Тема 3 · Урок 3.1

Датчики линии и первое следование

Робот «смотрит» на пол, отличает чёрное от белого и сам держится линии.

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

РЕЗУЛЬТАТ УРОКА

Робот проезжает по чёрной линии с плавным изгибом, поворачивая по сигналам двух датчиков. В бортовом журнале записаны показания датчиков на белом и чёрном.

Ты научишься

- Прочитать показания датчика командой analogRead и вывести их в монитор порта.
- Различать строки Serial Monitor и каналы телеметрии с временем и единицами; сравнивать сохранённые показания модели и настоящего робота.
- Измерить показания на белом и чёрном и объяснить, почему у каждого датчика они свои.
- Записать правило «показание ближе к чёрному — значит, под датчиком линия».
- Составить два правила поворота и проверить их сначала на подставке, затем на дорожке.

Нужно знать

- 1.1 Собираем робота и оживляем плату
- 2.1 Моторы: едем и останавливаемся

Понадобится

Датчик линии (аналоговый, Trema) × 2

USB-кабель для Arduino

Зачем этот урок

До сих пор робот делал только то, что ты заранее записал: ехать столько-то, повернуть так-то. Сегодня он начнёт сам принимать решения. Два датчика будут смотреть на пол, и робот будет поворачивать, когда линия уходит в сторону.

РАЗМИНКА · ВСПОМНИ

1. Какая команда задаёт скорость мотора и какие числа она принимает?
2. Что делает `delay(500)`?
3. Какое колесо надо остановить, чтобы робот повернул налево?

ПРЕДСКАЖИ

Как робот может «увидеть» чёрную линию, если у него нет камеры? Обсудите с напарником и запишите догадку в бортовой журнал. В конце урока проверите.

БЕЗОПАСНОСТЬ

Пока меняешь провода, USB-кабель отключён, питание моторов выключено. Опыты с моторами — сначала на подставке: колёса не касаются стола.

ЗАНЯТИЕ 1 · 45 МИН

1 Датчики и их показания

1 Подготовь дорожку и робота

5 мин

Что должен делать робот. Чтобы ехать по дорожке, роботу нужно замечать, когда линия оказалась под левым или правым датчиком. Для этого используем два направленных вниз датчика отражения; они смотрят на пол, а не ищут предметы впереди.

Что используем и как это работает. В датчике есть излучатель и приёмник отражённого света. Поверхности отражают свет по-разному, а приёмник превращает отражённый свет в электрический сигнал. analogRead считывает его как число. Это не готовый ответ «чёрное»: сначала измеряем белое и чёрное для каждого датчика при рабочей высоте.

Как связать измерение с действием. Программа сравнивает чтение с измеренным порогом и выбирает действие колёс. Над линией слева и справа должны получиться разные команды коррекции. Проверим четыре сочетания карточек на подставке, затем движение по линии. Одинаковые показания обоих датчиков ещё не доказывают, что робот стоит правильно: оба могут оказаться и на белом, и на чёрном.

Подготовь дорожку для своего уровня и проверь, подходит ли она твоему роботу.

1. Сделай дорожку: прямая, плавный изгиб, крутая дуга и перекрёсток. Подготовь белую и чёрную карточки.
2. Измерь линейкой ширину линии и расстояние между датчиками (между серединами окошек).
3. Поставь робота на линию. Сколько миллиметров остаётся от края линии до каждого датчика? Запиши в журнал: что будет, если линия шире расстояния между датчиками?

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Дорожка готова, в журнале — ширина линии, расстояние между датчиками, запас с каждой стороны и ответ на вопрос.

2 Посмотри, что видит датчик

12 мин

Датчик светит на пол и меряет, сколько света вернулось. Левый датчик подключён к `A4`, правый — к `A5`.

Как это устроено: Справка по командам

Команда	Что делает
`analogRead(pin)`	Показание датчика: число от 0 до 1023
`Serial.begin(9600)`	Включает связь с компьютером, один раз в `setup()`
`Serial.print(x)` / `Serial.println(x)`	Печатает в монитор порта; `println` — с переходом строки
`millis()`	Сколько миллисекунд прошло с включения платы

Требование. Скетч `L31_Read_Sensors` пять раз в секунду печатает строку `t=… L=… R=…`, где `t` — время в миллисекундах с включения.

Приёмка.

1. По числам `t` в мониторе проверь, что строки идут примерно каждые 200 мс. На сколько миллисекунд отличается промежуток от 200 и почему?
2. Поднеси под левый датчик белую, затем чёрную карточку на рабочей высоте. Запиши, на сколько меняется число.

Подсказка: Почему промежуток не ровно 200

Сколько времени занимают сами чтение и печать, кроме `delay(200)`?

ДОЛЖНО ПОЛУЧИТЬСЯ

Программа выполняет требование. В журнале — реальный промежуток между строками с объяснением и изменение числа на белом и чёрном.

3 Проверь оба датчика

8 мин

Сними **профиль** датчиков: как меняются числа, пока линия проходит под роботом поперёк.

1. Положи робота поперёк линии так, чтобы линия была далеко справа от правого датчика.
2. Сдвигай робота влево шагами по 5 мм, пока линия не уйдёт за левый датчик. На каждом шаге записывай L и R.

Сдвиг, мм	L	R
0		
5		
…		

3. Нарисуй в журнале два графика: L и R от сдвига. Найди по графику, где линия ровно между датчиками и на сколько миллиметров можно уйти в сторону, пока датчик её не заметит.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Таблица и два графика. В журнале — ширина «слепой зоны» между датчиками в миллиметрах.

4 Измерь белое и чёрное

12 мин

Откалибруй датчики и докажи, что калибровке можно доверять.

1. Для каждого датчика запиши по 10 показаний на белом и 10 на чёрном. Найди наименьшее и наибольшее в каждой серии — это **разброс**.
2. Сравни разброс с разницей белого и чёрного. Во сколько раз разница больше разброса?
3. Повтори белое у окна и в тёмном углу класса. Меняются ли числа сильнее разброса?
4. Запиши вывод: где можно доверять калибровке и когда её нужно повторить.

Датчик	Белое: мин–макс	Чёрное: мин–макс	Разница типичных
Левый			
Правый			

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Таблица разбросов, сравнение с разницей и вывод, когда калибровку повторять.

2

ЗАНЯТИЕ 2 · 45 МИН

Робот едет по линии

5 Научи программу замечать линию

10 мин

Требование 1. Функция `onLine(value, white, black)` отвечает `true`, когда под датчиком линия:

- показание ближе к чёрному измерению, чем к белому;
- работает и когда у датчика чёрное — большое число, и когда маленькое;
- внутри нет готовых чисел-порогов: только `value`, `white` и `black`.

ПРЕДСКАЖИ

Почему нельзя просто написать `value > 500`? Назови два случая, когда такое правило ошибётся.

Приёмка. Напиши `onLine` в тестовом скетче, загрузи его и открой монитор порта (9600). Тест проверяет восемь случаев для двух видов датчиков, в том числе показания за пределами твоих измерений.

Тест функции onLine L31_OnLine_Test.ino

```
// Проверка функции onLine. Напиши функцию здесь, загрузи и открой монитор по
// Когда все строки PASS, перенеси onLine в свою программу.

// Требование 1: true, если показание ближе к чёрному, чем к белому.
bool onLine(int value, int white, int black) {
  return false;
}

int failed = 0;

void expect(int value, int white, int black, bool expected) {
  bool actual = onLine(value, white, black);
  Serial.print(actual == expected ? "PASS " : "FAIL ");
  Serial.print("onLine(");
  Serial.print(value);
  Serial.print(", ");
  Serial.print(white);
  Serial.print(", ");
  Serial.print(black);
  Serial.print(") = ");
  Serial.print(actual ? "true" : "false");
  Serial.print(", ожидаем ");
  Serial.println(expected ? "true" : "false");
  if (actual != expected) {
    failed++;
  }
}

void setup() {
  Serial.begin(9600);

  // Датчик, у которого чёрное — большое число, а белое — маленькое.
  expect(780, 120, 780, true);
  expect(120, 120, 780, false);
  expect(600, 120, 780, true);
  expect(300, 120, 780, false);
  // За пределами измерений.
  expect(900, 120, 780, true);
  expect(50, 120, 780, false);
  // Датчик, у которого наоборот: белое — большое число, чёрное — маленькое.
  expect(300, 900, 120, true);
```

```
  expect(700, 900, 120, false);

  if (failed == 0) {
    Serial.println("Все проверки пройдены");
  } else {
    Serial.print("Ошибок: ");
    Serial.println(failed);
  }
}

void loop() {
}
```

Когда все строки `PASS`, добавь `onLine` в свою программу из шага «Посмотри, что видит датчик» и сохрани её как `L31_Line_Detect`. Пусть она печатает «слева линия» и «справа линия». Впиши свои измерения и проверь карточкой.

Подсказка: Идея (1 из 3)

Нужны два расстояния: от показания до чёрного и от показания до белого. Какое меньше?

Подсказка: План (2 из 3)

1. Расстояние до чёрного — разница без знака.
2. Расстояние до белого — так же.
3. Вернуть результат сравнения двух расстояний.

Подсказка: Фрагмент кода (3 из 3)

`abs(value - black)` — расстояние до чёрного. Зачем здесь `abs`? Что будет без него у датчика, где чёрное — маленькое число?

ДОЛЖНО ПОЛУЧИТЬСЯ

Тест показывает «Все проверки пройдены». Твоя программа печатает «слева линия» и «справа линия», только когда под датчиком чёрное.

6 Придумай, как поворачивать

7 мин

Робот едет вперёд, и линия должна оставаться между датчиками. **Составь правила сам**: что делают колёса в каждом из четырёх случаев и почему.

Что видят датчики	Левое колесо	Правое колесо	Куда едет робот	Почему
Оба белое				
Левый — линия				
Правый — линия				
Оба — линия				

1. Проверь каждую строку на выключенном роботе: сдвинь его рукой и покажи, куда он должен поехать.
2. «Оба белое» бывает в двух разных положениях робота. В каких? Может ли программа их различить? Запиши, что тогда случится с роботом.
3. На прямой дорожке «оба — линия» почти не бывает. Где на итоговом поле это встретится и что тогда должен делать робот?

Подсказка: Два случая «оба белое»

Подумай, где может быть линия, когда ни один датчик её не видит.

ДОЛЖНО ПОЛУЧИТЬСЯ

Таблица из четырёх строк с объяснением. В журнале записано, в каком положении программа не знает, где линия.

7 Запиши правила в программу

10 мин

Возьми каркас: в нём контакты, моторы и функция `drive` из урока 2.1. Перенеси в него свои измерения и функцию `onLine` и сохрани как `L31_First_Line`. Эта программа понадобится тебе в уроке 3.2.

Каркас движения по линии L31_First_Line_Skeleton.ino

```
// Каркас движения по линии. Требования к программе — в тексте урока.
const int LEFT_SENSOR_PIN = A4;
const int RIGHT_SENSOR_PIN = A5;

// Впиши свои измерения из бортового журнала.
const int LEFT_WHITE = 0;
const int LEFT_BLACK = 0;
const int RIGHT_WHITE = 0;
const int RIGHT_BLACK = 0;

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;

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);
}

// Требование 1: перенеси функцию onLine из тестового скетча.
bool onLine(int value, int white, int black) {
  return false;
}

void setup() {
  pinMode(LEFT_DIR_PIN, OUTPUT);
  pinMode(LEFT_SPEED_PIN, OUTPUT);
  pinMode(RIGHT_DIR_PIN, OUTPUT);
  pinMode(RIGHT_SPEED_PIN, OUTPUT);
  drive(FORWARD, 0, FORWARD, 0);

  // Требование 3: защита от запуска без измерений и время убрать руки.
}

void loop() {
  // Требование 2: прочитать датчики, решить по своей таблице правил, задать
  drive(FORWARD, 0, FORWARD, 0);
```

```
42 }
```

Требование 2. Колёса выполняют твою таблицу правил:

- в каждом повторе программа заново читает оба датчика и решает заново: колесо, остановленное один раз, снова едет, когда его датчик видит белое;
- решение принимает твоя функция `onLine`;
- скорость — одна именованная переменная `speed` в начале скетча.

Требование 3. Робот безопасен:

- без измерений белого и чёрного колёса стоят;
- после включения есть 2 секунды, чтобы убрать руки.

Приёмка — на подставке в следующем шаге: робот выполняет каждую строку твоей таблицы правил, а без измерений колёса не крутятся.

ПРЕДСКАЖИ

Что сделает робот, если скорости обоих колёс задать один раз в `setup()`? Запиши ответ до проверки.

Подсказка: Идея (1 из 3)

Каждый повтор: прочитать датчики → решить → отправить скорости на моторы. Решение начинается с «оба едут», а правила его меняют.

Подсказка: План (2 из 3)

1. Прочитать оба датчика.
2. Спросить `onLine` для каждого.
3. Обе скорости — `speed`.
4. Правила из таблицы меняют скорости.
5. Отправить скорости в `drive`.

Подсказка: Фрагмент кода (3 из 3)

`if (onLine(leftValue, LEFT_WHITE, LEFT_BLACK)) { leftSpeed = 0; }` — а для правого датчика? А строка «оба — линия» из твоей таблицы?

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Программа выполняет требования 2 и 3, **Проверить** проходит без ошибок.

8 Проверь на подставке

6 мин

⚠ **БЕЗОПАСНОСТЬ**

Загружай программу при выключенном питании моторов. Колёса не касаются стола.

Приёмка требований 2–3 на подставке.

1. Загрузи программу с нулями вместо измерений. Колёса не должны крутиться. Впиши измерения и загрузи снова.
2. Проверь каждую строку своей таблицы правил карточками.
3. Засеки, через сколько секунд после включения трогаются колёса.

Проверка	Ждал	Увидел
Без измерений		
Оба белое		
Левый — линия		
Правый — линия		
Оба — линия		
Пауза на старте		

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Все проверки совпали с требованиями. Если нет — исправлено и проверено снова.

9 Первый заезд по линии

8 мин

1. Выключи питание моторов, отключи USB-кабель. Будь готов выключить питание, если робот уходит с дорожки.
2. Найди наибольшую `speed`, при которой робот проходит плавный изгиб **три раза из трёх**. Начни с 80 и поднимай шагами по 20.
3. На наибольшей скорости, которую робот не выдержал, посмотри, где он теряет линию. Объясни в журнале причину через твою таблицу правил.
4. Запиши, что нужно изменить в правилах, чтобы робот держал линию быстрее. Проверишь это в уроке 3.2.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Наибольшая надёжная скорость, место и причина схода, идея улучшения правил.

★★ **ПРОДВИНУТЫЙ УРОВЕНЬ · НЕ СТОП, А ЗАМЕДЛЕНИЕ**

Вместо остановки колеса замедляй его: например, до трети от `speed`. Сравни на одном изгибе два варианта — «стоп» и «замедление». Какой плавнее? При какой скорости каждый вариант ещё держит линию? Запиши результаты в журнал: это первый шаг к П-регулятору, который понадобится на крутых дугах продвинутого поля.

Проверь себя

★★ Продвинутый уровень

1. **speed = 150. По заданию ★★ колесо не останавливается, а замедляется до трети скорости. Какое значение получит колесо?**

- ○ 50
- ○ 0
- ○ 100

2. **Чем замедление колеса лучше остановки?**

- ○ Робот поворачивает плавнее и меньше дёргается
- ○ Робот больше никогда не теряет линию
- ○ Батарея садится вдвое медленнее

3. **Тест onLine не проходит для датчика «наоборот» (white = 900, black = 120). Почему?**

```
bool onLine(int value, int white, int black) {
  return value > (white + black) / 2;
}
```

- ○ bool не может хранить результат сравнения
- ○ Функция считает линией большое число, а у этого датчика чёрное — маленькое
- ○ Деление на 2 даёт дробное число

4. Оба датчика видят белое. В каких двух положениях может быть робот?

- ○ Робот на перекрёстке или на дуге
- ○ Только линия ровно между датчиками
- ○ Линия между датчиками или линия потеряна за обоими датчиками

5. Что будет делать робот с этой программой?

```
void setup() {
  // … настройка моторов
  leftSpeed = speed;
  rightSpeed = speed;
}

void loop() {
  readSensors();
  if (onLine(leftValue, LEFT_WHITE, LEFT_BLACK)) {
    leftSpeed = 0;
  }
  drive(FORWARD, leftSpeed, FORWARD, rightSpeed);
}
```

- ○ Колесо, остановленное один раз, так и стоит — робот кружит вокруг него
- ○ Робот поедет прямо и не будет поворачивать
- ○ Программа не пройдёт проверку

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 научился сам выбирать, какое колесо остановить, по показаниям двух датчиков. Мы не угадывали числа, а измерили их на своём поле. Рывки при движении — цена простого правила «едет или стоит». В уроке 3.2 робот научится поворачивать плавно: тем сильнее, чем дальше линия ушла в сторону.

Вернись к своей догадке из начала урока. Подтвердилась ли она?

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

датчик линии	Датчик, который светит на пол и измеряет, сколько света отразилось. Белое отражает много, чёрное — мало.
показание	Число, которое программа получает от датчика. У нашей платы это число от 0 до 1023.
монитор порта	Панель Serial Monitor в MyRobotica, где видны сообщения, которые плата отправляет на компьютер.
калибровка	Настройка программы по измерениям своего робота и своего поля.
переменная	Именованное место в памяти, где программа хранит число.
условие	Проверка if: команды в фигурных скобках выполняются, только если проверка верна.
телеметрия	Передача именованных измерений с временем и единицами, чтобы строить графики и сохранять наблюдения.
АЦП	Аналого-цифровое преобразование: напряжение на входе платы превращается в число от 0 до 1023.