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?

ЗАНЯТИЕ 1 · 45 МИН

1 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. Обсудите с напарником. Поставьте робота на перекрёсток и медленно поверните руками.
2. Следите за левым датчиком: когда он окажется над новой линией?

ДОЛЖНО ПОЛУЧИТЬСЯ

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. Если повернуть сразу, робот окажется сбоку от новой линии. Поэтому сначала надо проехать вперёд так, чтобы колёса встали над центром перекрёстка.

1. Поставь робота так, чтобы датчики были на поперечной линии.
2. Измерь линейкой, сколько ещё ехать колёсам до центра перекрёстка. Запиши в журнал.
3. Время этого проезда — `centerMs` — подберём на следующем шаге.

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты знаешь, почему перед поворотом нужен выезд на центр.

3 Поворот на перекрёстке

22 мин

1. Создай скетч `L42_Turn_At_Crossing`. Перенеси измерения, `kp` и `baseSpeed`.

Поворот на перекрёстке L42_Turn_At_Crossing.ino

```
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 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);
}
```

```
42
43  void followLineStep() {
44    int error = leftLevel - rightLevel;
45    int correction = kp * error;
46    drive(FORWARD, constrain(baseSpeed + correction, 0, 255),
47          FORWARD, constrain(baseSpeed - correction, 0, 255));
48  }
49
50  void setupRobot() {
51    pinMode(LEFT_DIR_PIN, OUTPUT);
52    pinMode(LEFT_SPEED_PIN, OUTPUT);
53    pinMode(RIGHT_DIR_PIN, OUTPUT);
54    pinMode(RIGHT_SPEED_PIN, OUTPUT);
55    stopMotors();
56    // Без измерений робот не поедет.
57    if (LEFT_WHITE == LEFT_BLACK || RIGHT_WHITE == RIGHT_BLACK) {
58      while (true) {
59      }
60    }
61    delay(2000);
62  }
63
64  // Перекрёсток: оба датчика видят чёрное.
65  bool onCrossing() {
66    return leftLevel < 30 && rightLevel < 30;
67  }
68
69  // Ехать по линии, пока не найдём перекрёсток.
70  void followLineToCrossing() {
71    while (true) {
72      readSensors();
73      if (onCrossing()) {
74        return;
75      }
76      followLineStep();
77    }
78  }
79
80  // Проехать перекрёсток прямо, пока датчики не съедут с чёрного.
81  void passCrossing() {
82    drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
83    do {
84      readSensors();
```

```
  } while (onCrossing());
  delay(100);
}

int turnSpeed = 100;
int centerMs = 250; // подбираем: проехать от датчиков до центра перекрёстка
int leaveMs = 200;  // время, чтобы датчик ушёл со старой линии

// 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);
}
```

```
128
129 // Ехать по линии заданное время.
130 void followLineFor(unsigned long ms) {
131   unsigned long start = millis();
132   while (millis() - start < ms) {
133     readSensors();
134     followLineStep();
135   }
136   stopMotors();
137 }
138
139 void setup() {
140   setupRobot();
141   followLineToCrossing();
142   turnLeftAtCrossing();
143   followLineFor(1500);
144   turnAroundOnLine();
145   followLineToCrossing();
146   stopMotors();
147 }
148
149 void loop() {
150 }
```

2. Разберите функцию `turnUntilLine`. Зачем в ней `delay(leaveMs)` перед поиском линии?
3. Запусти на дорожке с перекрёстком. Подбери `centerMs`: после поворота робот должен уверенно поехать по новой линии.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

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

ⓘ **Как это устроено: Зачем leaveMs**

В начале поворота датчик ещё над старой линией и сразу «нашёл бы» её. Поэтому первые `leaveMs` миллисекунд робот просто крутится, не глядя на датчики, — чтобы датчик ушёл со старой линии. Потом ищем новую.

🔧 **ЕСЛИ НЕ ПОЛУЧИЛОСЬ**

Что видишь	Что сделать
После поворота робот сбоку от линии	Подбери `centerMs`
Робот крутится бесконечно	Датчик не видит линию: проверь порог 30 и измерения
Робот останавливается сразу	Увеличь `leaveMs`

ЗАНЯТИЕ 2 · 45 МИН

2 Маршрут из решений

4 Разворот на линии

10 мин

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

1. В скетче найди вызов `turnAroundOnLine()`. Проверь на дорожке: проехать по ответвлению, развернуться, вернуться к перекрёстку.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот разворачивается, находит ту же линию и возвращается к перекрёстку.

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. Создай скетч `L42_Route`. Начерти в журнале тренировочную дорожку и путь робота.

Маршрут L42_Route.ino

```
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 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);
}
```

```
42
43  void followLineStep() {
44    int error = leftLevel - rightLevel;
45    int correction = kp * error;
46    drive(FORWARD, constrain(baseSpeed + correction, 0, 255),
47          FORWARD, constrain(baseSpeed - correction, 0, 255));
48  }
49
50  void setupRobot() {
51    pinMode(LEFT_DIR_PIN, OUTPUT);
52    pinMode(LEFT_SPEED_PIN, OUTPUT);
53    pinMode(RIGHT_DIR_PIN, OUTPUT);
54    pinMode(RIGHT_SPEED_PIN, OUTPUT);
55    stopMotors();
56    // Без измерений робот не поедет.
57    if (LEFT_WHITE == LEFT_BLACK || RIGHT_WHITE == RIGHT_BLACK) {
58      while (true) {
59      }
60    }
61    delay(2000);
62  }
63
64  // Перекрёсток: оба датчика видят чёрное.
65  bool onCrossing() {
66    return leftLevel < 30 && rightLevel < 30;
67  }
68
69  // Ехать по линии, пока не найдём перекрёсток.
70  void followLineToCrossing() {
71    while (true) {
72      readSensors();
73      if (onCrossing()) {
74        return;
75      }
76      followLineStep();
77    }
78  }
79
80  // Проехать перекрёсток прямо, пока датчики не съедут с чёрного.
81  void passCrossing() {
82    drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
83    do {
84      readSensors();
```

```
  } while (onCrossing());
  delay(100);
}

int turnSpeed = 100;
int centerMs = 250; // подбираем: проехать от датчиков до центра перекрёстка
int leaveMs = 200;  // время, чтобы датчик ушёл со старой линии

// 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);
}
```

```
128
129 // Ехать по линии заданное время.
130 void followLineFor(unsigned long ms) {
131   unsigned long start = millis();
132   while (millis() - start < ms) {
133     readSensors();
134     followLineStep();
135   }
136   stopMotors();
137 }
138
139 // Маршрут: решения на перекрёстках по порядку.
140 void setup() {
141   setupRobot();
142
143   followLineToCrossing();
144   turnLeftAtCrossing();   // решение 1: налево
145
146   followLineToCrossing();
147   passCrossing();         // решение 2: прямо
148
149   followLineToCrossing();
150   turnRightAtCrossing();  // решение 3: направо
151
152   followLineToCrossing();
153   stopMotors();           // конец маршрута
154 }
155
156 void loop() {
157 }
```

ПРЕДСКАЖИ

По программе: налево, прямо, направо. Нарисуй, где робот остановится на дорожке.

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

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

ОСНОВНОЕ · ПОВОРОТ И РАЗВОРОТ

Повернуть с перекрёстка налево и направо и уехать по новой линии; развернуться на линии и вернуться.

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

Основное

1. Почему поворот «до линии» надёжнее поворота по времени?

- ○ Он заканчивается, когда датчик видит новую линию, и не зависит от батареи
- ○ Он быстрее
- ○ Он не использует моторы

2. Зачем перед поворотом проехать немного вперёд (centerMs)?

- ○ Датчики впереди колёс: колёса должны встать над центром перекрёстка
- ○ Чтобы набрать скорость
- ○ Чтобы проверить батарею

3. Что случится, если в turnUntilLine убрать delay(leaveMs)?

- ○ Робот сразу остановится: датчик ещё видит старую линию
- ○ Робот повернёт быстрее и точнее
- ○ Ничего не изменится

4. Маршрут: налево, прямо, направо. Что выполнит робот на втором перекрёстке?

- ○ Проедет прямо
- ○ Повернёт налево
- ○ Повернёт направо

5. char route[] = {'L', 'S', 'R'}; Чему равно route[2]?

- ○ 'S'
- ○ 'R'
- ○ 'L'

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.