Тема 4 · Урок 4.1

Находим и считаем перекрёстки

Робот замечает перекрёсток, останавливается на нём и не считает один перекрёсток дважды.

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

РЕЗУЛЬТАТ УРОКА

Робот едет по линии с П-регулятором и останавливается на нужном по счёту перекрёстке. Каждый перекрёсток считается один раз.

Ты научишься

- Описать, что показывают датчики на перекрёстке.
- Записать условие перекрёстка и объяснить выбор порога.
- Объяснить, почему один перекрёсток может посчитаться много раз, и исправить это.
- Остановить робота на перекрёстке с заданным номером.

Нужно знать

- 3.2 П-регулятор: плавно и быстро

Понадобится

Датчик линии (аналоговый, Trema) × 2

Зачем этот урок

На полях итоговой задачи к грузам ведут ответвления от перекрёстков. Чтобы свернуть в нужное место, робот должен замечать перекрёстки и знать, какой по счёту перед ним. Сегодня робот научится находить, считать и проезжать перекрёстки.

РАЗМИНКА · ВСПОМНИ

1. Чему равна ошибка П-регулятора, когда линия ровно посередине?
2. Что показывает датчик по шкале 0–100 на чёрном?
3. Что делает `constrain`?

ЗАНЯТИЕ 1 · 45 МИН

1 Перекрёсток глазами датчиков

1 Что видят датчики на перекрёстке

12 мин

Что должен делать робот. Курьер должен заметить перекрёсток, доехать до него и при необходимости остановиться. Отдельный датчик перекрёстка не нужен: используем оба датчика линии и их показания вместе.

Что используем и как это работает. На обычной узкой дорожке датчики находятся по её краям. На широкой поперечной полосе оба могут одновременно увидеть чёрное. Программа проверяет это сочетание и решает, продолжать ли движение. Пока датчики остаются над одной полосой, многократные чтения относятся к одному перекрёстку, а не к нескольким.

Как связать измерение с действием. Сначала проверим признаки на неподвижном роботе, затем въезд и выезд с полосы. Для счёта нужно отделить обнаружение перекрёстка от проезда за его границу. Иначе быстрый цикл насчитает одну и ту же полосу много раз. Другая широкая тёмная область тоже может дать такой сигнал: датчики не знают географию поля.

ПРЕДСКАЖИ

Робот подъезжает к перекрёстку. Что покажут оба датчика по шкале 0–100, когда он окажется на поперечной линии?

1. Загрузи скетч `L32_Error` из урока 3.2.
2. Медленно провези робота руками по тренировочной дорожке через перекрёсток. Смотри на монитор порта.
3. Запиши в журнал показания: на линии посередине, на перекрёстке, после перекрёстка.

ДОЛЖНО ПОЛУЧИТЬСЯ

На перекрёстке оба датчика показывают почти 0: под ними чёрная поперечная линия.

2 Условие перекрёстка

10 мин

Запишем правило: **перекрёсток — когда оба датчика видят чёрное**. «Чёрное» — показание меньше порога 30.

```
bool onCrossing() {
  return leftLevel < 30 && rightLevel < 30;
}
```

`&&` читается как «и»: условие верно, только если верны обе части.

Подсказка: Почему 30, а не 0

Даже на чёрном показание редко ровно 0, а при движении робот видит край линии. Порог 30 надёжно отделяет «чёрное» от «края», но его можно подобрать по своим измерениям.

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить, почему на изгибе линии условие не срабатывает, а на перекрёстке — срабатывает.

3 Стоп на перекрёстке

20 мин

1. Создай скетч `L41_Crossing_Stop`. Перенеси измерения, `kp` и `baseSpeed` из журнала.

Стоп на перекрёстке L41_Crossing_Stop.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  void setup() {
81    setupRobot();
82    followLineToCrossing();
83    stopMotors();
84  }
```

```
85
86 void loop() {
87 }
```

2. Найди функцию `followLineToCrossing()`. Объясни напарнику, как из неё выходит программа.
3. Поставь робота на линию перед перекрёстком и запусти.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот едет по линии и останавливается на первом перекрёстке.

Как это устроено: Как работает followLineToCrossing

`while (true)` — повторять всегда. Внутри: прочитать датчики; если перекрёсток — `return`, то есть выйти из функции; иначе сделать шаг П-регулятора. После выхода программа продолжается со строки `stopMotors();` в `setup()`.

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Робот проезжает перекрёсток	Снизь `baseSpeed` или подними порог до 40
Робот останавливается на изгибе	Порог слишком высокий, попробуй 20
Робот не едет	Вписаны ли измерения из журнала?

ЗАНЯТИЕ 2 · 45 МИН

2 Считаем перекрёстки

4 Найди ошибку счёта

8 мин

НАЙДИ ОШИБКУ · РОБОТ НАСЧИТАЛ 40 ПЕРЕКРЁСТКОВ

Соседи хотели считать перекрёстки так:

```
void loop() {
  readSensors();
  if (onCrossing()) {
    count = count + 1;
  }
  followLineStep();
}
```

Робот проехал один перекрёсток, а `count` стал 40. Почему?

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить: пока робот едет по широкой чёрной полосе, условие верно много раз подряд, и каждый раз счётчик растёт.

5 Проехать перекрёсток

10 мин

Чтобы посчитать перекрёсток один раз, его нужно **проехать**: ехать прямо, пока датчики не съедут с чёрного.

```
void passCrossing() {
  drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
  do {
    readSensors();
  } while (onCrossing());
  delay(100);
}
```

`do { … } while (условие)` — сделать и повторять, пока условие верно. Короткая пауза в конце уводит датчики подальше от края поперечной линии.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Ты можешь объяснить, когда закончится цикл в `passCrossing()`.

6 Стоп на нужном перекрёстке

17 мин

1. Создай скетч `L41_Count`. В `TARGET` укажи, на каком перекрёстке остановиться.

Счёт перекрёстков L41_Count.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);
}

const int TARGET = 1; // на каком перекрёстке остановиться

void setup() {
  setupRobot();
  for (int i = 1; i < TARGET; i++) {
    followLineToCrossing();
    passCrossing();
  }
  followLineToCrossing();
  stopMotors();
}

void loop() {
}
```

ПРЕДСКАЖИ

`TARGET = 2`. Сколько раз выполнится цикл `for`? Что робот сделает на первом перекрёстке?

2. Проверь на дорожке с тремя перекрёстками: `TARGET` = 1, 2, 3.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот останавливается ровно на перекрёстке с номером `TARGET`.

7 Задания по уровням

10 мин

ОСНОВНОЕ · ПЕРВЫЙ ПЕРЕКРЁСТОК

Доехать по линии до первого перекрёстка и остановиться на нём.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу объяснить условие перекрёстка.
- [ ] Я знаю, почему перекрёсток нужно проехать, прежде чем искать следующий.
- [ ] Я могу остановить робота на нужном перекрёстке.

Проверь себя

Основное

1. По какому признаку программа узнаёт перекрёсток?

- ○ Оба датчика одновременно видят чёрное
- ○ Один датчик видит чёрное
- ○ Оба датчика видят белое

2. Что вернёт выражение leftLevel < 30 && rightLevel < 30, если leftLevel = 10, rightLevel = 80?

- ○ true — это перекрёсток
- ○ false — чёрное только слева
- ○ Ошибку

3. Почему счётчик перекрёстков растёт много раз на одном перекрёстке?

- ○ Пока робот на чёрной полосе, условие верно в каждом повторе loop()
- ○ Датчики сломаны
- ○ Так работает П-регулятор

4. Что делает return внутри followLineToCrossing()?

- ○ Останавливает моторы
- ○ Выходит из функции, программа продолжается после её вызова
- ○ Начинает программу заново

5. TARGET = 3. Сколько перекрёстков робот проедет прямо до остановки?

- ○ Два
- ○ Три
- ○ Ни одного

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 находит перекрёстки и считает их без повторов. Следующий шаг — сворачивать на перекрёстке в нужную сторону и проходить маршрут.

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

перекрёсток	Место, где линию пересекает другая линия. Оба датчика одновременно видят чёрное.
порог	Число для сравнения: показание меньше 30 по шкале 0–100 считаем чёрным.
цикл while	Повторяет команды, пока условие верно.
return	Немедленно завершает функцию и возвращает программу туда, откуда функцию вызвали.