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. Поле выбирают по тому, какие задания уроков ты уже выполнил.

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

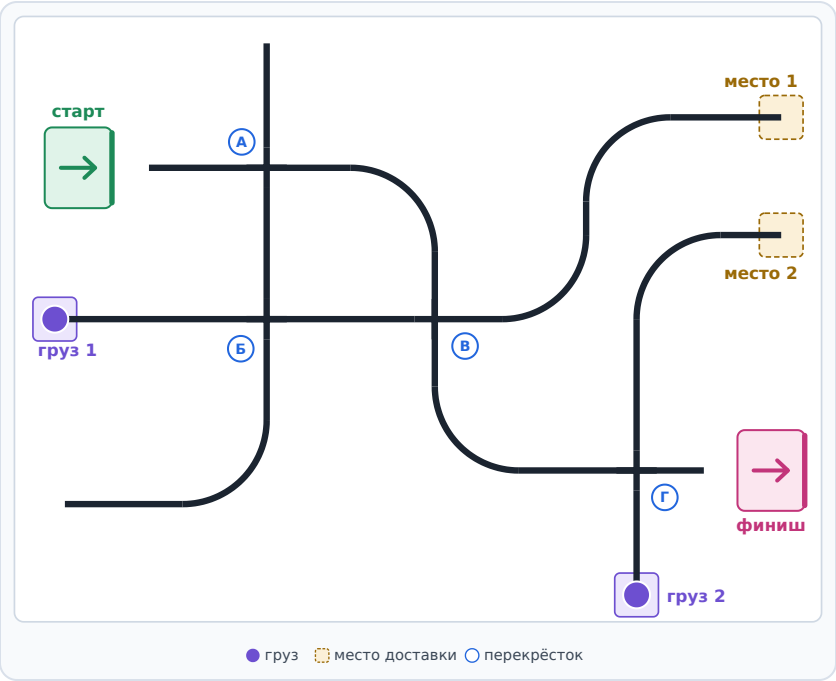
1. Открой бортовой журнал. Выпиши, какие задания ★ у тебя получились и в каких уроках.
2. Сравни со списком, который нужен для среднего поля. Чего не хватает?
3. Выбери поле сам. Если выбираешь среднее, но какого-то задания ★ нет, запиши, как и когда ты его доделаешь.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Поле выбрано, в журнале — список выполненных заданий ★ и план для недостающих.

2 Разберись в задании своего уровня

8 мин



Среднее поле. Перекрёстки отмечены буквами А, Б, В, Г.

Груз 1 → место 1, груз 2 → место 2, затем финиш. Маршрут задан по перекрёсткам:

Участок	Через перекрёстки
Старт → груз 1	А → В → Б
Груз 1 → место 1	Б → В
Место 1 → груз 2	В → Г
Груз 2 → место 2	Г
Место 2 → финиш	Г

Куда поворачивать на каждом перекрёстке — налево, направо или прямо, — определяешь ты. Через некоторые перекрёстки робот проезжает несколько раз. Ветки, которых нет в маршруте, не используются.

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 **три раза** коротким опытом.

Умение	Урок	Прошло из 3
Колёса вперёд, назад и стоп	2.1	
Поворот 90° и разворот 180°	2.2	
Линия и дуги своего поля	3.2	
Перекрёсток и поворот с него	4.1, 4.2	
Остановка перед грузом	5.1	
Захват берёт и отпускает	5.2	

ДОЛЖНО ПОЛУЧИТЬСЯ

Таблица заполнена. Умения, которые прошли меньше трёх раз из трёх, ты чинишь первыми.

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 есть повороты налево, направо и проезд прямо. Каждая встреча с перекрёстком записана отдельной строкой — даже если это тот же перекрёсток. Программа не видит букв: она знает только, какое по счёту решение сейчас выполняет.

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. Когда маршрут записан вызовами функций, его легко сверить с планом.

Заготовка: маршрут по плану L61_Courier_Plan.ino

```
// Робот-курьер, средний уровень: маршрут по плану из журнала.
// Функций здесь нет: перенеси свои проверенные функции из уроков 2.1–5.2 в р
// Маршрут в setup() разбит на участки плана. В каждый участок впиши вызовы ф

// --- Контакты, измерения и настройки из своих скетчей ---

// --- Мои функции (перенеси и проверь каждую отдельно) ---
// Нужны: остановка моторов, прямой проезд, линия до перекрёстка, проезд пере
// прямо, поворот налево и направо на перекрёстке, разворот на линии, подъезд
// взять груз, поставить груз.

void setup() {
  // Подготовь выводы, датчики и захват. Моторы стоят.

  // Участок 1. Старт → линия:

  // Участок 2. Линия → груз 1:

  // Участок 3. Груз 1 → место 1:

  // Участок 4. Место 1 → груз 2:

  // Участок 5. Груз 2 → место 2:

  // Участок 6. Место 2 → финиш, стоп:
}

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

1. Открой заготовку и сохрани её под своим именем. Готовых функций в ней нет.
2. Перенеси свои проверенные функции из уроков 2.1–5.2. Проверь по списку в заготовке, что ни одной не забыл.

3. Перед каждой строкой плана в журнале запиши, какая функция её выполняет. Если для строки нет функции — напиши её и проверь отдельно.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Все нужные функции перенесены и проверены, у каждой строки плана есть своя функция.

★★ **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. Запиши в журнал вызовы функций первого участка по плану: выезд, линия, перекрёстки, повороты, подъезд к грузу. Потом перенеси их в `setup()`.
2. Проверь на поле только эту часть **три раза**. Робот должен каждый раз остановиться перед грузом 1.
3. Работает — сохрани копию: «рабочая версия 1».

Подсказка: Где искать ошибку

Сравни, что сделал робот, со строкой плана. На какой строке разошлось? Проверь эту одну функцию отдельно.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Робот три раза подряд выезжает со старта и останавливается перед грузом 1.

7 Часть 2: первая доставка

12 мин

1. Запиши в журнал вызовы функций первой доставки: захват, разворот, путь к месту 1, отпускание, возвращение к дороге. Потом допиши их в программу.
2. Проверь только эту часть три раза: ставь робота перед грузом 1 и запускай.
3. Работает — сохрани «рабочую версию 2».

Подсказка: Порядок с захватом

Вспомни правило урока 5.2: остановиться — закрыть; открыть — подождать — отъехать.

ДОЛЖНО ПОЛУЧИТЬСЯ

Три раза подряд груз 1 стоит в месте 1, 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 по плану. Груз 2 стоит в другом месте, поэтому повороты будут другими. Каждую встречу с перекрёстком — отдельной строкой.
2. В конце — прямой въезд в финиш и остановка. После финиша робот должен стоять.
3. Запусти весь маршрут от старта. Если сорвалось — запусти с места срыва только оставшуюся часть. Сохрани «рабочую версию 3».

Подсказка: Как понять, что пора въезжать в финиш

Чёрной метки финиша на поле нет. Используй последний перекрёсток и проверенное время прямого движения.

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.