Тема 6 · Урок 6.1

Робот-курьер: план и программа

Итоговая задача, часть 1: выбираешь уровень, планируешь маршрут и собираешь программу из проверенных частей.

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

РЕЗУЛЬТАТ УРОКА

Выбран уровень итоговой задачи, записан и проверен план маршрута. Программа проходит весь маршрут хотя бы в пробном заезде, работающие версии сохранены.

Ты научишься

- Выбрать уровень итоговой задачи по своим выполненным заданиям ●, ★ и ★★.
- Разбить задачу на участки и записать план маршрута до написания кода.
- Собрать программу из функций прошлых уроков и проверять её по частям.
- Сохранять рабочую версию после каждой проверенной части.

Нужно знать

- 2.1 Моторы: едем и останавливаемся
- 2.2 Повороты и развороты
- 3.1 Датчики линии и первое следование
- 3.2 П-регулятор: плавно и быстро
- 4.1 Находим и считаем перекрёстки
- 4.2 Повороты на перекрёстке и маршрут
- 5.1 Ультразвук: видим груз
- 5.2 Захват: берём и ставим

Понадобится

- Ультразвуковой датчик расстояния
- Захват с сервоприводом
- Датчик линии (аналоговый, Trema) × 2

Твоя задача

Ты — инженер службы доставки. Такие роботы-курьеры уже ездят по тротуарам и складам: сами находят дорогу, забирают груз и привозят его по адресу. Твой робот должен сам выехать со старта, перевезти два груза на новые места и остановиться в финише. Во время заезда ему никто не помогает: всё решает твоя программа.

Новых команд здесь нет. Всё нужное ты уже делал: ездил и поворачивал, держался линии с П-регулятором, находил перекрёстки, видел груз ультразвуком и работал захватом. Сейчас нужно собрать эти умения в одну программу. В уроке 6.2 — зачётные заезды.

РАЗМИНКА · ВСПОМНИ

1. Чем `setup()` отличается от `loop()` и где должен выполняться маршрут, чтобы после финиша робот стоял?
2. Как программа понимает, что под роботом перекрёсток?
3. Что делать, если ультразвук не получил эхо?

1

ЗАНЯТИЕ 1 · 45 МИН

Уровень и план

1 Выбери уровень

6 мин

Что должен делать робот. Теперь роботу нужно доставить два груза и закончить поездку на финише. Нового датчика не добавляем: объединяем уже проверенные возможности — движение, поиск линии и перекрёстков, подъезд по ультразвуку и захват.

Что используем и как это работает. План — последовательность действий с условиями завершения. Для каждого участка запиши, какие показания нужны, какое действие выполняют моторы или захват и по какому признаку переходить дальше. Функции позволяют назвать эти действия и проверить их по отдельности; вызов функции передаёт управление её командам и затем возвращает его в маршрут.

Как связать измерение с действием. Перед сборкой маршрута проверь каждую функцию своим коротким опытом. Пустая функция в заготовке не выполняет действие, даже если программа успешно собирается. После последнего участка нужна явная остановка; повторение маршрута в loop без решения автора запустило бы доставку заново.

Задача есть в трёх уровнях. Правила и критерии оценки у них общие, а поля разные. Уровень выбирают по тому, какие задания уроков ты уже выполнил: задания ★ готовили к среднему полю, ★★ — к продвинутому.

Уровень итоговой задачи	Какие задания уроков нужны
Начальный S-образная линия, два перекрёстка, порядок действий известен заранее.	● Основное 1.1 2.1 2.2 3.1 3.2 4.1 4.2 5.1 5.2
Средний Сеть дорог, четыре перекрёстка, проезд прямо и повторные встречи с перекрёстками.	● Основное 1.1 2.1 2.2 3.1 3.2 4.1 4.2 5.1 5.2 ★ Средний уровень 1.1 2.1 2.2 3.1 3.2 4.1 4.2 5.1 5.2
Продвинутый Шесть перекрёстков, крутые S-дуги, маршрут по карточке составляешь сам.	● Основное 1.1 2.1 2.2 3.1 3.2 4.1 4.2 5.1 5.2 ★ Средний уровень 1.1 2.1 2.2 3.1 3.2 4.1 4.2 5.1 5.2 ★★ Продвинутый уровень 1.1 2.1 2.2 3.1 3.2 4.1 4.2 5.1 5.2

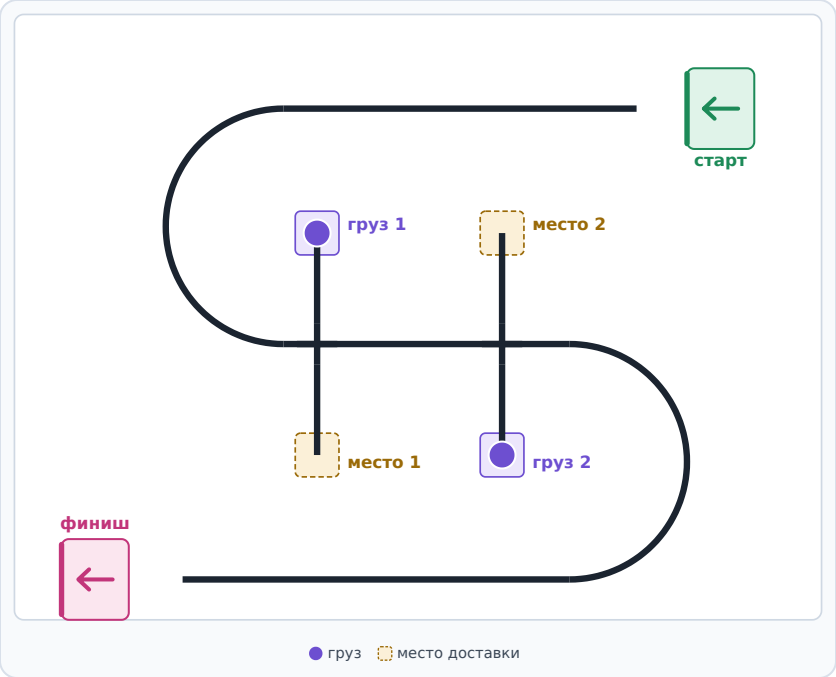
1. Открой бортовой журнал и посмотри, какие задания ●, ★ и ★★ у тебя получились.
2. Выбери уровень сам и отметь его ниже. Если сомневаешься, начни с начального: после зачёта можно попробовать следующий.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Уровень выбран и записан в журнал.

2 Разберись в задании своего уровня

8 мин



Начальное поле.

1. Выехать из зоны старта влево и найти линию.
2. По линии пройти левую дугу и доехать до первого перекрёстка.
3. Перевезти груз 1 с верхнего конца ответвления на нижний — в место 1.
4. Доехать до второго перекрёстка и перевезти груз 2 с нижнего конца на верхний — в место 2.
5. Пройти правую дугу, выехать с конца линии в зону финиша и остановиться.

У старта и финиша есть короткие участки без линии: там робот едет прямо по твоим командам.

1. Покажи пальцем на схеме: старт, оба груза, оба места доставки, финиш.
2. Прочитай критерии оценки в конце урока. За что дают больше всего баллов?

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Ты можешь пересказать задание своего уровня своими словами.

3 Проверь умения по отдельности

10 мин

> ⚠ **БЕЗОПАСНОСТЬ**
> После загрузки старой программы робот может сразу поехать. Загружай программы при выключенном питании моторов, первый запуск — на подставке.

Если хотя бы одно умение не работает, вся программа не поедет. Проверь каждое коротким опытом.

1. Оба колеса крутятся вперёд и назад; моторы стоят, когда выключены (урок 2.1).
2. Робот поворачивает на 90° и разворачивается на 180° (урок 2.2).
3. Робот едет по линии с П-регулятором и проходит дуги своего поля (урок 3.2).
4. Робот находит перекрёсток и поворачивает с него на нужную линию (уроки 4.1, 4.2).
5. Ультразвук видит груз на нужном расстоянии (урок 5.1).
6. Захват берёт, держит и отпускает груз (урок 5.2).

> ✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**
> Все умения проверены. То, что не работает, ты сначала чинишь по разделу «Если не получилось» своего урока.

4 Составь план маршрута

16 мин

Хороший инженер сначала планирует, потом пишет код. Самое сложное — повороты на обратном пути: «налево» и «направо» считаются от носа робота, а не от тебя.

Участок	Что делает робот	Какая функция
Старт → линия	Едет прямо до линии	`driveStraight()`

...

1. Заполни таблицу плана в бортовом журнале. Одна строка — один участок.
2. Для каждого перекрёстка запиши: откуда приехал, куда повернул, что делаешь дальше.
3. Проведи выключенного робота по плану руками. Куда смотрит его нос после каждого разворота?
4. Запиши, где скорее всего будет ошибка и как ты это проверишь.

В плане семь участков: выезд, линия до перекрёстка 1, груз 1, место 1, перекрёсток 2, груз 2 и место 2, финиш.

Подсказка: Робот развернулся — а что теперь «налево»?

Встань и повернись так же, как робот. Твоя левая рука показывает его «налево».

ДОЛЖНО ПОЛУЧИТЬСЯ

План записан. Напарник прошёл по нему с выключенным роботом и не нашёл ошибок.

2

ЗАНЯТИЕ 2 · 45 МИН

Программа по частям

5 Подготовь заготовку

6 мин

Программу собираем из функций. Функция — действие с понятным именем, например `turnLeft()`. Когда маршрут записан вызовами функций, его легко сверить с планом.

Заготовка с функциями L61_Courier.ino

```ino
// Робот-курьер: моя программа итоговой задачи.
// Это заготовка, а не решение. Наполни функции своим проверенным кодом
// из прошлых уроков и проверяй каждую отдельно, прежде чем собирать маршрут.

// --- Моторы (урок 2.1) ---
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;

// --- Датчики линии (уроки 3.1 и 3.2) ---
const int LEFT_SENSOR_PIN = A4;
const int RIGHT_SENSOR_PIN = A5;
// Перенеси сюда свои измерения белого и чёрного и коэффициент П-регулятора.

// --- Ультразвук (урок 5.1) и захват (урок 5.2) ---
// Перенеси сюда контакты и настройки из своих проверенных скетчей.

void stopMotors() {
  // Остановить оба мотора (урок 2.1).
}

void driveStraight() {
  // Проехать прямо проверенное время (урок 2.1), например въезд в зону финиш
}

void turnLeft() {
  // Повернуть налево на 90° (урок 2.2) или с перекрёстка до линии (урок 4.2)
}

void turnRight() {
  // Повернуть направо.
}

void turnAround() {
  // Развернуться на 180° (урок 2.2).
}

void followLineToCrossing() {
  // Ехать по линии с П-регулятором до перекрёстка (уроки 3.2, 4.1).
```

```
  // После перекрёстка сначала съедь с широкой чёрной области.
}

void approachCan() {
  // Ехать по линии и остановиться перед грузом по ультразвуку (урок 5.1).
}

void pickUp() {
  // Остановить моторы и закрыть захват (урок 5.2).
}

void putDown() {
  // Открыть захват и отпустить груз (урок 5.2).
}

void setup() {
  // Подготовь выводы, датчики и захват. Убедись, что моторы стоят.

  // Здесь — твой маршрут: вызывай функции в порядке своего плана.

  stopMotors();
}

void loop() {
  // Задание выполняется один раз в setup(). После финиша робот стоит.
}
```

1. Открой заготовку и сохрани её под своим именем.
2. Перенеси в функции свой проверенный код из прошлых уроков.

ⓘ Как это устроено: Зачем делить программу на функции

Если поворот работает неправильно, исправлять нужно одно место — саму функцию, а не каждый поворот на маршруте. А маршрут в `setup()` читается почти как план из журнала.

★★ ПРОДВИНУТЫЙ УРОВЕНЬ · МАРШРУТ МАССИВОМ

Запиши решения на перекрёстках массивом, например `char route[] = {'L', 'S', 'R'};`, и выполняй их по очереди в цикле, как в уроке 4.2. Тогда для новой карточки достаточно поменять одну строку.

6 Часть 1: от старта до первого груза

12 мин

1. В `setup()` вызови функции первого участка: выезд, линия, перекрёсток, поворот, подъезд к грузу.
2. Проверь на поле только эту часть.
3. Работает — сохрани копию: «рабочая версия 1».

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот выезжает со старта, доезжает по линии до груза 1 и останавливается перед ним.

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что проверить
Робот не находит линию после старта	Смотрит ли нос робота вдоль линии? Хватает ли длины выезда?
Робот останавливается на дуге	Дуга — не перекрёсток. Проверь условие поиска перекрёстка.
Робот проезжает перекрёсток	Скорость велика или условие перекрёстка не срабатывает.
После поворота робот теряет линию	Проверь, что поворот заканчивается, когда датчик нашёл новую линию.

7 Часть 2: первая доставка

12 мин

1. Допиши: захват, разворот, путь к месту 1, отпускание, возвращение к дороге.
2. Проверь только эту часть: поставь робота перед грузом 1 и запусти.
3. Работает — сохрани «рабочую версию 2».

Подсказка: Когда закрывать захват?

Сначала останови моторы, потом закрывай захват. Подъезд заканчивай по измерению ультразвука. Если эха нет, это не значит «груз совсем рядом».

Подсказка: Груз уезжает вместе с роботом

Отъезжай только после того, как захват полностью открылся.

ДОЛЖНО ПОЛУЧИТЬСЯ

Груз 1 стоит в месте 1, захват открыт, робот на дороге и смотрит туда, куда нужно дальше.

8 Часть 3: вторая доставка и финиш

15 мин

1. Допиши вторую доставку и путь к финишу. Груз 2 стоит в другом месте, поэтому повороты будут другими — сверься с планом.
2. В конце — прямой въезд в финиш и остановка. После финиша робот должен стоять.
3. Запусти весь маршрут от старта. Сохрани «рабочую версию 3».

Подсказка: Как понять, что пора въезжать в финиш

Чёрной метки финиша на поле нет. Используй последний ориентир — конец дуги или последний перекрёсток — и проверенное время прямого движения. Два «белых» показания ещё не значат, что линия кончилась: она может быть между датчиками.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот проходит весь маршрут хотя бы в одной пробе. Все рабочие версии сохранены.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу показать в программе каждый участок своего плана.
- [ ] Я знаю, какая часть работает ненадёжно и что я проверю в следующем уроке.

Критерии оценки

Зачётные заезды пройдут в уроке 6.2 по этим критериям.

Как оценивается заезд

Максимум **100** | Зачёт от **70**

Критерий	Баллы	Макс.
Прямой выезд со старта	5 — Сам выехал вперёд из зоны старта. 10 — Доехал до линии и продолжил задание без подталкивания.	**10**
Движение по линии	10 — Терял линию, но сам возвращался на неё и продолжал маршрут. 20 — Прошёл все участки линии и дуги по датчикам, не теряя линию.	**20**
Выбор дороги на перекрёстках	5 — За каждое из четырёх решений на перекрёстках, которые совпали с планом.	**20**
Груз 1: обнаружение	5 — Остановился перед грузом по ультразвуку.	**5**
Груз 1: захват	5 — Взял груз захватом и увёз, а не вытолкнул.	**5**
Груз 1: доставка	5 — Груз стоит и частично находится в нужной зоне. 10 — Груз стоит целиком в нужной зоне, захват отпущен.	**10**
Груз 2: обнаружение	5 — Остановился перед грузом по ультразвуку.	**5**
Груз 2: захват	5 — Взял груз захватом и увёз, а не вытолкнул.	**5**
Груз 2: доставка	5 — Груз стоит и частично находится в нужной зоне. 10 — Груз стоит целиком в нужной зоне, захват отпущен.	**10**
Прямой въезд и остановка	5 — После линии сам въехал вперёд в зону финиша. 10 — Остановился внутри зоны и стоит с выключенными моторами не меньше 3 секунд.	**10**

Обязательные условия зачёта

- Команда объясняет свою программу: показывает прямое движение, расчёт П-регулятора, остановку по ультразвуку, управление захватом и строку, которая завершает поездку.
- Изгибы пройдены по показаниям датчиков линии с П-регулятором.

- Подъезд к грузу заканчивается по измерению ультразвука, а не только по времени.
- После старта робота никто не трогает и не отправляет ему команд.

Правила

- Скорость не оценивается: нет бонуса за быстрый заезд.
- Отрицательных штрафов нет. Невыполненное действие просто не получает баллов.
- Груз в чужой зоне или доставленная не по порядку — 0 за её доставку.
- После помощи руками автономный заезд окончен. Продолжать можно для тренировки, но баллы не добавляются.
- Каждая попытка оценивается отдельно. Лучшие части разных попыток не складываются.
- Аварийная остановка ради безопасности не штрафуется.

Проверь себя

● Основное

1. Робот ехал на восток и развернулся на 180°. Теперь ему нужно повернуть на север. Куда он поворачивает?

- ○ Налево
- ○ Направо
- ○ Прямо

2. Робот нашёл перекрёсток и повернул. Почему перед поиском следующего перекрёстка нужно сначала съехать с чёрной области?

- ○ Иначе программа сразу найдёт тот же самый перекрёсток ещё раз
- ○ Иначе сядет батарея
- ○ Так требует П-регулятор

3. Поворот налево оформлен функцией turnLeft() и вызывается четыре раза. Поворот работает неправильно. Сколько мест нужно исправить?

- ○ Одно — саму функцию
- ○ Четыре — каждый вызов
- ○ Пять

4. Зачем собирать программу по частям и сохранять копию после каждой работающей части?

- ○ Если появится ошибка, понятно, в какой части её искать, и можно вернуться к рабочей версии
- ○ Так программа быстрее загружается
- ○ Так требует Arduino

5. На среднем поле робот проезжает перекрёсток B несколько раз. Как программа понимает, что делать на нём в этот раз?

- ○ Читает букву на поле
- ○ Знает, какое по счёту решение маршрута сейчас выполняет
- ○ Всегда поворачивает налево

★ Средний уровень

1. **На среднем поле одни и те же перекрёстки встречаются несколько раз. Что поможет не перепутать решения?**

- ○ Записать маршрут списком решений по порядку и сверить его со схемой
- ○ Всегда поворачивать налево
- ○ Проезжать все перекрёстки без решений

2. **У строки плана «проехать перекрёсток Б прямо» нет своей функции. Что сделать до сборки маршрута?**

- ○ Найти или дописать функцию для этого действия и проверить её отдельно
- ○ Пропустить строку плана
- ○ Убрать перекрёсток из плана

3. **Часть 1 сработала один раз из трёх. Можно ли переходить к части 2?**

- ○ Да: часть 2 исправит часть 1
- ○ Нет: сначала найти, что срывается, и добиться трёх из трёх
- ○ Да: один раз уже достаточно

4. **Весь маршрут сорвался в части 3. Как быстрее проверить исправление?**

- ○ Каждый раз запускать маршрут от старта
- ○ Переписать часть 1
- ○ Поставить робота в начало части 3 и запустить только её

5. **Почему маршрут записывают в setup(), а не в loop()?**

- ○ После финиша программа не должна начинать маршрут заново
- ○ loop() не умеет вызывать функции
- ○ setup() выполняется быстрее

★★ **Продвинутый уровень**

1. **Карточка маршрута: налево, направо, прямо. Какая запись верна?**

- ○ char route[] = {'L', 'R', 'S'};
- ○ char route[] = {'L', 'S', 'R'};
- ○ char route[] = {'R', 'L', 'S'};

2. **Для новой карточки пришлось менять функцию turnLeftAtCrossing(). Что это говорит о программе?**

- ○ Всё в порядке: так и должно быть
- ○ Маршрут не отделён от действий: данные и функции перемешаны
- ○ Карточка составлена неверно

3. Функция подъезда ждёт груз, а груз забыли поставить. Что должно уберечь робота?

- ○ Ничего: робот сам остановится
- ○ П-регулятор
- ○ Защита от бесконечного ожидания: остановка по времени или в конце линии

4. Часть 1 проходит, если робот стоит на старте ровно, и срывается, если он на 2 см левее. Где слабое место?

- ○ Выезд со старта и поиск линии
- ○ Захват
- ○ Ультразвук

5. Маршрут по карточке из четырёх решений обрывается раньше. Почему?

```
char route[] = {'L', 'S', 'R', 'L'};
for (int i = 0; i < 3; i++) {
  followLineToCrossing();
  decide(route[i]);
}
```

- ○ decide не понимает 'S'
- ○ Цикл выполняет только три решения: последнее не выполнится
- ○ Элемента route[3] не существует

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

У тебя есть план и программа, которая проходит маршрут. В уроке 6.2 ты сделаешь её надёжной и проведёшь зачётные заезды.

Как подтвердить результат

Результат опыта связывает условия, версию программы, наблюдение и вывод. Сравнение попыток с одним изменением помогает проверить причину, включая неудачные попытки. Отметка относится к уже выполненной практике; повтор ради отметки не нужен. Расхождение с прогнозом задаёт следующую проверяемую гипотезу. Модель, измерения настоящего робота и объяснение — разные виды доказательств.

Словарь урока

функция	Именованный блок команд, который можно вызывать много раз: например, turnLeft().
план маршрута	Таблица участков пути: что делает робот и какая функция за это отвечает.
рабочая версия	Копия программы, в которой всё написанное уже проверено на поле.