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).

Левый мотор — L21_Left_Motor.ino

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

void setup() {
  pinMode(LEFT_DIR_PIN, OUTPUT);
  pinMode(LEFT_SPEED_PIN, OUTPUT);

  digitalWrite(LEFT_DIR_PIN, FORWARD);
  analogWrite(LEFT_SPEED_PIN, 100);
  delay(2000);
  analogWrite(LEFT_SPEED_PIN, 0);
}

void loop() {
}
```

ПРЕДСКАЖИ

Что сделает колесо после загрузки? Сколько секунд оно будет крутиться? Остановится ли оно само? Скажи напарнику до запуска.

1. Набери программу и загрузи её при выключенном питании моторов.
2. Включи питание моторов и нажми кнопку Reset на плате: программа начнётся заново.
3. Сравни, что произошло, со своим прогнозом.

ДОЛЖНО ПОЛУЧИТЬСЯ

Левое колесо крутится вперёд 2 секунды и останавливается.

Как это устроено: Что значит каждая строка

- `const int LEFT_DIR_PIN = …;` — имя для номера контакта. Дальше в программе пишем имя, а не число: так понятнее и меньше ошибок.
- `digitalWrite(LEFT_DIR_PIN, FORWARD);` — выбрать направление «вперёд».
- `analogWrite(LEFT_SPEED_PIN, 100);` — включить мотор с силой 100 из 255.
- `analogWrite(LEFT_SPEED_PIN, 0);` — остановить мотор.
- Всё написано в `setup()`, поэтому выполняется один раз. `loop()` пустой: после остановки робот просто стоит.

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Колесо не крутится	Включено ли питание моторов? Нажми Reset после включения.
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

1. Создай скетч `L21_Both_Motors` и набери программу.

Оба мотора — L21_Both_Motors.ino

```
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;

void setup() {
  pinMode(LEFT_DIR_PIN, OUTPUT);
  pinMode(LEFT_SPEED_PIN, OUTPUT);
  pinMode(RIGHT_DIR_PIN, OUTPUT);
  pinMode(RIGHT_SPEED_PIN, OUTPUT);

  digitalWrite(LEFT_DIR_PIN, FORWARD);
  digitalWrite(RIGHT_DIR_PIN, FORWARD);
  analogWrite(LEFT_SPEED_PIN, speed);
  analogWrite(RIGHT_SPEED_PIN, speed);
  delay(1000);

  analogWrite(LEFT_SPEED_PIN, 0);
  analogWrite(RIGHT_SPEED_PIN, 0);
}

void loop() {
}
```

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

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

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

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 НЕ ОСТАНАВЛИВАЕТСЯ

Соседи перенесли команды движения в `loop()` и убрали остановку:

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

Они ждали, что робот проедет одну секунду. Что будет на самом деле и почему? Как исправить?

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить: `delay` только ждёт, моторы сами не выключаются, а `loop()` повторяет команды снова и снова.

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()`. Внутри них — знакомые команды.

1. Создай скетч `L21_Moves` и набери программу.

Функции движения L21_Moves.ino

```
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;

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);
}

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

void backward(int ms) {
  drive(BACKWARD, speed, BACKWARD, speed);
  delay(ms);
}

void stopMotors() {
  analogWrite(LEFT_SPEED_PIN, 0);
  analogWrite(RIGHT_SPEED_PIN, 0);
}

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

  forward(1000);
  stopMotors();
```

```
  delay(500);
  backward(1000);
  stopMotors();
}

void loop() {
}
```

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

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.

`drive(…)` — вспомогательная функция: она сразу задаёт направление и скорость обоих колёс. Ей пользуются остальные функции.

ДОЛЖНО ПОЛУЧИТЬСЯ

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

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, включи питание моторов и нажми Reset.
3. Отметь на полу мелом или скотчем точку старта и точку, где робот остановился.

> ✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**
> Робот проезжает вперёд, останавливается и возвращается примерно на место старта.

🔧 ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Робот едет не прямо	Моторы не одинаковые — это нормально. Запиши, куда уводит.
Робот вернулся не на старт	Вперёд и назад колёса крутятся чуть по-разному. Запиши разницу.
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 — назад три раза подряд с паузами.

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.

ОСНОВНОЕ · ВПЕРЁД, СТОП, НАЗАД

Робот проезжает вперёд 1 секунду, останавливается и возвращается назад.

Проверь себя

● Основное

1. Что задаёт число в команде analogWrite(LEFT_SPEED_PIN, 100)?

- ○ Скорость в сантиметрах в секунду
- ○ Насколько сильно включён мотор: от 0 до 255
- ○ Сколько секунд крутится мотор

2. Как сделать, чтобы левое колесо крутилось назад?

- ○ Записать в analogWrite отрицательное число
- ○ Поменять значение на контакте направления
- ○ Убрать delay

3. Что сделает робот?

```
1 forward(1000);
2 delay(500);
```

- ○ Проедет 1 секунду и постоит полсекунды
- ○ Проедет полторы секунды и продолжит ехать: команды остановки нет
- ○ Не поедет

4. Зачем описывать движения функциями forward(), backward(), stopMotors()?

- ○ Без функций моторы не работают
- ○ Программа короче, её легче читать и исправлять в одном месте
- ○ Робот едет быстрее

5. Почему в setup() перед движением стоит delay(2000)?

- ○ Чтобы успеть убрать руки и поставить робота
- ○ Чтобы моторы прогрелись
- ○ Без паузы программа не загрузится

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.