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.

Проверь идею «крутиться до новой линии» измерением, прежде чем её программировать.

1. Положи под робота лист с транспортиром или линиями через 10°. Поставь робота на центр перекрёстка.
2. Медленно поворачивай руками налево. Запиши угол, на котором левый датчик **впервые** видит новую линию, и угол, на котором робот смотрит точно вдоль неё.
3. Повтори направо.
4. Совпадают ли углы? Что тогда сделает робот, который останавливается в момент, когда датчик увидел линию? Как это исправить?

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

В журнале — два угла для каждого поворота и вывод о том, где остановится робот.

2 Выезд на центр перекрёстка

12 мин

Датчики стоят **впереди** колёс, поэтому перед поворотом робот выезжает на центр перекрёстка. Рассчитай время этого выезда, а не подбирай наугад.

1. Измерь расстояние от датчиков до оси колёс в миллиметрах.
2. Возьми из журнала урока 2.1 свою скорость в миллиметрах в секунду при близкой `baseSpeed`. Нет таблицы — измерь: проезд 1 секунду, путь линейкой, три раза.
3. Посчитай `centerMs` и запиши расчёт. На следующем шаге сравни его с подобранным на дорожке.
4. Запиши, почему выезд по времени — слабое место и от чего его точность зависит.

ДОЛЖНО ПОЛУЧИТЬСЯ

В журнале — расстояние, скорость, расчёт `centerMs` и две причины, по которым выезд по времени может ошибаться.

3 Поворот на перекрёстке

22 мин

Требование 1. Продолжи свою программу `L41_Count` и сохрани как `L42_Turn_At_Crossing`. Функция `turnUntilLine(left)` поворачивает робота на месте до новой линии:

- сначала робот крутится, не глядя на датчики, пока датчик не уйдёт со старой линии;
- потом — пока датчик со стороны поворота не увидит линию;
- если линия не нашлась за 3 секунды, робот останавливается: на поле без защиты он будет крутиться бесконечно.

Требование 2. `turnLeftAtCrossing()` и `turnRightAtCrossing()`: выезд на центр по твоему `centerMs`, затем поворот до линии.

Приёмка. По пять поворотов налево и направо: после каждого робот уезжает по новой линии. Отдельно проверь защиту: подними робота над столом во время поворота.

Подсказка: Идея (1 из 3)

Поиск линии — цикл «читать датчики, пока линии нет». Для защиты запомни время начала поиска.

Подсказка: План (2 из 3)

1. Включить поворот, подождать `leaveMs`.
2. Запомнить `millis()`.
3. Повторять чтение, пока линии нет и прошло меньше 3 секунд.
4. Остановить моторы.

Подсказка: Фрагмент кода (3 из 3)

`while ((left ? leftLevel : rightLevel) > 30 && millis() - start < 3000)` — прочитай условие словами.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Программа выполняет требования 1–2, десять поворотов и проверка защиты записаны. Подобранный `centerMs` сравнён с расчётом.

2

ЗАНЯТИЕ 2 · 45 МИН

Маршрут из решений

4 Разворот на линии

10 мин

Требование 3. `turnAroundOnLine()` разворачивает робота на линии и останавливается, когда линия снова под датчиком. Защита 3 секунды из требования 1 работает и здесь.

Приёмка. Пять разворотов на конце ответвления: после каждого робот возвращается к перекрёстку. Для каждого измерь, насколько нос смотрит мимо линии сразу после разворота.

Разворот	Вернулся?	Нос мимо линии, мм
1		

Если нос каждый раз смотрит в одну сторону, предложи и проверь одно исправление.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Пять разворотов в таблице, исправление систематической ошибки проверено.

5 Маршрут по порядку

22 мин

Требование 4. Маршрут планируется до программы и проверяется по частям.

1. Выбери на тренировочном поле путь из четырёх решений или больше, в котором есть налево, направо и прямо. Если в классе разложено поле итоговой задачи, возьми путь от старта до одной зоны доставки. Запиши его таблицей:

№	Перекрёсток	Решение	Куда смотрит робот после
1			

2. Запиши маршрут в программу `L42_Route` из своих функций.
3. Если маршрут сломался, не переписывай всё: запусти только первые решения, потом половину, и найди первое решение, на котором робот ошибается. Запиши, как искал.

ПРЕДСКАЖИ

На каком решении маршрут скорее всего сломается и почему? Запиши до первого запуска.

Подсказка: Как проверять по частям

Временно поставь `stopMotors();` и `return;` после нужного решения в `setup()`.

ДОЛЖНО ПОЛУЧИТЬСЯ

Таблица маршрута, программа по ней, прогноз и запись поиска ошибки. Робот проходит маршрут на тренировочной дорожке.

6 Задания по уровням

13 мин

★★ **ПРОДВИНУТЫЙ УРОВЕНЬ · МАРШРУТ МАССИВОМ**

Запиши решения массивом: `char route[] = {'L', 'S', 'R', 'L'};`. Напиши функцию `decide(d)`, которая выполняет одно решение, и цикл, который проходит весь массив. Составь маршрут по схеме поля так, чтобы для нового маршрута менялась только строка с массивом.

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

★★ Продвинутый уровень

1. Какое решение выполнится третьим?

```
1 char route[] = {'L', 'S', 'R', 'L'};
```

- ○ Направо
- ○ Прямо
- ○ Налево

2. Что нужно изменить в программе с массивом, чтобы пройти новый маршрут?

- ○ Только строку с массивом route
- ○ Функцию decide()
- ○ Весь цикл обхода маршрута

3. От датчиков до оси колёс 60 мм, скорость робота 200 мм/с. Каким должен быть centerMs по расчёту?

- ○ 120 мс
- ○ 3000 мс
- ○ 300 мс

4. Робота подняли над столом во время поворота. Что произойдёт?

```
unsigned long start = millis();
do {
  readSensors();
} while ((left ? leftLevel : rightLevel) > 30 && millis() - start < 3000);
stopMotors();
```

- ○ Через 3 секунды моторы выключатся, хотя линия не найдена
- ○ Робот будет крутиться бесконечно
- ○ Моторы выключатся сразу

5. Маршрут из восьми решений ломается. Как быстрее найти сломанное решение?

- ○ Переписать маршрут заново
- ○ Запустить первую половину маршрута, потом половину оставшейся части
- ○ Каждый раз запускать весь маршрут и смотреть

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.