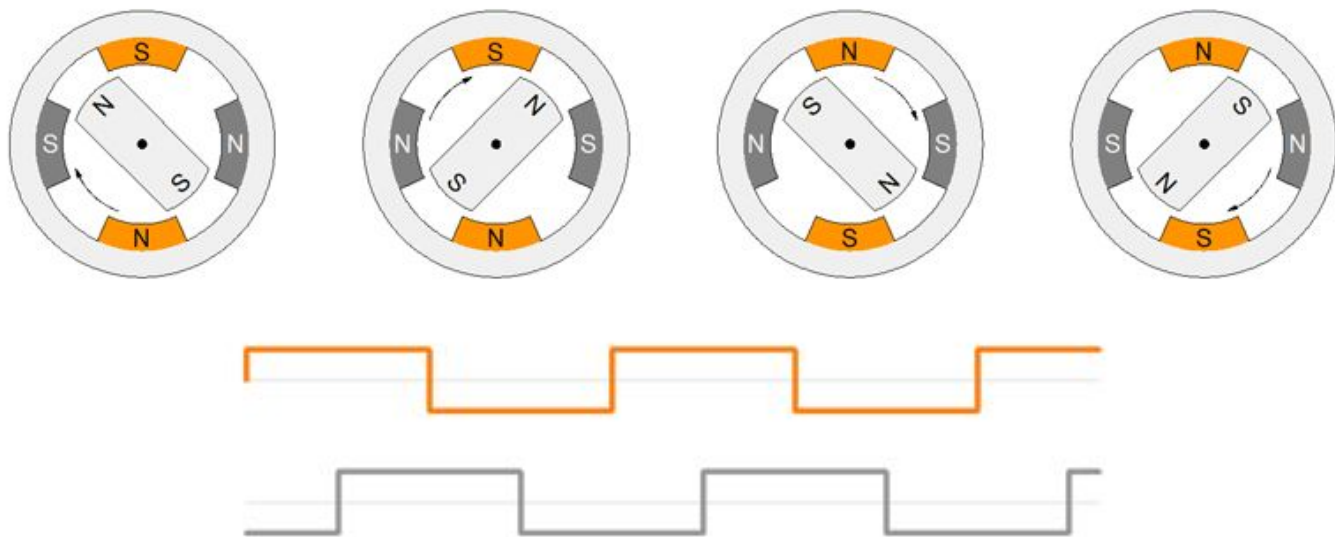
Волновое управление шаговым двигателем

При таком управлении в один момент времени возбуждается только одна обмотка. Но такой способ управления не обеспечивает максимально возможного момента.



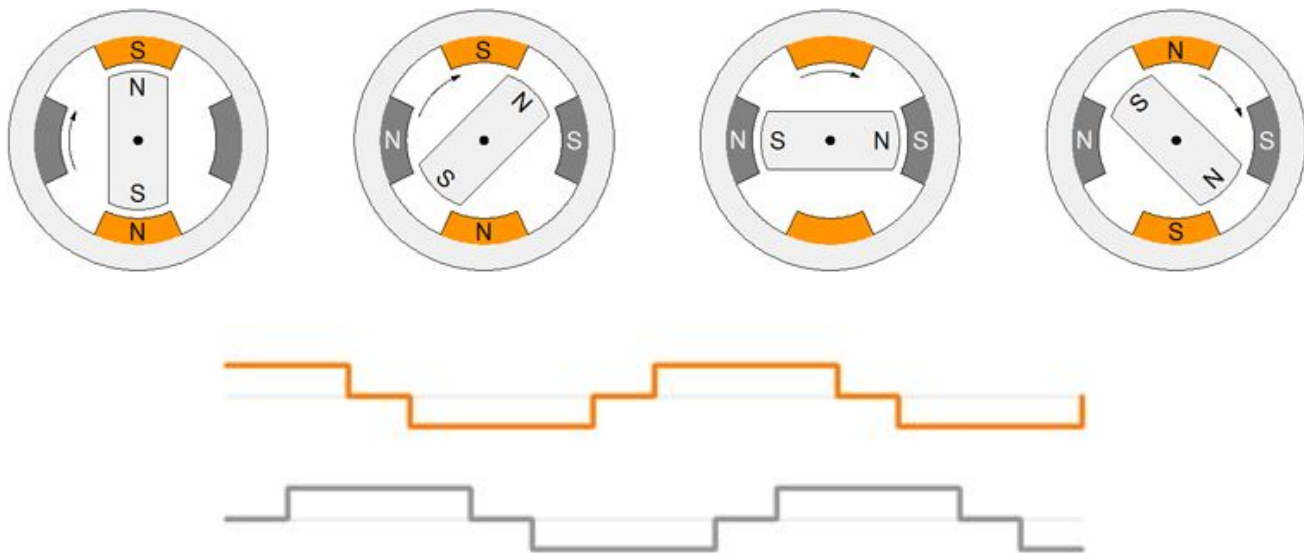
Полношаговое управление шаговым двигателем

Полношаговое управление обеспечивает больший момент, чем волновое управление так как обе обмотки двигателя включены одновременно.



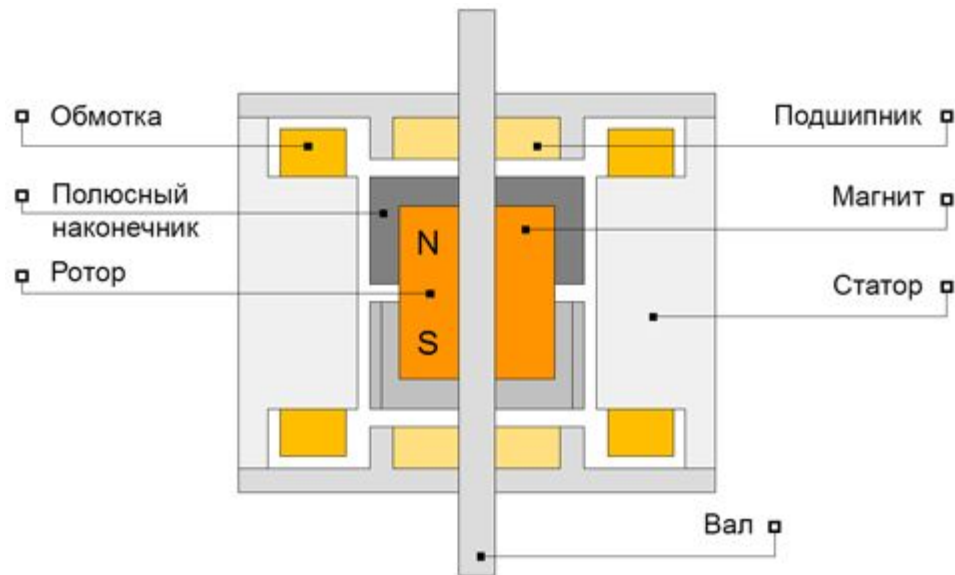
Полушаговое управление шаговым двигателем

Шаг для данной геометрии шагового двигателя делится пополам. Полушаговое управление обеспечивает большее разрешение при позиционировании вала.



Гибридный шаговый двигатель

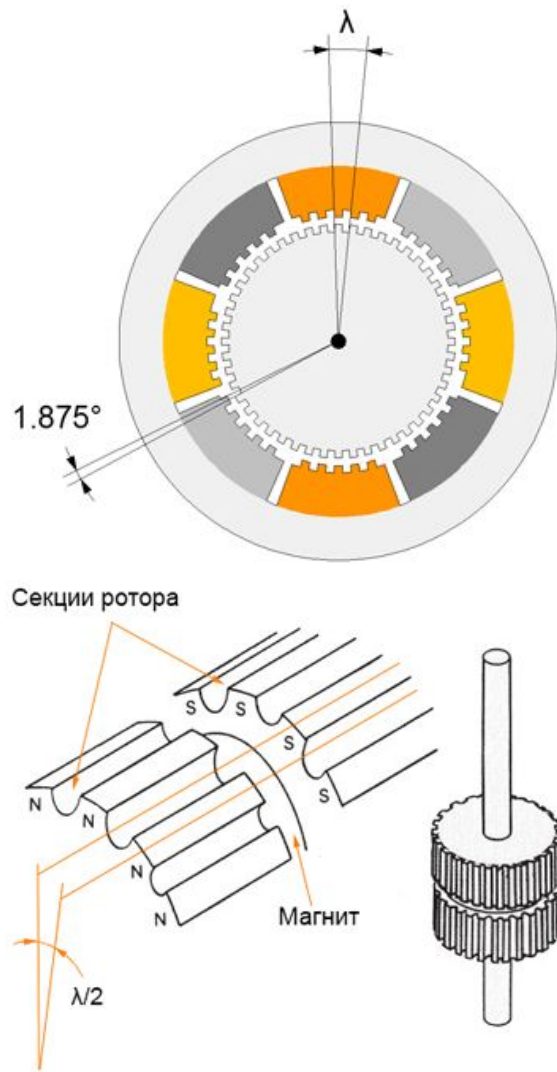
Ротор гибридного шагового двигателя представляет из себя цилиндрический постоянный магнит, намагниченный вдоль продольной оси с радиальными зубьями из магнитомягкого материала



Гибридный шаговый двигатель

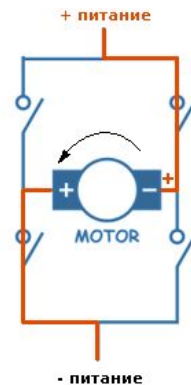
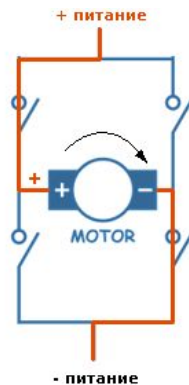
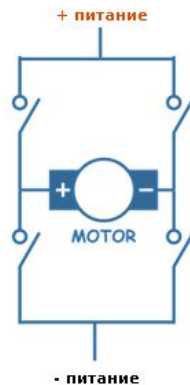
48 зубьев на одной секции ротора смещены на половину зубцового деления λ относительно другой секции. Из-за этого смещения ротор фактически имеет 96 перемежающихся полюсов противоположной полярности.

[Источник](#)



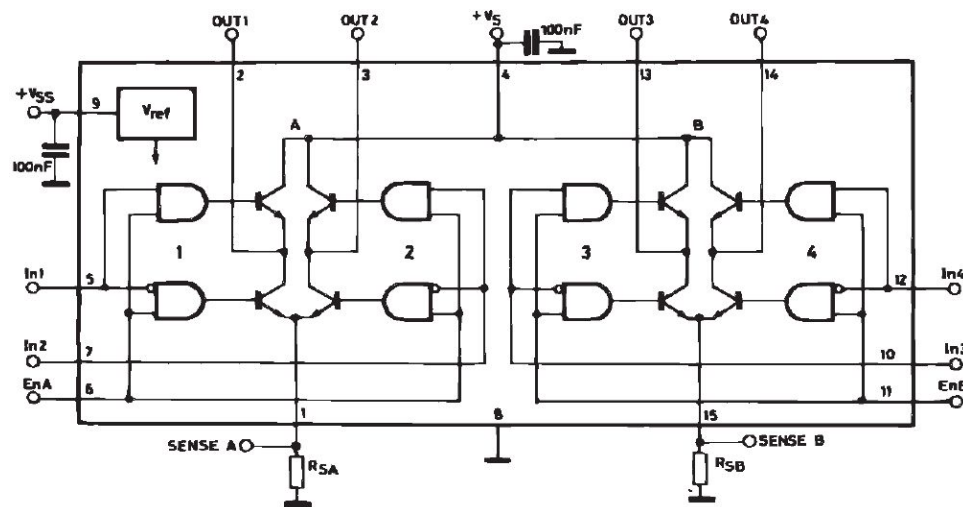
Использование Н-моста

Чтобы иметь возможность управлять обмоткой двигателя, необходимо выполнять коммутацию ее питания с переполюсовкой. Для подобной цели удобно использовать Н-мост



Микросхема Н-моста

Микросхема L298 - это универсальный мостовой драйвер для управления двигателями постоянного тока, шаговыми двигателями, электромагнитными реле и электромагнитами (соленоидами).



Библиотека Stepper для работы с шаговым двигателем

`Stepper(int number_of_steps, int motor_pin_1, int motor_pin_2)`
`Stepper(int number_of_steps, int motor_pin_1, int motor_pin_2,`
`int motor_pin_3, int motor_pin_4)` — конструкторы,
создающий объект Stepper

Параметры:

- `number_of_steps`: количество шагов в полном обороте используемого двигателя. Если в документации к двигателю указан угол одного шага, то следует разделить 360° на этот угол, что даст нам искомое количество шагов;
- `motor_pin_1`, `motor_pin_2`, `motor_pin_3`, `motor_pin_4`: номера выводов для подключения двигателя

Возвращаемое значение: нет

Библиотека Stepper для работы с шаговым двигателем

`void setSpeed(long whatSpeed)` — устанавливает скорость вращения в оборотах в минуту. Эта функция не заставляет двигатель вращаться, а лишь устанавливает скорость вращения, которая будет использована при вызове функции `step()`

Параметры:

- `whatSpeed`: скорость, на которой будет производиться вращение шагового двигателя, выражается в оборотах в минуту

Возвращаемое значение: нет

`void step(int number_of_steps)` — вращает шаговый двигатель на определенное количество шагов на скорости, заданной функцией `setSpeed()`

Параметры:

- `number_of_steps`: количество шагов, знак задает направление вращения

Возвращаемое значение: нет

Управление скоростью вращения с помощью потенциометра

```
#include <Stepper.h> //подключение библиотеки

const int stepsPerRevolution = 200; //количество шагов на оборот
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11); //управление выводами 8 - 11

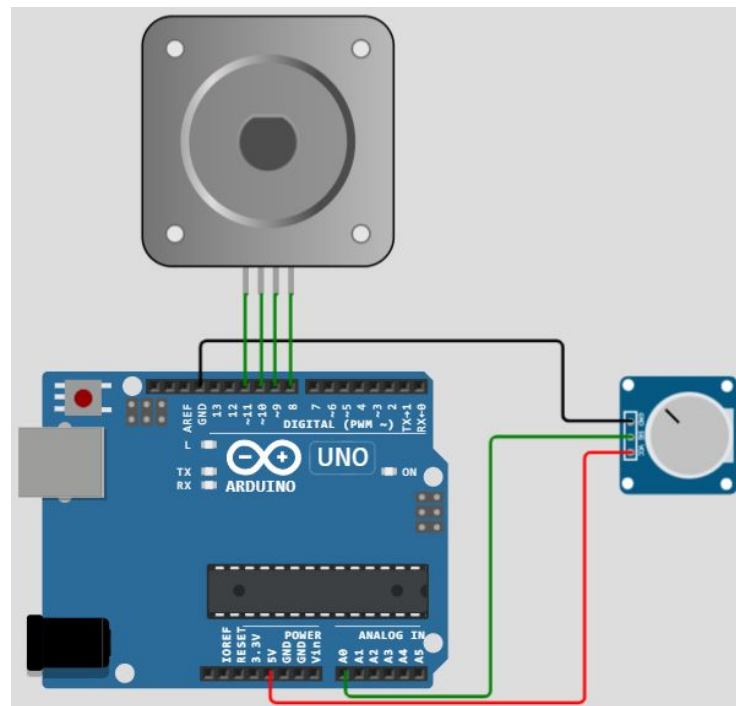
int stepCount = 0; //количество шагов, которое сделал двигатель

void setup() {
}

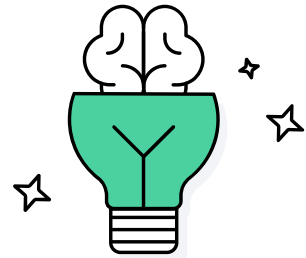
void loop()
{
    int sensor = analogRead(A0); // чтение значения потенциометра
    int motorSpeed = map(sensor, 0, 1023, 0, 100); // масштабирование этого значения
    if (motorSpeed > 0)
    {
        myStepper.setSpeed(motorSpeed); //установка нового значения скорости
        myStepper.step(stepsPerRevolution / 100); // сделать 1/100 полного оборота
    }
}
```


Управление скоростью вращения с помощью потенциометра

На схеме не показана микросхема H-моста



Практическое задание N°1



Практика: управление скоростью вращения с помощью потенциометра

Задание:

- 1) соберите схему в симуляторе WOKWI, подключив шаговый двигатель к выводам с 8 по 11, а потенциометр к выводу A0;
- 2) создайте скетч с текстом, приведенным выше;
- 3) проведите моделирование работы

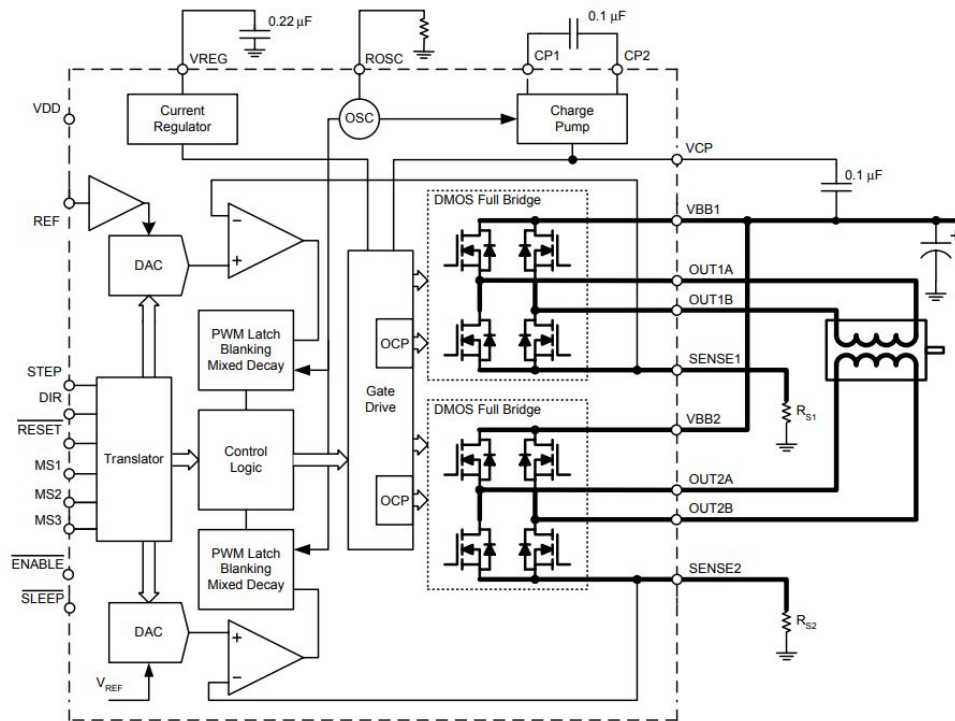
Как выполнять: напишите в чат об удачной работе схемы

Время выполнения: 5 минут



Драйвер шагового двигателя A4988

Драйвер шагового двигателя содержит силовые ключи для управления обмотками двигателя, и схему управления.



Подключение драйвера A4988

ENABLE – включение/выключение драйвера
MS1, MS2, MS3 – контакты для установки микрошага

RESET - сброс микросхемы

STEP - генерация импульсов для движения двигателей (каждый импульс – шаг), можно регулировать скорость двигателя

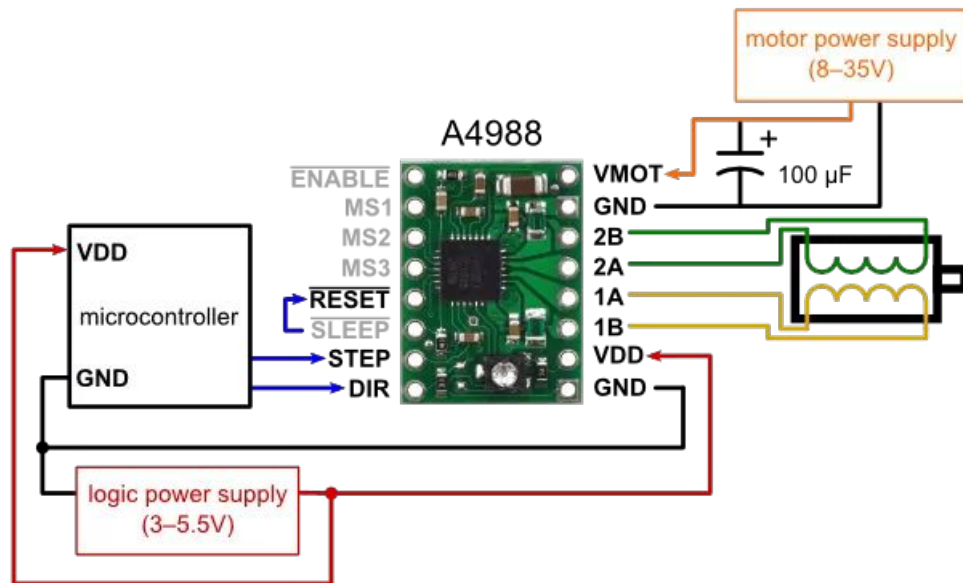
DIR – установка направление вращения

VMOT – питание для двигателя

GND – общий

2B, 2A, 1A, 1B – для подключения обмоток двигателя

VDD – питание микросхемы



Микрошаговый режим шагового двигателя

Микрошаг - режим деления шага, при работе в котором обмотки шагового двигателя в каждый момент времени запитаны не полным током, а его уровнями, изменяющимися по закону \sin в одной фазе и \cos во второй. Такой метод дает возможность фиксировать вал в промежуточных положениях между шагами.



Использование драйвера шагового двигателя

Пример программы:

```
const int dirPin = 2; //вывод для направления
const int stepPin = 3; //вывод для шага
const int stepsPerRevolution = 200; //количество шагов на
один оборот

void setup()
{
    pinMode(stepPin, OUTPUT); //выводы как выходы
    pinMode(dirPin, OUTPUT);
}
```

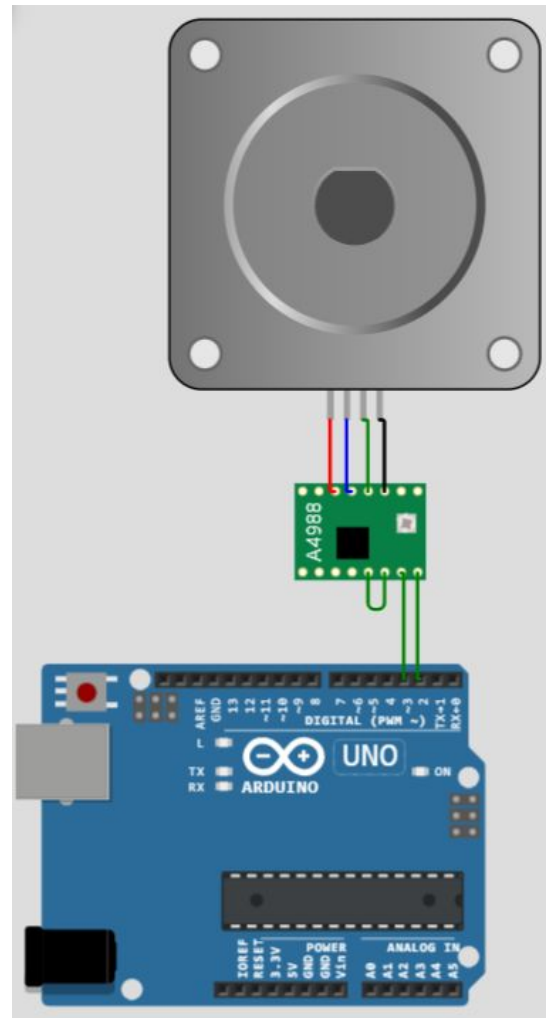
Использование драйвера шагового двигателя

```
void loop()
{
    digitalWrite(dirPin, HIGH); //установка направления
    for(int x = 0; x < stepsPerRevolution; x++) // медленный поворот двигателя
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(2000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(2000);
    }
    delay(1000); // ожидание  секунды

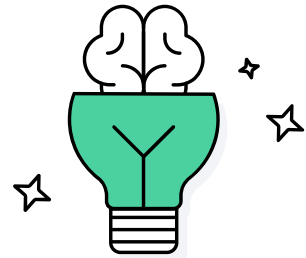
    digitalWrite(dirPin, LOW); //противоположное направление
    for(int x = 0; x < stepsPerRevolution; x++) // быстрый поворот двигателя
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(1000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(1000);
    }
    delay(1000); // ожидание  секунды
}
```


Использование драйвера шагового двигателя

Не показан источник питания для двигателя



Практическое задание №2



Практика: использование драйвера шагового двигателя

Задание:

- 1) соберите схему в симуляторе WOKWI, драйвер A4988 к выводам 2 и 3 платы Arduino UNO, а шаговый двигатель - к выводам драйвера;
- 2) создайте скетч с текстом, приведенным выше;
- 3) проведите моделирование работы

Как выполнять: напишите в чат об удачной работе схемы

Время выполнения: 5 минут



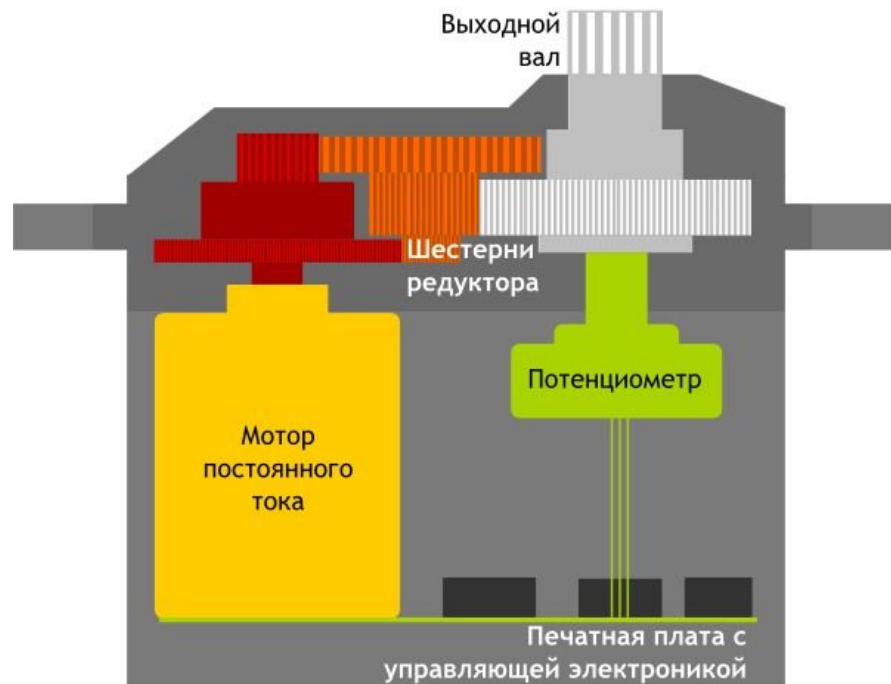
Как подключить сервопривод



3

Конструкция сервопривода

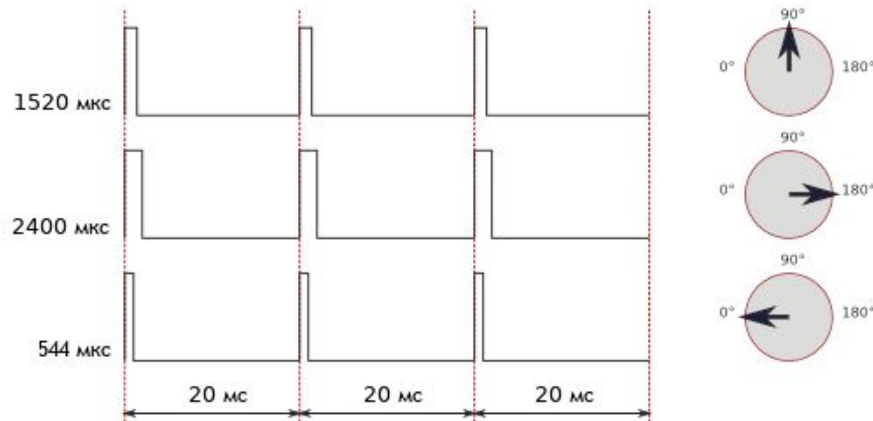
Сервопривод (или следящий привод) — механический привод с автоматической коррекцией состояния через внутреннюю отрицательную обратную связь, в соответствии с параметрами, заданными извне



Управление сервоприводом

Для управления цифровым сервоприводом используется широтно-импульсная модуляция. Ширина импульса может задавать:

- угол поворота сервопривода;
- скорость вращения



Библиотека Servo для работы с сервоприводом

На всех платах кроме Arduino Mega, при работе с данной библиотекой, пропадает возможность использовать цифровые выводы 9 и 10 в режиме ШИМ.

`Servo()` — конструктор, создающий объект Servo

Параметры: нет

Возвращаемое значение: нет

Библиотека Servo для работы с сервоприводом

`uint8_t attach(int pin, int min, int max)` — указывает вывод к которому подключен сервопривод

Параметры:

- `pin`: вывод, к которому подключен сигнальный провод сервопривода
- `min`: (необязательный параметр) ширина импульса в микросекундах, соответствующая минимальному (угол 0 градусов) положению сервопривода (по умолчанию 544)
- `max`: (необязательный параметр) ширина импульса в микросекундах, соответствующая максимальному (угол 180 градусов) положению сервопривода (по умолчанию 2400)

Возвращаемое значение: индекс созданного канала управления

`void detach()` — отсоединяет экземпляр класса от вывода. При отсоединения всех сервоприводов заблокированные ШИМ выводы снова станут доступны

Параметры: нет

Возвращаемое значение: нет

Библиотека Servo для работы с сервоприводом

`bool attached()` — проверяет, указан ли управляющий вывод для экземпляра класса Servo

Параметры: нет

Возвращаемое значение: true — если вывод был указан и false — если нет

`void write(int value)` — поворачивает сервопривод на заданный угол. Для сервоприводов постоянного вращения устанавливает скорость и направление вращения

Параметры:

- value: устанавливает угол от 0 до 180 градусов. При использовании сервопривода постоянного вращения значение 90 используется для неподвижного состояния, значение 0 - для максимальной скорости вращения в одну сторону, а 180 - для максимальной скорости вращения в другую сторону

Возвращаемое значение: нет

Библиотека Servo для работы с сервоприводом

`void writeMicroseconds(int value)` — поворачивает сервопривод на угол заданный в микросекундах. С сервоприводами постоянного вращения работает по такому же принципу как и функция `write()`

Параметры:

- `value`: значение в микросекундах

Возвращаемое значение: нет

`int read()` — возвращает текущее положение сервопривода

Параметры: нет

Возвращаемое значение: значение угла в диапазоне от 0 до 180

Слежение за положением потенциометра

```
#include <Servo.h> //подключение библиотеки

Servo myservo; // создание объекта

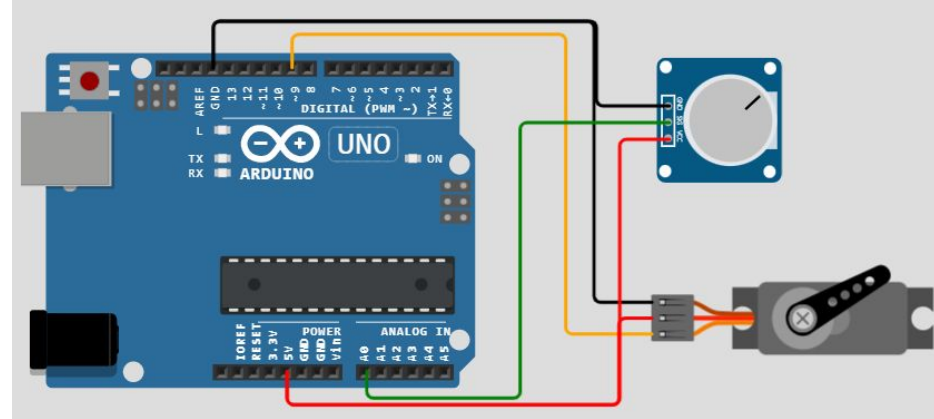
int potpin = A0; // вывод для потенциометра
int val; // значение на выходе потенциометра

void setup() {
    myservo.attach(9); // привязка линии управления к выводу 9
}

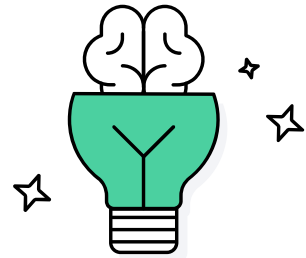
void loop() {
    val = analogRead(potpin); // чтение значения потенциометра
    val = map(val, 0, 1023, 0, 180); // масштабирование этого значения
    myservo.write(val); // установка новой позиции сервопривода
    delay(15); // отладка
}
```

Слежение за положением потенциометра

Плата Arduino UNO поддерживает до 12 каналов сервоприводов



Практическое задание №3



Практика: слежение за положением потенциометра

Задание:

- 1) соберите схему в симуляторе WOKWI, подключив сервопривод к выводу 9, а потенциометр - к выводу A0;
- 2) создайте скетч с текстом, приведенным выше;
- 3) проведите моделирование работы

Как выполнять: напишите в чат об удачной работе схемы

Время выполнения: 5 минут



Итоги

4

Итоги занятия

Сегодня мы

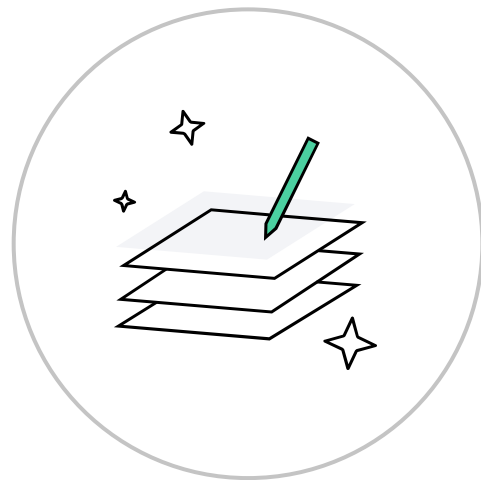
- 1 Узнали особенности подключения электромагнитного реле
- 2 Научились подключать шаговый двигатель и управлять им с помощью специальной библиотеки
- 3 Научились подключать шаговый двигатель через микросхему драйвера
- 4 Научились использовать сервопривод и управлять им



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Павел Пронин
C++ разработчик

