Тема 2 · Урок 2.1

Моторы: едем и останавливаемся

Первые движения робота: вперёд, назад и обязательная остановка из программы.

2 × 45 мин | 1–2 ученика на робота | Практика

РЕЗУЛЬТАТ УРОКА

Робот проезжает вперёд, останавливается и возвращается назад по программе. Движения оформлены функциями forward(), backward() и stopMotors().

Ты научишься

- Объяснить, что задаёт контакт направления и что — контакт скорости.
- Включить и остановить моторы командами digitalWrite и analogWrite.
- Написать функции движения и вызвать их по порядку.
- Объяснить, почему движение без команды остановки не заканчивается.

Нужно знать

- 1.1 Собираем робота и оживляем плату

Понадобится

- Мотор-редуктор × 2
- Шилд моторов L298P (Амперка)
- Батарейный отсек моторов с выключателем
- USB-кабель для Arduino

Зачем этот урок

Робот собран, но пока стоит. Сегодня он поедет. Ты узнаешь, как плата командует моторами, и научишь робота ехать вперёд, назад и — самое важное — останавливаться по программе.

РАЗМИНКА · ВСПОМНИ

1. Сколько раз выполняется `setup()`, а сколько — `loop()`?
2. Что делает `digitalWrite(вывод, HIGH)`?
3. Где на твоём роботе левый мотор?

БЕЗОПАСНОСТЬ

Моторы проверяем на подставке: колёса не касаются стола. Программу загружаем при выключенном питании моторов. Чтобы остановить опыт, выключи питание моторов выключателем.

ЗАНЯТИЕ 1 · 45 МИН

1 Один мотор, потом два

1 Подготовь робота

5 мин

Что должен делать робот. Роботу нужно начать движение, проехать участок и остановиться. Для этого научимся отдельно задавать направление и мощность каждого из двух моторов, а затем объединим команды в понятные действия.

Что используем и как это работает. Моторы подключены через драйвер L298P: вывод Arduino подаёт управляющий сигнал, а ток к мотору поступает через драйвер. DIR выбирает направление. ШИМ быстро чередует включение и выключение питания; большее значение analogWrite означает большую долю включённого времени. Это команда мощности, а не измеренная скорость в сантиметрах в секунду.

Как связать измерение с действием. Если оба колеса крутятся вперёд, робот едет; если одно слабее, он отклоняется. delay только задерживает следующую команду: мотор продолжает выполнять предыдущую. Поэтому в программе нужна отдельная остановка. Сначала наблюдаем колёса на подставке, затем проверяем направление, расстояние и остановку на поле.

1. Поставь робота на подставку так, чтобы оба колеса висели в воздухе.
2. Найди выключатель питания моторов и выключи его.
3. Подключи USB-кабель.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Колёса висят, питание моторов выключено, плата подключена к компьютеру.

2 Включи левый мотор

15 мин

Левый мотор подключён к контактам `7` (направление) и `6` (скорость). Направление «вперёд» на этом комплекте — уровень `HIGH`.

Как это устроено: Справка по командам

Команда	Что делает
`pinMode(pin, OUTPUT)`	Разрешает плате управлять контактом
`digitalWrite(pin, HIGH)` / `LOW`	Направление мотора
`analogWrite(pin, v)`	Сила мотора: `v` от 0 (стоит) до 255
`delay(ms)`	Ждать `ms` миллисекунд

Требование 1. Скетч `L21_Left_Motor`: левое колесо крутится вперёд ровно 2 секунды с силой из именованной переменной и останавливается само. Номера контактов — именованные константы, а не числа в командах.

Приёмка.

1. На подставке напарник засекает по секундомеру, сколько крутится колесо.
2. Замени уровень направления на противоположный. Что сделало колесо? Запиши, какой уровень на твоём роботе — «вперёд».
3. Объясни в журнале, почему колесо остановилось, хотя `loop()` пустой.

Подсказка: Идея (1 из 2)

Один раз при включении: настроить контакты, выбрать направление, включить, подождать, выключить.

Подсказка: Подсказка к пункту 3 (2 из 2)

Где выполнилась команда остановки — в `setup()` или в `loop()`? Сколько раз?

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Колесо крутится 2 секунды и останавливается. В журнале — время по секундомеру, уровень «вперёд» и объяснение остановки.

3 Включи оба мотора

15 мин

Требование 2. Оба колеса крутятся вперёд ровно одну секунду со скоростью из одной именованной переменной `speed` и останавливаются. Правый мотор подключён к контактам `4` (направление) и `5` (скорость). Начни со своей программы для левого мотора и сохрани её как `L21_Both_Motors`.

Исследование: с какого числа колесо трогается. На подставке ставь `speed` = 20, 40, 60, 80… и запиши для каждого колеса, при каком наименьшем числе оно начинает уверенно крутиться.

`speed`	Левое крутится?	Правое крутится?
20		
40		
60		

ПРЕДСКАЖИ

Будут ли эти числа одинаковыми у двух колёс? А у тебя и у соседней команды?

Подсказка: Колесо гудит, но не крутится

Мотору не хватает силы сдвинуть редуктор. Такое число — ещё «мёртвая зона».

ДОЛЖНО ПОЛУЧИТЬСЯ

Оба колеса крутятся одну секунду и останавливаются. В журнале — наименьшая скорость трогания для каждого колеса и вывод, какие числа `speed` брать нельзя.

4 Найди ошибку остановки

10 мин

НАЙДИ ОШИБКУ · ПУСТОЙ LOOP(), А РОБОТ ЕДЕТ

У соседей `loop()` пустой, а движение — в `setup()`:

```
void forward(int ms) {
  drive(FORWARD, speed, FORWARD, speed);
  delay(ms);
}

void setup() {
  // … настройка контактов
  forward(1000);
}

void loop() {
}
```

Робот должен был проехать одну секунду, а едет, пока его не выключат. Почему, если `loop()` пустой? Запиши ответ в журнал.

Требование. Перепиши программу так, чтобы движение осталось в `loop()`, но робот проехал ровно одну секунду и больше не трогался. `while (true)` использовать нельзя.

1. Проверь на подставке: колёса крутятся одну секунду и стоят, пока не нажмёшь Reset.
2. Объясни в журнале, как твоя программа «помнит», что движение уже было.

Подсказка: Идея

Нужна переменная, которая хранит «уже ехали» — `bool`. Её проверяют в начале `loop()`.

ДОЛЖНО ПОЛУЧИТЬСЯ

Программа выполняет требование, объяснение записано.

ЗАНЯТИЕ 2 · 45 МИН

2 Функции движения

5 Оформи движения функциями

15 мин

Требование 3. Программа `L21_Moves` построена из функций:

- `drive(leftDir, leftSpeed, rightDir, rightSpeed)` — единственное место, где задаются направление и скорость колёс;
- `forward(ms)` и `backward(ms)` — движение на заданное время, `stopMotors()` — остановка;
- направление «назад» не записано числом, а получено из `FORWARD`: на другом комплекте `FORWARD` может быть другим;
- после включения есть 2 секунды, чтобы убрать руки;
- в `setup()`: вперёд 1 секунду, стоп, пауза полсекунды, назад 1 секунду, стоп.

Подсказка: Идея (1 из 3)

Сначала одна общая функция, которая ставит направление и скорость обоих колёс. Остальные функции только вызывают её с нужными значениями.

Подсказка: План (2 из 3)

1. `drive` — четыре строки из своей программы, но с параметрами вместо чисел.
2. `forward(ms)` — `drive` вперёд и `delay(ms)`.
3. `backward(ms)` — то же назад. `stopMotors()` — скорость 0.

Подсказка: Фрагмент кода (3 из 3)

`const int BACKWARD = (FORWARD == HIGH) ? LOW : HIGH;` — прочитай как «если FORWARD равен HIGH, то LOW, иначе HIGH».

ДОЛЖНО ПОЛУЧИТЬСЯ

Программа выполняет требование 3 и проверяется без ошибок.

6 Первый заезд по полу

15 мин

> ⚠ **БЕЗОПАСНОСТЬ**
> Освободи на полу или столе место не меньше метра вперёд и назад. Держи руку рядом с выключателем.

Узнай, насколько прямо едет твой робот, и попробуй это исправить **одним изменением**.

1. Проверь программу на подставке, потом поставь робота на пол у начала метровой полосы скотча. Проедь вперёд 2 секунды. Измерь, на сколько сантиметров нос ушёл вбок от полосы. Повтори три раза.
2. Предложи одно изменение, которое уменьшит увод: например, разные скорости левого и правого колеса. Запиши, почему ты ожидаешь улучшения.
3. Сделай изменение и снова три заезда. Стало лучше? На сколько?

	Увод 1, см	Увод 2, см	Увод 3, см
До			
После			

> 💡 **Подсказка: Колёса разные**
> Если робот уходит вправо, какое колесо крутится быстрее?

> ✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**
> Две серии по три заезда, одно изменение с обоснованием и вывод по числам.

7 Измени программу

10 мин

Требование 4. Функция `shuttle(times, ms)` делает «челнок» `times` раз: вперёд `ms` миллисекунд, стоп, назад, стоп. Вызовы движений в ней написаны один раз.

Исследование: сколько проезжает робот.

1. Три раза запусти `forward(1000)` при `speed = 100` и измерь путь линейкой в миллиметрах. Повтори при `speed = 200`.
2. Посчитай, сколько миллиметров в секунду проезжает робот при каждой скорости. Во сколько раз отличаются результаты? Вдвое ли, как отличаются числа `speed`?
3. Подбери время `backward` так, чтобы после одного челнока робот вставал не дальше 2 см от старта. Запиши, почему вперёд и назад время получилось разным.

`speed`	Путь 1, мм	Путь 2, мм	Путь 3, мм	мм за секунду
100				
200				

Подсказка: Пути разные при одной программе

Это нормально: колёса проскальзывают, батарея садится. Поэтому меряем три раза и берём среднее.

ДОЛЖНО ПОЛУЧИТЬСЯ

`shuttle` работает. В журнале — таблица путей, скорость в мм/с для двух `speed`, вывод про связь числа `speed` и настоящей скорости и подобранное время возврата.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу объяснить, почему одинаковая программа даёт разный путь.
- [ ] Я знаю, почему без `stopMotors()` робот не остановится.

★★ **ПРОДВИНУТЫЙ УРОВЕНЬ · ВОЗВРАТ НА СТАРТ**

Измерь, сколько миллиметров робот проезжает за секунду, и подбери время назад так, чтобы после челнока он вставал не дальше 2 см от старта. Запиши, почему время вперёд и назад разное.

Проверь себя

★★ Продвинутый уровень

1. Робот едет «прямо», но плавно уходит вправо. Что проверить первым?

- ○ Немного уменьшить скорость левого колеса или увеличить скорость правого
- ○ Поменять местами провода обоих моторов
- ○ Увеличить время езды

2. При speed = 30 колесо гудит, но не крутится. Что это значит для программы?

- ○ Ниже порога трогания мотор не сдвигает редуктор — такие значения speed брать нельзя
- ○ Мотор неисправен, его нужно заменить
- ○ analogWrite не принимает чисел меньше 50

3. **Почему направление «назад» получают из FORWARD, а не пишут числом?**

```
1  const int BACKWARD = (FORWARD == HIGH) ? LOW : HIGH;
```

- ○ Так программа работает быстрее
- ○ На другом комплекте FORWARD может быть другим, и «назад» поменяется само
- ○ Константе нельзя присвоить LOW

4. **При speed = 100 робот за 1 секунду проехал 180, 200 и 190 мм. Какую скорость записать в журнал?**

- ○ 200 мм/с — самый большой путь
- ○ 100 мм/с — как число speed
- ○ 190 мм/с — среднее трёх замеров

5. В чём ошибка функции?

```
void shuttle(int times, int ms) {
  for (int i = 0; i < times; i++) {
    forward(ms);
    backward(ms);
  }
  stopMotors();
}
```

- ○ Между движениями нет остановки: робот резко меняет направление, и путь назад не совпадает с путём вперёд
- ○ Цикл выполнится на один раз больше, чем times
- ○ stopMotors() нельзя вызывать после цикла

Отвечай на вопросы своего уровня. Нужно верных ответов: Начальный — 4 из 5; Средний — 4 из 5; Продвинутый — 4 из 5. Ответы проверит преподаватель или программа курса.

Итог

Робот ездит по твоей программе. Главное: `analogWrite` включает мотор, и он крутится, пока другая команда его не остановит. Функции делают программу короче и понятнее. В следующем уроке робот научится поворачивать.

Как подтвердить результат

Для каждого опыта сохраняй условия, версию программы, наблюдение и вывод. Сравнивая попытки, меняй одну причину за раз; неудачные попытки тоже остаются в журнале. Отметка результата относится к уже выполненным действиям, повторять их ради отметки не нужно. Если результат не получен, запиши расхождение и следующую проверяемую гипотезу. Модель, измерения настоящего робота и объяснение — разные виды доказательств.

Словарь урока

ШИМ	Частое включение и выключение питания мотора. Число от 0 до 255 задаёт, какую долю времени мотор включён.
направление	Состояние контакта направления: одно значение крутит колесо вперёд, другое — назад.
константа	Число с именем, которое не меняется во время работы программы, например номер контакта.
функция	Именованный блок команд. Его можно вызвать много раз, написав имя и скобки.
параметр	Число, которое передают функции в скобках, например время движения.