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`?

1

ЗАНЯТИЕ 1 · 45 МИН

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? А на изгибе линии?

1. Загрузи скетч со шкалой из урока 3.2 (`L32_Error` или свой шаблон).
2. Медленно провези робота руками по дорожке через изгиб и через перекрёсток. Смотри на монитор порта.
3. Заполни таблицу и ответь: по какому признаку перекрёсток можно отличить от изгиба?

Положение	leftLevel	rightLevel
Линия посередине		
На изгибе		
На перекрёстке		
После перекрёстка		

ДОЛЖНО ПОЛУЧИТЬСЯ

Таблица заполнена, признак перекрёстка записан словами.

2 Условие перекрёстка

10 мин

Сформулируй правило сам по своим записям из прошлого шага: **какие показания бывают только на перекрёстке** и никогда — на прямой или изгибе?

1. Запиши правило словами в журнал: что должно быть у левого датчика и что у правого.
2. Выбери порог «чёрного» по своим измерениям. Объясни, почему не 0 и почему не 50.
3. Запиши условие на языке Arduino. Понадобится знак «и»: `&&`.

ПРЕДСКАЖИ

Кто-то написал условие через `||` («или»). Где робот тогда остановится по ошибке?

Подсказка: Как записать «и»

`a < 30 && b < 30` верно, только если верны обе части.

ДОЛЖНО ПОЛУЧИТЬСЯ

В журнале — правило словами, порог с объяснением, условие и ответ про `||`.

3 Стоп на перекрёстке

20 мин

1. Создай скетч `L41_Crossing_Stop` по шаблону. Перенеси измерения, `kp` и `baseSpeed` из журнала. Функции линии из урока 3.2 уже готовы.
2. Напиши сам `onCrossing()` по своему правилу из прошлого шага, `followLineToCrossing()` и движение в `setup()`. Пока `onCrossing()` не написана, она всегда отвечает «перекрёсток», и робот не трогается.

Шаблон: стоп на перекрёстке L41_Crossing_Stop_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;

// ===== Твоя часть: напиши сам =====

bool onCrossing() {
  // true — только когда робот на поперечной линии; на прямой и на изгибе — f
  return true;
}

void followLineToCrossing() {
  // Ехать по линии с П-регулятором, пока не найдёшь перекрёсток; на перекрёс
}

void setup() {
  setupRobot();

  // Доехать по линии до первого перекрёстка и остановиться.
}

// ===== Готовое из уроков 3.1–3.2 — 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);
}

void loop() {
```

```
85 }
```

ПРЕДСКАЖИ

Что сделает робот, если после поиска перекрёстка забыть остановить моторы?

3. Поставь робота на линию перед перекрёстком и запусти.

Подсказка: Ехать, пока…

Функция повторяет одно и то же — прочитать датчики, сделать шаг по линии — и заканчивается, когда найден перекрёсток. Какой цикл повторяет «всегда», и как выйти из функции посреди него?

Подсказка: Как выйти из функции

`return` внутри функции сразу заканчивает её. Программа продолжается со следующей строки в `setup()`.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот едет по линии и останавливается на первом перекрёстке. Прогноз записан.

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Робот проезжает перекрёсток	Снизь `baseSpeed` или подними порог
Робот останавливается на изгибе	Порог слишком высокий
Робот не едет	Вписаны ли измерения? Написана ли `onCrossing()`?

ЗАНЯТИЕ 2 · 45 МИН

2 Считаем перекрёстки

4 Найди ошибку счёта

8 мин

НАЙДИ ОШИБКУ · ПАУЗА НЕ ПОМОГЛА

Соседи заметили двойной счёт и добавили паузу после счёта:

```
void loop() {
  readSensors();
  if (onCrossing()) {
    count = count + 1;
    delay(100);
  }
  followLineStep();
}
```

Робот проехал один перекрёсток, а `count` стал 3. Почему пауза не помогла? Чем она ещё опасна для езды по линии?

1. Запиши ответ в журнал.
2. Предложи исправление словами: что должна сделать программа после того, как нашла перекрёсток, прежде чем искать следующий?

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить, почему пауза не помогает, и предложить исправление.

5 Проехать перекрёсток

10 мин

Чтобы посчитать перекрёсток один раз, его нужно **проехать**: ехать прямо, пока датчики не съедут с чёрного. Напиши функцию `passCrossing()` в журнале:

1. Включи оба мотора вперёд со скоростью `baseSpeed`.
2. Читай датчики, **пока** робот на перекрёстке.
3. После этого подожди 100 мс.

`do { … } while (условие);` — сделать и повторять, пока условие верно.

Подсказка: Почему do, а не while

Датчики нужно прочитать хотя бы один раз, прежде чем проверять условие.

Подсказка: Зачем пауза в конце

Сразу после чёрного датчики видят край поперечной линии. Что будет, если в этот момент снова искать перекрёсток?

ДОЛЖНО ПОЛУЧИТЬСЯ

Функция записана в журнал. Ты можешь объяснить, когда закончится цикл и зачем пауза.

6 Стоп на нужном перекрёстке

17 мин

1. Создай скетч `L41_Count` по шаблону. В `TARGET` укажи, на каком перекрёстке остановиться.
2. Напиши сам `passCrossing()` — по своей записи в журнале — и движение в `setup()`. Пока твоя часть не написана, робот не трогается.

Шаблон: счёт перекрёстков L41_Count_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;
const int TARGET = 1; // на каком перекрёстке остановиться

// ===== Твоя часть: напиши сам =====

void passCrossing() {
  // Проехать перекрёсток прямо. Функция заканчивается, когда датчики уже не
}

void setup() {
  setupRobot();

  // Перекрёстки до TARGET — найти и проехать прямо; на перекрёстке номер TAR
}

// ===== Готовое из уроков 3.2 и 4.1 — 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();
  }
}

void loop() {
}
```

ПРЕДСКАЖИ

`TARGET = 3`. Сколько раз робот должен проехать перекрёсток прямо?

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

Подсказка: Повторить нужное число раз

Цикл `for` из урока 1.1 повторяет команды заданное число раз. Сколько перекрёстков нужно проехать перед тем, на котором остановиться?

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот останавливается ровно на перекрёстке с номером `TARGET`.

7 Задания по уровням

10 мин

★ **СРЕДНИЙ УРОВЕНЬ · ТРЕТИЙ ПЕРЕКРЁСТОК**

Остановиться на третьем перекрёстке, проехав два первых прямо. На среднем поле итоговой задачи робот проезжает перекрёстки, не сворачивая.

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

★ Средний уровень

1. Счётчик crossings начинается с 0 и увеличивается на каждом перекрёстке. При каком значении остановиться на третьем перекрёстке?

- ○ 3
- ○ 2
- ○ 4

2. Условие перекрёстка записали через «или»: leftLevel < 30 || rightLevel < 30. Где робот остановится?

- ○ Нигде: условие никогда не выполнится
- ○ На первой дуге или при вилянии: хватает одного тёмного датчика
- ○ Только на перекрёстке

3. Почему порог «чёрного» — 30, а не 0?

- ○ С нулём нельзя сравнивать
- ○ С порогом 0 робот едет медленнее
- ○ Даже на чёрном показание редко ровно 0

4. В шаблоне onCrossing() пока возвращает true. Что делает робот после запуска?

- ○ Сразу останавливается: «перекрёсток» найден в первом же повторе
- ○ Едет по линии без остановки
- ○ Крутится на месте

5. Зачем в passCrossing() цикл do … while, а не while?

- ○ while не умеет вызывать функции
- ○ Датчики нужно прочитать хотя бы один раз, прежде чем проверять условие
- ○ do выполняется быстрее

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.