Тема 5 · Урок 5.2

Захват: берём и ставим

Робот подъезжает к грузу, берёт его захватом, увозит и ставит в нужном месте.

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

РЕЗУЛЬТАТ УРОКА

Робот подъезжает к грузу по ультразвуку, берёт её, разворачивается, везёт по линии и ставит на другом конце. Действия оформлены функциями pickUp() и putDown().

Ты научишься

- Объяснить, как сервопривод поворачивается на заданный угол.
- Найти углы «открыто» и «закрыто» для своего захвата опытом.
- Соблюдать порядок: остановиться — закрыть; открыть — подождать — отъехать.
- Собрать перевозку груза из функций прошлых уроков.

Нужно знать

- 5.1 Ультразвук: видим груз
- 4.2 Повороты на перекрёстке и маршрут

Понадобится

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

Зачем этот урок

Робот умеет доезжать до груза. Остался последний навык курьера — взять груз, довезти и аккуратно поставить. Сегодня робот соберёт всё вместе: линия, ультразвук и захват.

РАЗМИНКА · ВСПОМНИ

1. Какое условие остановки перед грузом правильное и почему?
2. Как робот разворачивается на линии?
3. Что делает `followLineFor(1500)`?

БЕЗОПАСНОСТЬ

Не держи пальцы в захвате, когда включено питание. Сервопривод может резко повернуться при загрузке программы.

ЗАНЯТИЕ 1 · 45 МИН

1 Сервопривод и захват

1 Как работает сервопривод

8 мин

Что должен делать робот. Курьер должен взять груз, удержать его при перевозке и отпустить в нужном месте. Для этого добавим механический захват с сервоприводом. Ультразвук помогает подъехать, но сам ничего не захватывает.

Что используем и как это работает. Внутри сервопривода есть мотор, редуктор, датчик положения и управляющая схема. По управляющим импульсам он получает заданное положение вала и старается его удержать. Библиотека Servo формирует эти импульсы; gripper.write задаёт положение, а не скорость вращения. Рычаг превращает поворот вала в движение губок захвата.

Как связать измерение с действием. Углы «открыто» и «закрыто» подбираем для своей механики небольшими изменениями; чужие числа могут упереть механизм в ограничитель. Сначала останавливаем колёса, затем закрываем захват и даём ему время. Команда закрытия не доказывает, что груз удерживается: это проверяем наблюдением. Подключение и питание сверяем с подтверждённой схемой комплекта.

Обычный мотор крутится, пока включён. **Сервопривод** поворачивается на заданный угол и держит его. Командуем углом в градусах: `gripper.write(90)`.

ПРЕДСКАЖИ

Что сделает сервопривод после `gripper.write(90)`, а потом `gripper.write(40)`? А мотор колеса после `analogWrite(…, 90)`?

1. Запиши в журнал: чем команда сервоприводу отличается от команды мотору колеса — что задаёт число и когда движение заканчивается.

ДОЛЖНО ПОЛУЧИТЬСЯ

Сравнение сервопривода и мотора колеса записано.

2 Найди углы «открыто» и «закрыто»

22 мин

У каждого захвата свои углы. Найди их опытом — способ выбери сам, но меняй только `angle` и записывай каждую попытку.

Команда	Что делает
`gripper.attach(контакт);`	Связывает сервопривод с контактом
`gripper.write(угол);`	Поворачивает на угол 0...180 и держит его

1. Создай скетч `L52_Servo_Test` по шаблону. Посмотри контакт сервопривода на карточке у робота и впиши его вместо `-1`.
2. Допиши `setup()` сам: захват должен повернуться на `angle`.

Шаблон: проба угла L52_Servo_Test_Gaps.ino

```
// Шаблон среднего уровня. Напиши код в разделе «Твоя часть» по целям в комме

#include <Servo.h>

// Контакт сервопривода захвата: впиши номер из таблицы проводки своего робот
const int SERVO_PIN = -1;
int angle = 90; // меняй понемногу, чтобы найти «открыто» и «закрыто»

Servo gripper;

// ===== Твоя часть: напиши сам =====

void setup() {
  if (SERVO_PIN < 0) {
    while (true) {
    }
  }

  // Захват поворачивается на угол angle и держит его.
}

void loop() {
}
```

3. Найди угол «открыто»: груз свободно входит в захват.
4. Найди угол «закрыто»: груз держится крепко, а сервопривод не гудит.
5. Проверь «закрыто»: подними робота с грузом и слегка наклони. Груз не выпадает?

Попытка	`angle`	Что видно	Что поменять и почему
1	90		
2			
3			
4			

Подсказка: Сервопривод гудит или дрожит

Захват упёрся и не может дойти до угла. Что нужно сделать с углом?

ДОЛЖНО ПОЛУЧИТЬСЯ

В журнале — попытки с объяснением, углы «открыто» и «закрыто» и проверка наклоном.

3 Порядок действий важен

15 мин

1. До того как смотреть код, запиши в журнал по шагам: как роботу правильно **взять** груз и как **поставить**. Что должно случиться раньше: остановка или захват? Открытие или отъезд?
2. Сравни свой порядок с правилом:

- **Взять:** сначала остановиться, потом закрыть захват.
- **Поставить:** открыть захват, подождать, пока он раскроется, и только потом отъехать.

НАЙДИ ОШИБКУ · ГРУЗ УЕЗЖАЕТ С РОБОТОМ

У соседей `putDown()` такая: `gripper.write(OPEN_ANGLE);` и сразу `drive(BACKWARD, …)`. Груз то падает, то уезжает вместе с роботом. Почему? Как исправить?

ДОЛЖНО ПОЛУЧИТЬСЯ

Твой порядок записан и сверен с правилом, ошибка соседей объяснена.

2

ЗАНЯТИЕ 2 · 45 МИН

Перевозка груза

4 Взять и поставить

20 мин

1. Создай скетч `L52_Pick_Place` по шаблону. Перенеси измерения, `kp`, `baseSpeed`, контакты датчика и сервопривода, свои углы. Езда по линии, перекрёстки и подъезд к грузу уже готовы.
2. Напиши сам подготовку захвата `setupGripper()`, `pickUp()`, `putDown()` и доставку в `setup()` — по своим правилам порядка из прошлого шага.

Шаблон: взять и поставить L52_Pick_Place_Gaps.ino

```
// Шаблон среднего уровня. Напиши код в разделе «Твоя часть» по целям в комме

#include <Servo.h>
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;

// Контакты ультразвукового датчика: впиши номера из таблицы проводки своего
const int TRIG_PIN = -1;
const int ECHO_PIN = -1;
int turnSpeed = 100;
int centerMs = 250; // подбираем: проехать от датчиков до центра перекрёстка
int leaveMs = 200;  // время, чтобы датчик ушёл со старой линии

// Сервопривод захвата: впиши контакт и свои углы из журнала.
const int SERVO_PIN = -1;
const int OPEN_ANGLE = 90;
const int CLOSED_ANGLE = 40;
const int STOP_CM = 5;

Servo gripper;
const unsigned long CARRY_MS = 1500; // сколько ехать по линии с грузом

// ===== Твоя часть: напиши сам =====

void setupGripper() {
```

```
  // Без контакта сервопривода программа дальше не идёт.
  // Захват подключён и открыт; дай ему время раскрыться.
}

void pickUp() {
  // Взять груз: груз оказывается в захвате и не выталкивается.
}

void putDown() {
  // Поставить груз: груз остаётся на месте, когда робот отъезжает назад на 3
}

void setup() {
  setupSonar();
  setupGripper();
  setupRobot();

  // Доставка: подъехать к грузу, взять, развернуться на линии, проехать с гр
}

// ===== Готовое из уроков 3.2–5.1 — 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);
}
```

```
85
86  void followLineStep() {
87    int error = leftLevel - rightLevel;
88    int correction = kp * error;
89    drive(FORWARD, constrain(baseSpeed + correction, 0, 255),
90          FORWARD, constrain(baseSpeed - correction, 0, 255));
91  }
92
93  void setupRobot() {
94    pinMode(LEFT_DIR_PIN, OUTPUT);
95    pinMode(LEFT_SPEED_PIN, OUTPUT);
96    pinMode(RIGHT_DIR_PIN, OUTPUT);
97    pinMode(RIGHT_SPEED_PIN, OUTPUT);
98    stopMotors();
99    // Без измерений робот не поедет.
100   if (LEFT_WHITE == LEFT_BLACK || RIGHT_WHITE == RIGHT_BLACK) {
101     while (true) {
102     }
103   }
104   delay(2000);
105 }
106
107 // Расстояние в сантиметрах или -1, если эхо не вернулось.
108 int readDistanceCm() {
109   digitalWrite(TRIG_PIN, LOW);
110   delayMicroseconds(2);
111   digitalWrite(TRIG_PIN, HIGH);
112   delayMicroseconds(10);
113   digitalWrite(TRIG_PIN, LOW);
114   unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000);
115   if (echoUs == 0) {
116     return -1;
117   }
118   return echoUs / 58;
119 }
120
121 void setupSonar() {
122   // Без контактов датчика программа дальше не пойдёт.
123   if (TRIG_PIN < 0 || ECHO_PIN < 0) {
124     while (true) {
125     }
126   }
127   pinMode(TRIG_PIN, OUTPUT);
```

```
  pinMode(ECHO_PIN, INPUT);
}

// Выехать на центр перекрёстка: датчики стоят впереди колёс.
void toCenter() {
  drive(FORWARD, baseSpeed, FORWARD, baseSpeed);
  delay(centerMs);
  stopMotors();
}

// Крутиться на месте, пока датчик со стороны поворота не найдёт новую линию.
void turnUntilLine(bool left) {
  if (left) {
    drive(BACKWARD, turnSpeed, FORWARD, turnSpeed);
  } else {
    drive(FORWARD, turnSpeed, BACKWARD, turnSpeed);
  }
  delay(leaveMs);
  do {
    readSensors();
  } while ((left ? leftLevel : rightLevel) > 30);
  stopMotors();
}

void turnLeftAtCrossing() {
  toCenter();
  turnUntilLine(true);
}

void turnRightAtCrossing() {
  toCenter();
  turnUntilLine(false);
}

// Развернуться на линии: крутиться, пока линия снова не окажется под датчико
void turnAroundOnLine() {
  turnUntilLine(true);
}

// Ехать по линии заданное время.
void followLineFor(unsigned long ms) {
  unsigned long start = millis();
  while (millis() - start < ms) {
```

```
    readSensors();
    followLineStep();
  }
  stopMotors();
}

// Ехать по линии и остановиться перед грузом по ультразвуку.
void approachCan() {
  while (true) {
    int distance = readDistanceCm();
    if (distance > 0 && distance <= STOP_CM) {
      break;
    }
    readSensors();
    followLineStep();
  }
  stopMotors();
}

void loop() {
}
```

ПРЕДСКАЖИ

Прочитай свой `setup()` и расскажи напарнику по шагам, что сделает робот.

3. Поставь груз в конце линии, робота — на другом конце. Запусти.

Подсказка: Сколько ждать

Захвату нужно время, чтобы дойти до угла. Попробуй паузу 300 и 600 мс: успевает ли он?

Подсказка: Без контакта

Защиту без контакта ты уже видел в скетче пробы угла.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот подъезжает, берёт груз, разворачивается, везёт его по линии и ставит.

5 Надёжность

12 мин

1. Повтори перевозку пять раз подряд. Каждую неудачу запиши: на каком этапе и что случилось.

Заезд	Подъезд	Захват	Перевозка	Постановка	Что случилось
1					
2					
3					
4					
5					

2. Найди этап, где ошибок больше всего. Какое одно число поменять, чтобы это исправить? Поменяй и повтори три раза.

Подсказка: Какие числа есть

`STOP_CM` — где робот встаёт перед грузом, `CARRY_MS` — где ставит, углы захвата — как держит.

ДОЛЖНО ПОЛУЧИТЬСЯ

Таблица из пяти заездов, исправленный этап и три перевозки подряд без помощи рук.

6 Задания по уровням

13 мин

★ **СРЕДНИЙ УРОВЕНЬ · ЧЕРЕЗ ПЕРЕКРЁСТОК**

Груз стоит на одной ветке, место доставки — на другой. Перевези груз через перекрёсток, используя функции урока 4.2.

ПРОВЕРЬ СЕБЯ

- [ ] Мои углы «открыто» и «закрыто» записаны в журнале.
- [ ] Я соблюдаю порядок: остановиться — закрыть; открыть — подождать — отъехать.
- [ ] Робот перевозит груз без помощи рук.

Проверь себя

★ Средний уровень

1. Груз стоит на одной ветке, место доставки — на другой. Что нужно добавить к перевозке по линии?

- ○ Решение на перекрёстке из урока 4.2
- ○ Второй захват
- ○ Медиану измерений

2. В каком порядке выполняется доставка?

- ○ Подъехать к грузу, закрыть захват, доехать до места, открыть захват
- ○ Закрыть захват, подъехать к грузу, открыть захват
- ○ Открыть захват на месте, потом подъехать к грузу

3. Функция pickUp() осталась пустой. Что сделает робот у груза?

- ○ Развернётся и уедет без груза
- ○ Будет стоять у груза
- ○ Возьмёт груз как обычно

4. При проверке наклоном груз выпадает из захвата. Что поменять?

- ○ CARRY_MS
- ○ Угол «закрыто»: захват держит слишком слабо
- ○ STOP_CM

5. Три заезда из пяти сорвались на постановке: груз ставится мимо места. Какое одно число менять первым?

- ○ OPEN_ANGLE
- ○ kp
- ○ CARRY_MS — сколько робот едет с грузом

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 робота есть все умения итоговой задачи: ехать, поворачивать, держаться линии, находить перекрёстки, видеть груз и перевозить его. Дальше — итоговый проект.

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

сервопривод	Мотор, который поворачивается на заданный угол и держит его.
угол	Положение рычага сервопривода в градусах, обычно от 0 до 180.
библиотека	Готовый набор команд, который подключают строкой #include. Servo — библиотека для сервоприводов.