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

Каждым мотором управляют два контакта. Контакт **направления** решает, куда крутится колесо. Контакт **скорости** решает, насколько сильно: число от 0 (стоит) до 255 (полная мощность). Левый мотор подключён к контактам `7` (направление) и `6` (скорость).

Команда	Что делает
`pinMode(контакт, OUTPUT);`	Разрешает плате управлять контактом
`digitalWrite(контакт, FORWARD);`	Выбирает направление «вперёд»
`analogWrite(контакт, сила);`	Включает мотор с силой от 0 до 255; 0 — стоп
`delay(мс);`	Ждёт указанное число миллисекунд

1. Создай скетч `L21_Left_Motor` по шаблону и напиши `setup()` сам: что должно произойти, сказано в комментариях.

Шаблон: левый мотор — L21_Left_Motor_Gaps.ino

```
// Шаблон среднего уровня. Напиши код в разделе «Твоя часть» по целям в комме

const int LEFT_DIR_PIN = 7;
const int LEFT_SPEED_PIN = 6;
const int FORWARD = HIGH;

// ===== Твоя часть: напиши сам =====

void setup() {
  // Подготовь оба контакта левого мотора.
  // Левое колесо крутится вперёд 2 секунды с силой 100 из 255 и останавливае
}

void loop() {
}
```

ПРЕДСКАЖИ

Что будет, если забыть команду остановки? Запиши ответ и проверяй его только на подставке.

2. Загрузи программу при выключенном питании моторов.
3. Включи питание моторов и нажми кнопку Reset на плате: программа начнётся заново.

Подсказка: С чего начать

Сначала оба контакта должны стать выходами, потом — направление, потом сила.

ДОЛЖНО ПОЛУЧИТЬСЯ

Левое колесо крутится вперёд 2 секунды и останавливается. Прогноз записан.

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Колесо не крутится	Настроены ли контакты как выходы? Включено ли питание моторов? Нажми Reset.
Колесо крутится назад	Запиши это в журнал — на следующем шаге исправим направлением.
Крутится правое колесо	Провода моторов перепутаны на шилде. Выключи питание и поменяй местами провода в клеммах M1 и M2.

3 Включи оба мотора

15 мин

Правый мотор устроен так же, как левый: контакт направления `4` и контакт скорости `5`. Имена всех четырёх контактов в шаблоне уже есть.

1. Создай скетч `L21_Both_Motors` по шаблону и напиши `setup()` сам. Свою программу левого мотора используй как образец.

Шаблон: оба мотора L21_Both_Motors_Gaps.ino

```
// Шаблон среднего уровня. Напиши код в разделе «Твоя часть» по целям в комме

const int LEFT_DIR_PIN = 7;
const int LEFT_SPEED_PIN = 6;
const int RIGHT_DIR_PIN = 4;
const int RIGHT_SPEED_PIN = 5;
const int FORWARD = HIGH;
int speed = 100;

// ===== Твоя часть: напиши сам =====

void setup() {
  // Подготовь контакты обоих моторов.
  // Оба колеса крутятся вперёд со скоростью speed одну секунду и останавлива
}

void loop() {
}
```

2. Загрузи и запусти на подставке.
3. Поменяй `speed` на 60, потом на 200. Как меняется вращение?

ПРЕДСКАЖИ

Ты включаешь колёса по очереди: сначала левое, потом правое. Заметно ли, что они трогаются не одновременно? Почему?

Подсказка: Колёса ведут себя по-разному

Сравни команды для левого и правого мотора строка за строкой: для каждого

нужны настройка, направление, сила и остановка.

ДОЛЖНО ПОЛУЧИТЬСЯ

Оба колеса крутятся вперёд одну секунду и останавливаются. С большим `speed` колёса крутятся быстрее, прогноз записан.

Как это устроено: Почему 0–255, а не километры в час

Число в `analogWrite` — это не скорость, а доля времени, когда мотор включён. При 255 он включён всё время, при 100 — примерно 40 % времени. Настоящая скорость зависит ещё от батареи и пола. Поэтому скорость подбирают опытом.

4 Найди ошибку остановки

10 мин

НАЙДИ ОШИБКУ · РОБОТ ПОЧТИ НЕ ОСТАНАВЛИВАЕТСЯ

Соседи перенесли движение в `loop()` и добавили остановку:

```
void loop() {
  analogWrite(LEFT_SPEED_PIN, speed);
  analogWrite(RIGHT_SPEED_PIN, speed);
  delay(1000);
  analogWrite(LEFT_SPEED_PIN, 0);
  analogWrite(RIGHT_SPEED_PIN, 0);
}
```

Они ждали: секунду едет, потом стоит. На подставке колёса крутятся почти без остановки. Почему?

1. Запиши ответ в журнал.
2. Предложи **два** разных исправления и запиши их кодом: одно — без `loop()`, второе — оставив движение в `loop()`.

Подсказка: Второе исправление

Можно ли после остановки «застрять» на месте, чтобы `loop()` больше не повторялся? Вспомни защиту из урока 1.1… или пустой бесконечный цикл `while (true) { }`.

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить, почему остановка длится мгновение, и в журнале два исправления.

2

ЗАНЯТИЕ 2 · 45 МИН

Функции движения

5 Оформи движения функциями

15 мин

Писать каждый раз по четыре строки неудобно. Соберём их в **функции**: `forward()`, `backward()`, `stopMotors()`. Функции `drive` и `forward` в шаблоне — образец.

1. Создай скетч `L21_Moves` по шаблону. Напиши `backward`, `stopMotors` и движение в `setup()` сам.

Шаблон: функции движения L21_Moves_Gaps.ino

```
// Шаблон среднего уровня. Напиши код в разделе «Твоя часть» по целям в комме
// Функции drive и forward ниже — образец.

const int LEFT_DIR_PIN = 7;
const int LEFT_SPEED_PIN = 6;
const int RIGHT_DIR_PIN = 4;
const int RIGHT_SPEED_PIN = 5;
const int FORWARD = HIGH;
const int BACKWARD = (FORWARD == HIGH) ? LOW : HIGH;
int speed = 100;

// ===== Твоя часть: напиши сам =====

void backward(int ms) {
  // Оба колеса крутятся назад со скоростью speed ms миллисекунд.
}

void stopMotors() {
  // Оба мотора стоят.
}

void setup() {
  pinMode(LEFT_DIR_PIN, OUTPUT);
  pinMode(LEFT_SPEED_PIN, OUTPUT);
  pinMode(RIGHT_DIR_PIN, OUTPUT);
  pinMode(RIGHT_SPEED_PIN, OUTPUT);
  stopMotors();
  delay(2000); // время убрать руки от робота

  // Вперёд 1 секунду, остановка, пауза полсекунды, назад 1 секунду, остановк
}

// ===== Образец — не меняй =====

void drive(int leftDir, int leftSpeed, int rightDir, int rightSpeed) {
  digitalWrite(LEFT_DIR_PIN, leftDir);
  digitalWrite(RIGHT_DIR_PIN, rightDir);
  analogWrite(LEFT_SPEED_PIN, leftSpeed);
  analogWrite(RIGHT_SPEED_PIN, rightSpeed);
}

```

```
42 void forward(int ms) {
43   drive(FORWARD, speed, FORWARD, speed);
44   delay(ms);
45 }
46
47 void loop() {
48 }
```

2. Найди, где функции **описаны** и где они **вызываются**. Покажи напарнику.

Подсказка: Функция назад

Чем «назад» отличается от «вперёд»? В программе уже есть константа для этого направления.

Подсказка: Как вызвать функцию

Вызов — имя функции и значение параметра в скобках, в конце `;`.

Как это устроено: Как устроена функция с параметром

`void forward(int ms)` — функция с именем `forward` и параметром `ms`. Когда пишем `forward(1000)`, внутри функции `ms` равно 1000. Так одна функция умеет ехать любое время.

ДОЛЖНО ПОЛУЧИТЬСЯ

Программа проверяется без ошибок. Ты можешь показать описание и вызов `backward`.

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.

1. Загрузи программу на подставке и проверь: вперёд, пауза, назад, стоп.
2. Поставь робота на пол, отметь старт скотчем, включи питание моторов и нажми Reset.
3. Повтори три раза. Каждый раз отметь, где робот остановился, и измерь линейкой, на сколько сантиметров он ушёл от старта вперёд-назад и вбок.

Заезд	Вперёд или назад, см	Вбок, см	Куда уводит
1			
2			
3			

4. Одинаково ли робот ошибается в трёх заездах? Запиши вывод.

> ✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**
> Таблица из трёх заездов и вывод: куда и насколько уводит твоего робота.

🔧 ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Робот едет не прямо	Моторы не одинаковые — это нормально. Запиши, куда уводит.
Робот поехал сразу после загрузки	Так и есть: программа стартует сама. Поэтому в `setup()` есть пауза 2 секунды.

7 Измени программу

10 мин

1. Сделай так, чтобы робот проехал вперёд в два раза дальше, а назад — как раньше.
2. Сделай «челнок» — вперёд, стоп, назад, стоп три раза подряд, — написав вызовы `forward`, `stopMotors`, `backward` только один раз. Нужен цикл `for` из урока 1.1.

ПРЕДСКАЖИ

Вернётся ли робот после трёх челноков точно на старт? Запиши прогноз, потом отметь старт и финиш на полу.

Подсказка: Как устроен for

`for (int i = 0; i < 3; i++) { … }` повторяет команды в скобках три раза.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот выполняет оба задания. В журнале — прогноз и насколько робот ушёл от старта после трёх челноков.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу объяснить, что делает контакт направления и что — контакт скорости.
- [ ] Я могу написать вызов функции `forward` на нужное время.
- [ ] Я знаю, почему без `stopMotors()` робот не остановится.

★ **СРЕДНИЙ УРОВЕНЬ · ЧЕЛНОК БЕЗ КОПИРОВАНИЯ**

Сделай «челнок» из трёх повторов, написав вызовы `forward`, `stopMotors`, `backward` только один раз. Подсказка: цикл `for` из урока 1.1.

Проверь себя

★ Средний уровень

1. Нужно повторить «вперёд — стоп — назад» три раза, написав вызовы функций только один раз. Что подойдёт?

- ○ Поместить вызовы в цикл for из трёх повторов
- ○ Написать функцию loop() три раза
- ○ Добавить delay(3000) после вызовов

2. Скорость уменьшили со 200 до 100, а время езды вперёд осталось 1 секунда. Что изменится?

- ○ За ту же секунду робот проедет меньше
- ○ Робот проедет то же расстояние быстрее
- ○ Робот поедет назад

3. В программе забыли остановить правый мотор. Что сделает робот после секунды движения?

- ○ Правое колесо продолжит крутиться, и робот закружится
- ○ Оба колеса остановятся, потому что левое стоит
- ○ Программа не загрузится

4. Вызвали forward(1000), а сразу за ним — backward(1000) без stopMotors() между ними. Что получится?

- ○ Ошибка при проверке программы
- ○ Робот резко сменит направление без остановки
- ○ Робот постоит секунду и поедет назад

5. После трёх челноков робот оказался в 6 см от старта, а после одного — в 2 см. Почему ошибка растёт?

- ○ Цикл for с каждым разом прибавляет время
- ○ delay становится длиннее при каждом повторе
- ○ Небольшая разница путей вперёд и назад складывается в каждом повторе

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.