Тема 4 · Урок 4.2

Повороты на перекрёстке и маршрут

Робот сворачивает с перекрёстка на нужную линию и проходит маршрут из нескольких решений.

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

РЕЗУЛЬТАТ УРОКА

Робот поворачивает с перекрёстка налево или направо до новой линии, разворачивается на линии и проходит маршрут из нескольких решений, записанных в программе по порядку.

Ты научишься

- Объяснить, почему поворот по датчикам надёжнее поворота по времени.
- Подобрать выезд на центр перекрёстка, чтобы после поворота робот оказался на линии.
- Записать маршрут последовательностью решений «налево / направо / прямо».
- Проверить маршрут на поле и найти, на каком решении он ломается.

Нужно знать

- 4.1 Находим и считаем перекрёстки
- 2.2 Повороты и развороты

Понадобится

Датчик линии (аналоговый, Trema) × 2

Зачем этот урок

Робот находит перекрёсток. Теперь он должен **свернуть** — к грузу, к месту доставки или дальше по дороге. В уроке 2.2 повороты были по времени и зависели от батареи. Сегодня робот будет поворачивать по датчикам: крутиться, пока не найдёт новую линию.

РАЗМИНКА · ВСПОМНИ

1. Почему поворот по времени получается разным при разряженной батарее?
2. Что делает `passCrossing()`?
3. Как робот понимает, что под ним перекрёсток?

1

ЗАНЯТИЕ 1 · 45 МИН

Поворот по датчикам

1 Как повернуть точно

8 мин

Что должен делать робот. Роботу нужно пройти несколько перекрёстков и выбрать нужное направление на каждом. Используем датчики линии для обнаружения событий и заранее записанный план маршрута для выбора действия.

Что используем и как это работает. Датчики отвечают на вопрос «что сейчас под роботом», а маршрут — «что делать следующим». Программа доезжает до перекрёстка, выполняет поворот или проезд прямо, выходит с поперечной полосы и переходит к следующему пункту. Массив или последовательность вызовов хранит порядок, но не измеряет положение робота.

Как связать измерение с действием. Проверяем маршрут по участкам: отмечаем, какой перекрёсток найден и какое действие выполнено. Если робот не обнаружил полосу или повернул не туда, следующие команды будут выполняться уже в другом месте. Поэтому сначала проверим каждую функцию, затем соберём целый маршрут.

ПРЕДСКАЖИ

Робот стоит на перекрёстке и крутится налево. Как он может сам понять, что повернул достаточно и уже смотрит вдоль новой линии?

1. Поставь робота на центр перекрёстка и медленно поверни руками налево, потом направо.
2. Для каждого поворота запиши: какой датчик первым оказался над новой линией и какой датчик проходит над старой линией в начале поворота.

Поворот	Первым видит новую линию	Сначала проходит над старой
Налево		
Направо		

ДОЛЖНО ПОЛУЧИТЬСЯ

Таблица заполнена. Ты можешь объяснить идею: крутиться, пока датчик со стороны поворота не увидит новую линию.

2 Выезд на центр перекрёстка

12 мин

Датчики стоят **впереди** колёс. Когда датчики заметили перекрёсток, колёса ещё до него не доехали.

ПРЕДСКАЖИ

Робот повернул на месте сразу, как только датчики увидели перекрёсток. Где окажется робот относительно новой линии? Покажи на дорожке руками.

1. Поставь робота датчиками на поперечную линию и поверни его руками на месте на 90°. Где оказались датчики — над новой линией или сбоку?
2. Измерь линейкой, сколько ещё ехать колёсам до центра перекрёстка. Запиши в журнал.
3. Время этого проезда — `centerMs` — подберёшь на следующем шаге. Как ты поймёшь, что время слишком маленькое, а как — что слишком большое? Запиши.

ДОЛЖНО ПОЛУЧИТЬСЯ

Прогноз проверен руками, расстояние записано, признаки «мало» и «много» для `centerMs` записаны.

3 Поворот на перекрёстке

22 мин

1. Создай скетч `L42_Turn_At_Crossing` по шаблону. Перенеси измерения, `kp` и `baseSpeed`. Функции линии и перекрёстков уже готовы.
2. Напиши сам выезд на центр `toCenter()`, поворот до линии `turnUntilLine()` и повороты с перекрёстка налево и направо. `setup()` ниже проверяет их на дорожке.

Шаблон: поворот на перекрёстке L42_Turn_At_Crossing_Gaps.ino

```
// Шаблон среднего уровня. Напиши код в разделе «Твоя часть» по целям в комме
// setup() ниже — проверка твоих функций на дорожке.

const int LEFT_SENSOR_PIN = A4;
const int RIGHT_SENSOR_PIN = A5;
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;

// Измерения и настройки из журнала (уроки 3.1 и 3.2).
const int LEFT_WHITE = 0;
const int LEFT_BLACK = 0;
const int RIGHT_WHITE = 0;
const int RIGHT_BLACK = 0;
float kp = 0.5;
int baseSpeed = 100;
int leftLevel = 100;
int rightLevel = 100;
int turnSpeed = 100;
int centerMs = 250; // подбираем: проехать от датчиков до центра перекрёстка
int leaveMs = 200;  // время, чтобы датчик ушёл со старой линии

// ===== Твоя часть: напиши сам =====

void toCenter() {
  // Колёса встают над центром перекрёстка: проехать вперёд centerMs и остано
}

void turnUntilLine(bool left) {
  // Крутиться на месте (left = true — налево) до новой линии; в конце колёса
  // Пока датчик ещё над старой линией, она не должна считаться новой — для э
}

void turnLeftAtCrossing() {
  // Повернуть с перекрёстка налево на новую линию.
}

void turnRightAtCrossing() {
```

```
  // Повернуть с перекрёстка направо на новую линию.
}

// ===== Готовое из уроков 3.2–4.1 и проверка — не меняй =====

int level(int value, int white, int black) {
  long result = (long)(value - black) * 100 / (white - black);
  return constrain(result, 0, 100);
}

void readSensors() {
  leftLevel = level(analogRead(LEFT_SENSOR_PIN), LEFT_WHITE, LEFT_BLACK);
  rightLevel = level(analogRead(RIGHT_SENSOR_PIN), RIGHT_WHITE, RIGHT_BLACK);
}

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

void followLineStep() {
  int error = leftLevel - rightLevel;
  int correction = kp * error;
  drive(FORWARD, constrain(baseSpeed + correction, 0, 255),
        FORWARD, constrain(baseSpeed - correction, 0, 255));
}

void setupRobot() {
  pinMode(LEFT_DIR_PIN, OUTPUT);
  pinMode(LEFT_SPEED_PIN, OUTPUT);
  pinMode(RIGHT_DIR_PIN, OUTPUT);
  pinMode(RIGHT_SPEED_PIN, OUTPUT);
  stopMotors();
  // Без измерений робот не поедет.
  if (LEFT_WHITE == LEFT_BLACK || RIGHT_WHITE == RIGHT_BLACK) {
    while (true) {
```

```
    }
  }
  delay(2000);
}

// Перекрёсток: оба датчика видят чёрное.
bool onCrossing() {
  return leftLevel < 30 && rightLevel < 30;
}

// Ехать по линии, пока не найдём перекрёсток.
void followLineToCrossing() {
  while (true) {
    readSensors();
    if (onCrossing()) {
      return;
    }
    followLineStep();
  }
}

// Проехать перекрёсток прямо, пока датчики не съедут с чёрного.
void passCrossing() {
  drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
  do {
    readSensors();
  } while (onCrossing());
  delay(100);
}

// Развернуться на линии: крутиться, пока линия снова не окажется под датчико
void turnAroundOnLine() {
  turnUntilLine(true);
}

// Ехать по линии заданное время.
void followLineFor(unsigned long ms) {
  unsigned long start = millis();
  while (millis() - start < ms) {
    readSensors();
    followLineStep();
  }
  stopMotors();
```

```
}

void setup() {
  setupRobot();
  followLineToCrossing();
  turnLeftAtCrossing();
  followLineFor(1500);
  turnAroundOnLine();
  followLineToCrossing();
  stopMotors();
}

void loop() {
}
```

ПРЕДСКАЖИ

Что случится, если начать искать новую линию сразу, не выждав `leaveMs`?

3. Запусти на дорожке с перекрёстком. Подбери `centerMs` по признакам, которые записал в прошлом шаге.

Подсказка: Крутиться до линии

Это похоже на `passCrossing()` из урока 4.1: повторять чтение датчиков, пока условие верно. Какое условие значит «линии под датчиком ещё нет»?

Подсказка: Какой датчик смотреть

Поворачиваем налево — новую линию первым увидит левый датчик.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот доезжает до перекрёстка, поворачивает налево и едет по новой линии. Прогноз записан.

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
После поворота робот сбоку от линии	Подбери `centerMs`
Робот крутится бесконечно	Датчик не видит линию: проверь условие поиска и измерения
Робот останавливается сразу	Увеличь `leaveMs`

ЗАНЯТИЕ 2 · 45 МИН

2 Маршрут из решений

4 Разворот на линии

10 мин

В конце ответвления роботу нужно развернуться и вернуться к перекрёстку. Разворот — тот же поворот по датчикам, только линия теперь сзади.

1. Найди в скетче функцию `turnAroundOnLine()`. Почему в ней нет выезда на центр, как в повороте на перекрёстке? Запиши ответ.
2. Проверь на дорожке три раза: проехать по ответвлению, развернуться, вернуться к перекрёстку.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот три раза разворачивается и возвращается к перекрёстку. Ответ про выезд на центр записан.

5 Маршрут по порядку

22 мин

Маршрут — это решения на перекрёстках по порядку. Робот не видит, какой это перекрёсток: он просто выполняет следующее решение.

1. Начерти в журнале тренировочную дорожку и путь: налево, прямо, направо.
2. Создай скетч `L42_Route` по шаблону и напиши маршрут в `setup()` сам из готовых функций урока.

Шаблон: маршрут L42_Route_Gaps.ino

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

const int LEFT_SENSOR_PIN = A4;
const int RIGHT_SENSOR_PIN = A5;
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;

// Измерения и настройки из журнала (уроки 3.1 и 3.2).
const int LEFT_WHITE = 0;
const int LEFT_BLACK = 0;
const int RIGHT_WHITE = 0;
const int RIGHT_BLACK = 0;
float kp = 0.5;
int baseSpeed = 100;
int leftLevel = 100;
int rightLevel = 100;
int turnSpeed = 100;
int centerMs = 250; // подбираем: проехать от датчиков до центра перекрёстка
int leaveMs = 200;  // время, чтобы датчик ушёл со старой линии

// ===== Твоя часть: напиши сам =====

void setup() {
  setupRobot();

  // Маршрут из журнала: на каждом перекрёстке — своё решение по порядку.
  // В конце остановиться на следующем перекрёстке.
}

// ===== Готовое из уроков 3.2–4.2 — не меняй =====

int level(int value, int white, int black) {
  long result = (long)(value - black) * 100 / (white - black);
  return constrain(result, 0, 100);
}

void readSensors() {
```

```
  leftLevel = level(analogRead(LEFT_SENSOR_PIN), LEFT_WHITE, LEFT_BLACK);
  rightLevel = level(analogRead(RIGHT_SENSOR_PIN), RIGHT_WHITE, RIGHT_BLACK);
}

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

void followLineStep() {
  int error = leftLevel - rightLevel;
  int correction = kp * error;
  drive(FORWARD, constrain(baseSpeed + correction, 0, 255),
        FORWARD, constrain(baseSpeed - correction, 0, 255));
}

void setupRobot() {
  pinMode(LEFT_DIR_PIN, OUTPUT);
  pinMode(LEFT_SPEED_PIN, OUTPUT);
  pinMode(RIGHT_DIR_PIN, OUTPUT);
  pinMode(RIGHT_SPEED_PIN, OUTPUT);
  stopMotors();
  // Без измерений робот не поедет.
  if (LEFT_WHITE == LEFT_BLACK || RIGHT_WHITE == RIGHT_BLACK) {
    while (true) {
    }
  }
  delay(2000);
}

// Перекрёсток: оба датчика видят чёрное.
bool onCrossing() {
  return leftLevel < 30 && rightLevel < 30;
}

// Ехать по линии, пока не найдём перекрёсток.
```

```
void followLineToCrossing() {
  while (true) {
    readSensors();
    if (onCrossing()) {
      return;
    }
    followLineStep();
  }
}

// Проехать перекрёсток прямо, пока датчики не съедут с чёрного.
void passCrossing() {
  drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
  do {
    readSensors();
  } while (onCrossing());
  delay(100);
}

// Выехать на центр перекрёстка: датчики стоят впереди колёс.
void toCenter() {
  drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
  delay(centerMs);
  stopMotors();
}

// Крутиться на месте, пока датчик со стороны поворота не найдёт новую линию.
void turnUntilLine(bool left) {
  if (left) {
    drive(BACKWARD, turnSpeed, FORWARD, turnSpeed);
  } else {
    drive(FORWARD, turnSpeed, BACKWARD, turnSpeed);
  }
  delay(leaveMs);
  do {
    readSensors();
  } while ((left ? leftLevel : rightLevel) > 30);
  stopMotors();
}

void turnLeftAtCrossing() {
  toCenter();
  turnUntilLine(true);
```

```
}

void turnRightAtCrossing() {
  toCenter();
  turnUntilLine(false);
}

// Развернуться на линии: крутиться, пока линия снова не окажется под датчико
void turnAroundOnLine() {
  turnUntilLine(true);
}

// Ехать по линии заданное время.
void followLineFor(unsigned long ms) {
  unsigned long start = millis();
  while (millis() - start < ms) {
    readSensors();
    followLineStep();
  }
  stopMotors();
}

void loop() {
}
```

ПРЕДСКАЖИ

Нарисуй, где робот остановится на дорожке.

3. Запусти. Если маршрут сломался — на каком решении? Почему?

Подсказка: Одно решение

Перед каждым решением робот должен доехать до перекрёстка.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот проходит маршрут из трёх решений и останавливается там, где ты предсказал.

НАЙДИ ОШИБКУ · РОБОТ ПОВОРАЧИВАЕТ НЕ НА ТОМ ПЕРЕКРЁСТКЕ

У соседей после `turnLeftAtCrossing()` сразу стоит `turnRightAtCrossing()`, без `followLineToCrossing()` между ними. Что сделает робот?

6 Задания по уровням

13 мин

★ **СРЕДНИЙ УРОВЕНЬ · ЧЕТЫРЕ РЕШЕНИЯ**

Составь на тренировочном поле маршрут из четырёх решений, в котором есть налево, направо и прямо. Запиши его в журнал, затем в программу, и пройди без ошибок.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу объяснить, как робот понимает, что поворот закончен.
- [ ] Я могу записать маршрут решениями по порядку.
- [ ] Я нахожу, на каком решении сломался маршрут.

Проверь себя

★ Средний уровень

1. Маршрут: налево, прямо, направо, налево. Сколько решений на перекрёстках выполнит робот?

- ○ 4
- ○ 3
- ○ 2

2. Зачем маршрут сначала записывают в журнал, а потом в программу?

- ○ Так проще сверить маршрут со схемой поля до запуска
- ○ Плата читает журнал
- ○ Без журнала программа не соберётся

3. После поворота робот оказался сбоку от новой линии — со стороны, откуда приехал. Что поменять?

- ○ Уменьшить centerMs
- ○ Увеличить leaveMs
- ○ Увеличить centerMs: колёса не доехали до центра перекрёстка

4. Для какого поворота подходит этот поиск линии?

```
do {
  readSensors();
} while (leftLevel > 30);
```

- ○ Налево: робот крутится, пока левый датчик не увидит линию
- ○ Направо
- ○ Только для разворота

5. Почему каждое решение маршрута — это две строки: followLineToCrossing() и действие?

- ○ Вторая строка останавливает робота
- ○ Сначала надо доехать до следующего перекрёстка, иначе действие выполнится на том же
- ○ Так требует Arduino

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 сворачивает по датчикам и проходит маршрут из решений. Для итоговой задачи осталось научиться видеть груз и брать его захватом.

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

маршрут	Последовательность решений на перекрёстках: налево, направо или прямо.
центр перекрёстка	Точка, где пересекаются линии. Для поворота на месте над ней должны оказаться колёса.
массив	Список значений под одним именем: route[0], route[1] и так далее.
millis()	Сколько миллисекунд прошло с включения платы. Позволяет ехать заданное время, не останавливая программу.