

Движение и повороты

Занятие 12

Роботы-помощники

- Один из распространенных типов мобильного робота — автономный колёсный робот. Хотя в домах их встретишь нечасто, они широко используются для автоматизации задач на заводах и складах по всему миру.
- Самая элементарная задача для мотора любого колёсного робота — обеспечить точные управляемые движения.

Вопросы после видео

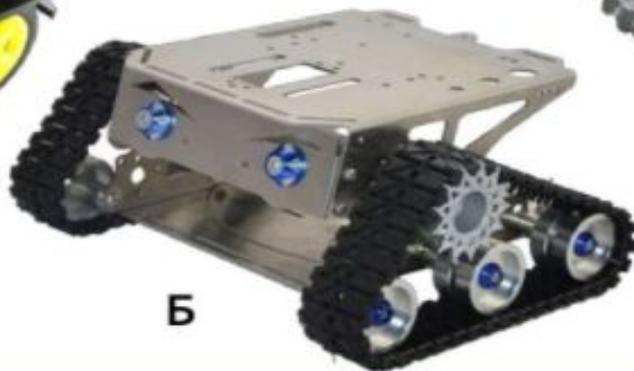
- Как можно настроить и запрограммировать колёсных роботов для выполнения конкретных задач?
- Какие движения они должны быть в состоянии совершать?
- Как они могут безопасно работать вместе с людьми?

Виды шасси

Моделирование шасси



А



Б



В

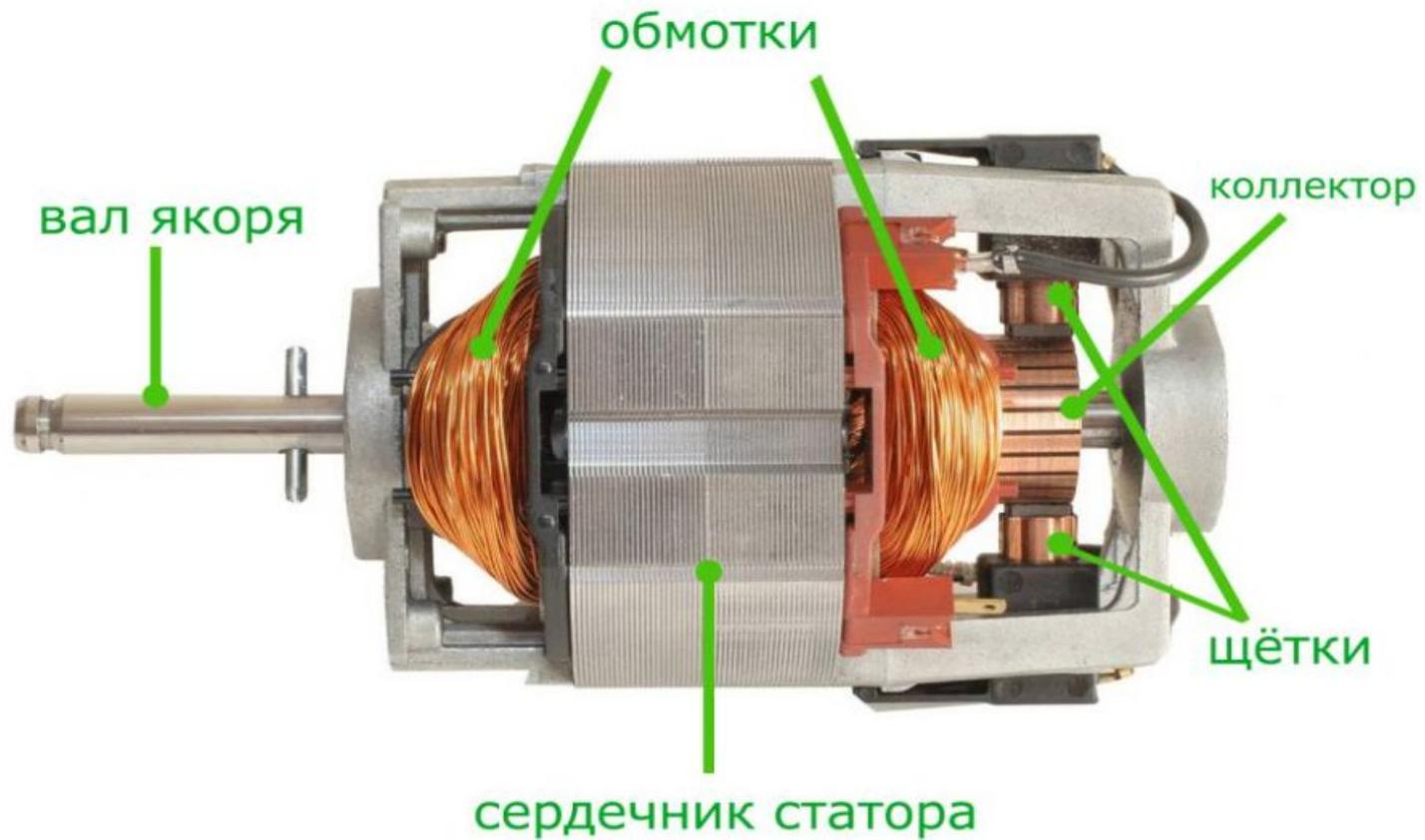


Г

Коллекторный электродвигатель

- Коллекторный двигатель состоит из подвижной части — ротор и неподвижной — статор (корпус).
- Коллектор - набор контактов, расположены на роторе и щётки - скользящие контакты, расположены вне ротора и прижаты к коллектору.
- Ротор с обмотками вращается внутри статора.

Устройство



Мотор - редуктор



Передаточное число редуктора:
1:120

Скорость вращения выходного
вала редуктора: 80 об/мин (при
напряжении питания 6V)

Напряжение питания моторов: 3
~ 6V

Колеса:

Диаметр колеса 65 мм

Motor Shield

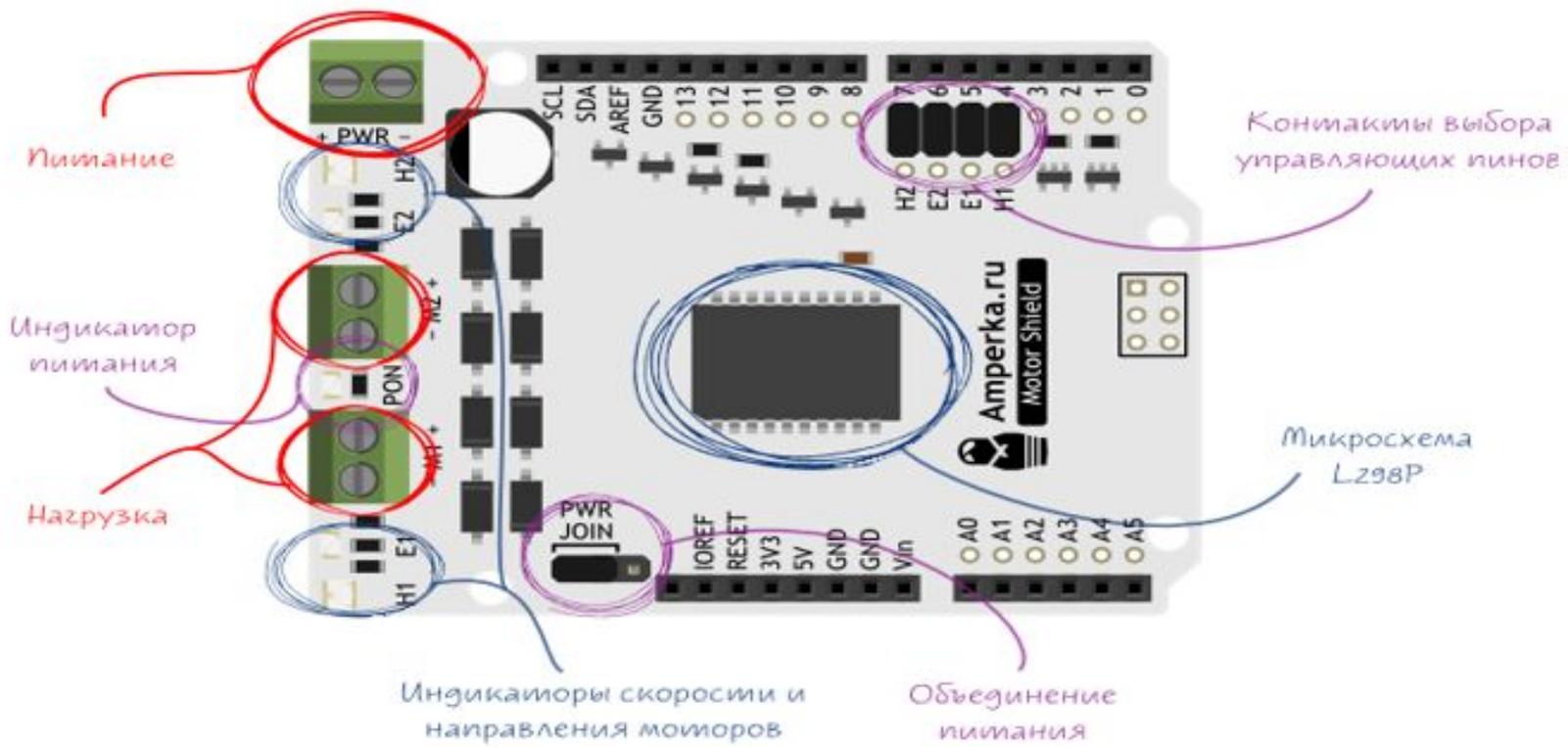


Motor Shield — плата расширения для Arduino на базе чипа L298P, позволяющая управлять моторами с напряжением 5–24 В в режиме отдельного питания и 7–12 В в режиме объединённого питания.

Плата имеет 2 независимых канала. Используя их, можно подключить на выбор:

Пару DC-моторов

Подключение



Распиновка



Для коммуникации с микроконтроллером используются цифровые контакты Arduino:

4 — направление, M1

5 — скорость (ШИМ), M1

6 — скорость (ШИМ), M2

7 — направление, M2

Программа

drive | Arduino 1.8.8

Файл Правка Скетч Инструменты Помощь



drive

```
#define E1 6 //Вывод платы расширения Motor Shield, используемый для контроля скорости мотора А.
#define E2 5 //Вывод платы расширения Motor Shield, используемый для контроля скорости мотора В.
#define M1 7 //Вывод платы расширения Motor Shield, используемый для контроля направления вращения мотора А.
#define M2 4 //Вывод платы расширения Motor Shield, используемый для контроля направления вращения мотора В.

int motorSpeed = 140;          //Текущая скорость вращения моторов.

void motor_drive_config()     //Инициализация выводов платы расширения Motor Shield.
{
    pinMode(M1, OUTPUT);      //Назначение вывода M1 выходом.
    pinMode(M2, OUTPUT);      //Назначение вывода M2 выходом.
    pinMode(E1, OUTPUT);      //Назначение вывода E1 выходом.
    pinMode(E2, OUTPUT);      //Назначение вывода E2 выходом.
}
```

Управление

loop()

Задаем направление работы первого мотора M1 (HIGH — в одну сторону, LOW — в другую):

```
digitalWrite(H1, HIGH);
```

Задаем направление работы второго мотора M2 (HIGH — в одну сторону, LOW — в другую):

```
digitalWrite(H2, HIGH);
```

Задаем скорость вращения первого мотора M1 от 0 до 255:

```
analogWrite(E1, 255);
```

Задаем скорость вращения второго мотора M2 от 0 до 255:

```
analogWrite(E2, 255);
```

Время работы моторов в указанном выше режиме:

```
delay(2000);
```

Вперёд, стоп, назад

```
void move_forward()           //Движение робота вперед.
{
    digitalWrite(M1, HIGH);    //Правое колесо вращается вперед.
    digitalWrite(M2, HIGH);    //Левое колесо вращается вперед.
    analogWrite(E1, motorSpeed); //Скорость вращения правого колеса.
    analogWrite(E2, motorSpeed); //Скорость вращения левого колеса.
}

void move_stop()              //Остановка робота.
{
    digitalWrite(M1, LOW);     //Правое колесо вращается назад.
    digitalWrite(M2, LOW);     //Левое колесо вращается назад.
    analogWrite(E1, 0);        //Снизить скорость вращения колеса до 0.
    analogWrite(E2, 0);        //Снизить скорость вращения колеса до 0.
}

void move_back()              //Движение робота назад.
{
    digitalWrite(M1, LOW);     //Правое колесо вращается назад.
    digitalWrite(M2, LOW);     //Левое колесо вращается назад.
    analogWrite(E1, motorSpeed); //Скорость вращения правого колеса.
    analogWrite(E2, motorSpeed); //Скорость вращения левого колеса.
}
```

Вправо, влево

```
void move_right()           //Движение робота вправо.
{
  digitalWrite(M1, LOW);    //Правое колесо вращается назад.
  digitalWrite(M2, HIGH);   //Левое колесо вращается вперед.
  analogWrite(E1, motorSpeed); //Скорость вращения правого колеса.
  analogWrite(E2, motorSpeed); //Скорость вращения левого колеса.
}

void move_left()           //Движение робота влево.
{
  digitalWrite(M1, HIGH);   //Правое колесо вращается вперед.
  digitalWrite(M2, LOW);    //Левое колесо вращается назад.
  analogWrite(E1, motorSpeed); //Скорость вращения правого колеса.
  analogWrite(E2, motorSpeed); //Скорость вращения левого колеса.
}
```

Управление движением

```
void loop() {  
  move_forward(); //Движение робота вперед.  
  delay (500);    //Продолжительность движения.  
  move_back();    //Движение робота назад.  
  delay (500);    //Продолжительность движения.  
}
```