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. Внутри у него мотор, шестерёнки и датчик положения. Командуем углом в градусах: `gripper.write(90)`.

ПРЕДСКАЖИ

Что сделает сервопривод после `gripper.write(90)`, а потом `gripper.write(40)`?

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить, чем сервопривод отличается от мотора колеса.

2 Найди углы «открыто» и «закрыто»

22 мин

У каждого захвата свои углы. Найдём их опытом, как время поворота в уроке 2.2.

1. Посмотри контакт сервопривода на карточке у робота и впиши его вместо `-1`.

Проба угла L52_Servo_Test.ino

```
#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;

void setup() {
  if (SERVO_PIN < 0) {
    while (true) {
    }
  }
  gripper.attach(SERVO_PIN);
  gripper.write(angle);
}

void loop() {
}
```

2. Начни с `angle = 90`. Загрузи. Меняй угол по 10°: найди положение, где захват открыт так, что груз свободно входит.
3. Найди угол, где захват держит груз крепко, но не давит изо всех сил.
4. Запиши оба угла в журнал.

Подсказка: Сервопривод гудит или дрожит

Захват упёрся в груз или в собственный упор и не может дойти до угла. Отступи на 5–10° от этого положения: сервопривод не должен работать в упор.

ДОЛЖНО ПОЛУЧИТЬСЯ

В журнале записаны углы «открыто» и «закрыто», при которых груз свободно входит и надёжно держится.

3 Порядок действий важен

15 мин

Сопоставь функции `pickUp()` и `putDown()` в итоговом скетче с правилом порядка (следующее занятие):

- **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

Ты можешь объяснить, зачем пауза после открытия захвата.

2

ЗАНЯТИЕ 2 · 45 МИН

Перевозка груза

4 Взять и поставить

20 мин

1. Создай скетч `L52_Pick_Place`. Перенеси измерения, `kp`, `baseSpeed`, контакты датчика и сервопривода, свои углы.

Взять и поставить L52_Pick_Place.ino

```
#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;

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);
}

// Контакты ультразвукового датчика: впиши номера из таблицы проводки своего
const int TRIG_PIN = -1;
const int ECHO_PIN = -1;

// Расстояние в сантиметрах или -1, если эхо не вернулось.
int readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000);
  if (echoUs == 0) {
    return -1;
  }
  return echoUs / 58;
}

void setupSonar() {
```

```
  // Без контактов датчика программа дальше не пойдёт.
  if (TRIG_PIN < 0 || ECHO_PIN < 0) {
    while (true) {
    }
  }
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

int turnSpeed = 100;
int centerMs = 250; // подбираем: проехать от датчиков до центра перекрёстка
int leaveMs = 200;  // время, чтобы датчик ушёл со старой линии

// 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);
}
```

```
128
129 // Развернуться на линии: крутиться, пока линия снова не окажется под датчико
130 void turnAroundOnLine() {
131   turnUntilLine(true);
132 }
133
134 // Ехать по линии заданное время.
135 void followLineFor(unsigned long ms) {
136   unsigned long start = millis();
137   while (millis() - start < ms) {
138     readSensors();
139     followLineStep();
140   }
141   stopMotors();
142 }
143
144 // Сервопривод захвата: впиши контакт и свои углы из журнала.
145 const int SERVO_PIN = -1;
146 const int OPEN_ANGLE = 90;
147 const int CLOSED_ANGLE = 40;
148 const int STOP_CM = 5;
149 Servo gripper;
150
151 void setupGripper() {
152   // Без контакта сервопривода программа дальше не пойдёт.
153   if (SERVO_PIN < 0) {
154     while (true) {
155     }
156   }
157   gripper.attach(SERVO_PIN);
158   gripper.write(OPEN_ANGLE);
159   delay(500);
160 }
161
162 // Ехать по линии и остановиться перед грузом по ультразвуку.
163 void approachCan() {
164   while (true) {
165     int distance = readDistanceCm();
166     if (distance > 0 && distance <= STOP_CM) {
167       break;
168     }
169     readSensors();
170     followLineStep();
```

```
  }
  stopMotors();
}

// Сначала остановиться, потом закрыть захват.
void pickUp() {
  stopMotors();
  gripper.write(CLOSED_ANGLE);
  delay(600);
}

// Открыть захват, подождать и только потом отъехать назад.
void putDown() {
  stopMotors();
  gripper.write(OPEN_ANGLE);
  delay(600);
  drive(BACKWARD, baseSpeed, BACKWARD, baseSpeed);
  delay(300);
  stopMotors();
}

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

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

  approachCan();
  pickUp();
  turnAroundOnLine();
  followLineFor(CARRY_MS);
  putDown();
}

void loop() {
}
```

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

ПРЕДСКАЖИ

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

ДОЛЖНО ПОЛУЧИТЬСЯ

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

5 Надёжность

12 мин

1. Повтори перевозку три раза подряд. Где робот ошибается чаще всего?
2. Подбери `STOP_CM` так, чтобы груз оказывался в середине захвата.
3. Подбери `CARRY_MS`, чтобы груз ставился в нужном месте.

ДОЛЖНО ПОЛУЧИТЬСЯ

Три перевозки подряд без помощи рук. Значения записаны в журнал.

6 Задания по уровням

13 мин

ОСНОВНОЕ · ПЕРЕВОЗКА ПО ЛИНИИ

Взять груз с конца линии и поставить его на другом конце той же линии.

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

Основное

1. Чем сервопривод отличается от мотора колеса?

- ○ Поворачивается на заданный угол и держит его
- ○ Крутится быстрее
- ○ Работает без питания

2. В каком порядке правильно взять груз?

- ○ Закрыть захват на ходу, потом остановиться
- ○ Остановиться, потом закрыть захват
- ○ Отъехать назад и закрыть захват

3. Зачем в putDown() пауза после открытия захвата?

- ○ Чтобы захват успел раскрыться и не утащил груз
- ○ Чтобы зарядилась батарея
- ○ Пауза не нужна

4. Сервопривод гудит, захват дрожит у груза. Что сделать?

- ○ Поставить угол «закрыто» на 5–10° мягче, чтобы не давить в упор
- ○ Увеличить скорость робота
- ○ Убрать delay

5. Почему углы захвата подбирают опытом?

- ○ Они зависят от конструкции захвата и размера груза
- ○ Углы у всех сервоприводов разные каждый день
- ○ Так требует библиотека Servo

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.