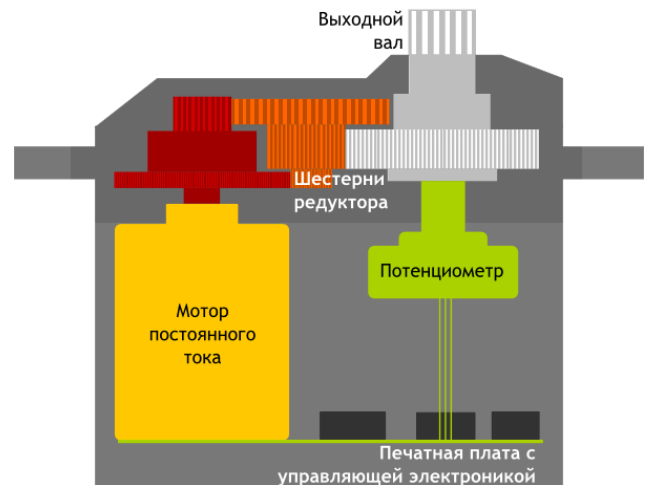


Курс «Основы робототехники». Занятие 13. Сервомоторы, манипуляторы

Сервомотор (далее серво) позволяет поворачивать вал на заданный угол и удерживать его на этом угле. Используется в рулях моделей машин и самолетов, манипуляторах, руках роботов, шагоходах и т.п.

Серво SG90



Весьма дешевый серво, хорошо подходит для учебных моделей. Мощность и надежность малы, поэтому для практического применения лучше использовать более мощные (но и более дорогие) серво. Рассмотренные ниже схемы подключения и программы управления подходят и для более мощных серво (с учетом того, что их обязательно нужно питать от внешнего источника питания). **Параметры SG90:**

- Рекомендуемое напряжение питания от 4,8 до 6 В (по испытаниям, нормально работает от двух аккумуляторов Li-Ion 18650, дающих $2 \cdot 3,7 = 7,4$ В, но большую нагрузку при этом давать не следует и при перегреве серво немедленно выключать его)
- Угол поворота вала от 0 до 180°. На крайних значениях работает плохо, рекомендуется использовать в диапазоне от 10 до 170° и менее.

Подключение серво:

- коричневый или черный провод к Gnd
- красный провод к + напряжения питания
- белый, желтый или оранжевый провод к цифровому пину Arduino

Примечание: рекомендуется питать серво от своего источника питания, соединив минус источника с Gnd Arduino. По результатам испытаний, SG90 можно

питать от +5 В Arduino, но не более четырех сразу и с небольшой нагрузкой. При этом Arduino должна питаться от USB, а не от внутреннего стабилизатора.

Библиотека Servo

Базовая библиотека, используемая для управления сервомоторами SG90 и другими. Команды библиотеки:

`#include <Servo.h>` – подключить библиотеку Servo, добавляется в начале программы

`Servo myservo;` – создать объект для контроля определенного серво. *myservo* – название серво, оно может быть любым. Добавляется в начале программы

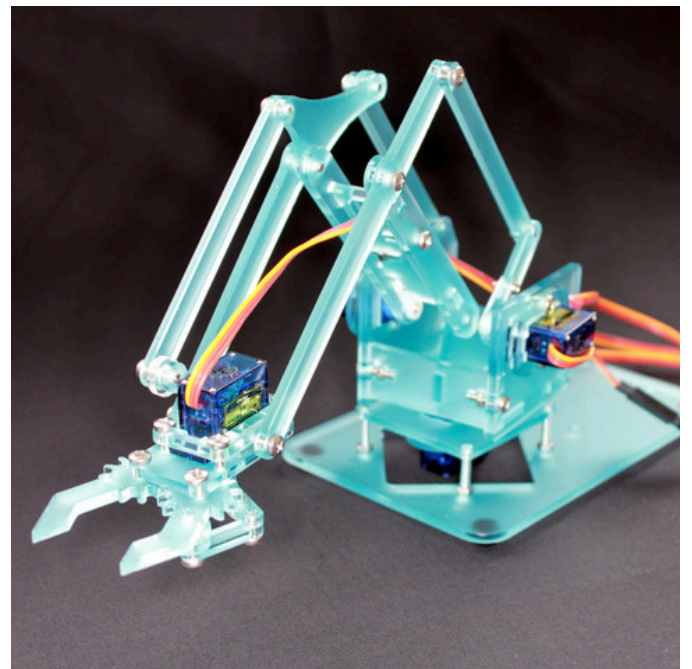
`myservo.attach(9);` – подключить серво с названием *myservo* к номеру пина в скобках (сигнальный провод серво должен быть физически подключен к этому пину). Добавляется в блоке *setup*

`myservo.write(30);` – повернуть серво с названием *myservo* на угол, заданный в скобках

Манипулятор meArm

Простой и дешевый манипулятор с открытыми чертежами. По этим чертежам может быть вырезан из оргстекла или фанеры на лазерном резке и собран по инструкции, также выложенной разработчиками в открытый доступ. Содержит 4 серво:

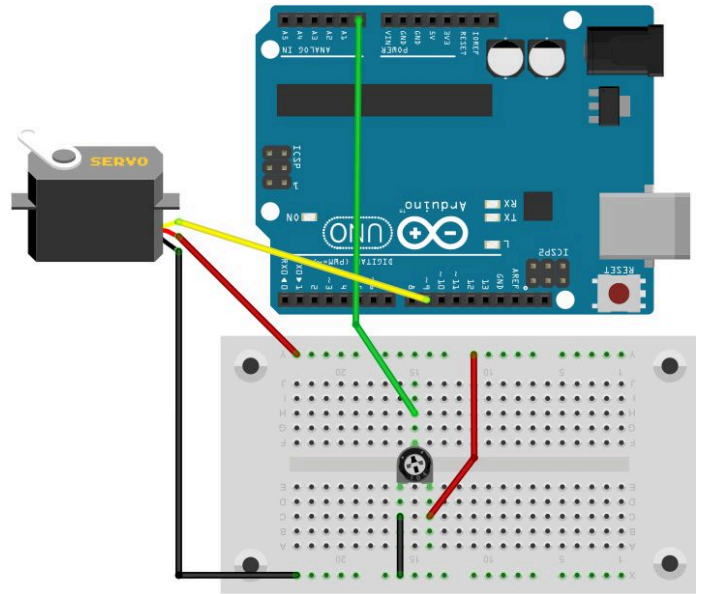
- Вращение манипулятора вокруг оси
- Подъем-опускание захвата
- Движение захвата от- и к манипулятору
- Открывание/закрывание захвата



Практическое занятие 1. Движение серво по циклу

1. Наденьте на серво один из рычагов, имеющихся в комплекте с ним. Привинчивать рычаг не нужно
2. Вставьте в разъем серво провода штырек-штырек: красный к красному проводу серво, черный или синий к коричневому, любой другой цвет к оранжевому

3. Подключите серво к стенду (красный к +5 В, черный или синий к Gnd, сигнальный провод к 9 пину Arduino, пример подключения см. ниже)
4. Откройте пример работы с библиотекой *Файл / Примеры / Servo / Sweep* и загрузите пример его в контроллер
5. Убедитесь, что рычаг серво движется по указанному циклу – от 0 до 180° и обратно



Практическое занятие 2.

Управляемое движение серво

1. Подключите переменный резистор в соответствии со схемой
2. Откройте пример работы с библиотекой *Файл / Примеры / Servo / Knob* и загрузите его в контроллер
3. Убедитесь, что рычаг серво движется при вращении переменного резистора

Практическое занятие 3. Движение двух серво по циклу

1. Откройте пример работы с библиотекой *Файл / Примеры / Servo / Sweep*
2. Внесите следующие изменения в программу:

Servo myservo;	Servo myservo1; Servo myservo2;
myservo.attach(9);	myservo1.attach(9); myservo2.attach(7);
myservo.write(pos);	myservo1.write(pos); myservo2.write(pos);

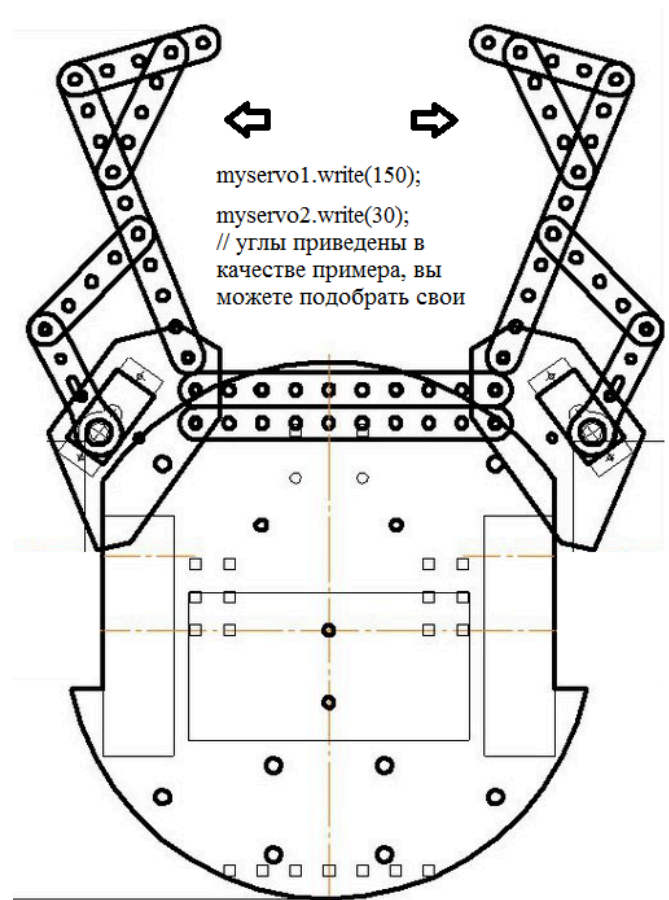
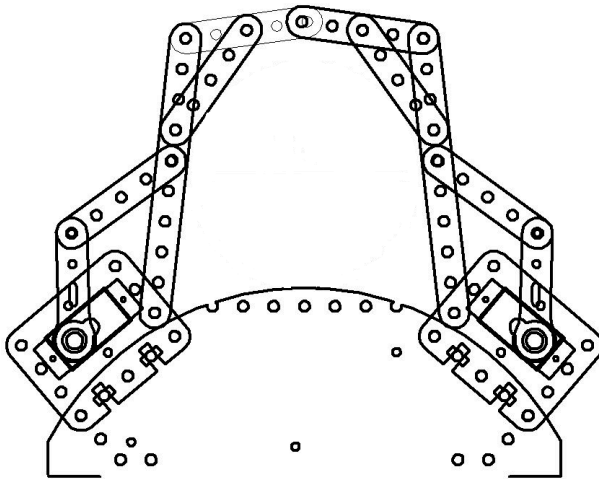
3. Подключите еще один серво к пину 7, загрузите программу в контроллер и убедитесь, что вращаются оба серво

Примечание: таким образом можно управлять большим количеством серво, подключив их к цифровым пинам Arduino и подавая на них командами *myservoX.write(Y)* различные углы. Например, для четырехногого шагохода с 2 серво в каждой ноге требуется $4 \cdot 2 = 8$ серво, для шестиногого с 3 серво в каждой ноге – $6 \cdot 3 = 18$ серво (в последнем случае серво можно подключить к аналоговым пинам, используя их как цифровые A0–14, A1–15, ..., A5–19)

4. Сохраните программу под другим именем, она понадобится далее

Практическое занятие 4. Управление захватом на двух сервоприводах

1. Попробуйте реализовать сжатие и разжатие захвата. Чтобы сервоприводы двигались в противоположные стороны



Практическое занятие 6. Управление захватами по Bluetooth через приложение Bluetooth rc car

Как вы уже заметили, в приложении в правом верхнем углу есть ползунок. Его мы и будем использовать для управления захватами.

Ползунок имеет 10 значений от '0' до '9' и 'q' вместо числа 10

Дополните программу управления роботом по Bluetooth и добавьте функцию управления захватами.

Пример:

```
//В функцию switch дописываем
case '0':
myservo1.write(150);
myservo2.write(30);
// когда ползунок в левом конце, захваты максимально
```

раздвинуты (или сдвинуты)

break;

case '1':

myservo1.write(140);

myservo2.write(40);

// когда ползунок четъ правее, захваты немного раздвигаются

break;

case '2':

myservo1.write(130);

myservo2.write(50);

// когда ползунок ещё немного правее, захваты раздвигаются ещё

break;

// и так далее до значения 9