Тема 5 · Урок 5.1

Ультразвук: видим груз

Робот измеряет расстояние звуком и останавливается перед грузом, не путая «нет эха» с «груз рядом».

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

РЕЗУЛЬТАТ УРОКА

Робот едет по линии и останавливается на заданном расстоянии от груза. Если эхо не вернулось, программа не принимает это за «груз рядом».

Ты научишься

- Объяснить, как ультразвуковой датчик измеряет расстояние.
- Вывести расстояние в монитор порта и проверить его линейкой.
- Отличить отсутствие эха от маленького расстояния.
- Остановить робота перед грузом по измерению, а не по времени.

Нужно знать

- 3.2 П-регулятор: плавно и быстро

Понадобится

Ультразвуковой датчик расстояния | Датчик линии (аналоговый, Trema) × 2

Зачем этот урок

Груз стоит в конце ответвления. Робот должен подъехать к ней и остановиться ровно так, чтобы захват мог её взять. Если ехать «по времени», робот то не доедет, то врежется. Сегодня он научится видеть груз ультразвуком — так же, как летучая мышь видит препятствия.

РАЗМИНКА · ВСПОМНИ

1. Как робот останавливается на перекрёстке в программе из урока 4.1?
2. Что делает `break` в цикле? (Если не помнишь — разберём сегодня.)
3. Какие две команды нужны, чтобы робот ехал по линии с П-регулятором?

1

ЗАНЯТИЕ 1 · 45 МИН

Измеряем расстояние

1 Как работает ультразвук

8 мин

Что должен делать робот. Курьеру нужно обнаружить предмет впереди, оценить расстояние до него и остановиться перед грузом. Датчики линии смотрят вниз и эту задачу не решают. Добавим ультразвуковой датчик: он поможет измерить расстояние, но не отличит груз от стены.

Что используем и как это работает. Передатчик датчика излучает короткую серию ультразвуковых колебаний, которые человек не слышит. Звук отражается от предмета, а приёмник улавливает эхо. Arduino запускает измерение импульсом на TRIG и измеряет длительность сигнала ECHO. Это электрический сигнал о времени измерения, а не сам звук. Путь звука двойной: до предмета и обратно. Поэтому расстояние равно скорости звука, умноженной на время и делённой на два.

Как связать измерение с действием. В учебной программе время в микросекундах делится на 58, чтобы получить приблизительные сантиметры: около 29 мкс на сантиметр в одну сторону и столько же обратно. Например, 580 мкс соответствует примерно 10 см. Целочисленный результат отбрасывает дробную часть; температура и отражающая поверхность тоже влияют на измерение. Сравниваем его с линейкой от передней поверхности датчика.

Если измерение не получилось. Наклонная поверхность может отразить звук в сторону, мягкая — поглотить, а слишком близкий или далёкий предмет — не дать пригодного эха. Поэтому «нет эха» означает «расстояние неизвестно», а не ноль и не свободный путь. В коде для этого есть отдельное значение −1. Программа принимает решение об остановке только после проверки измерения. Базовый пример при −1 продолжает движение — это ограничение учебного алгоритма, которое разберём в задании на потерю эха.

Датчик посылает короткий щелчок ультразвука и слушает эхо. Чем дальше предмет, тем дольше звук летит туда и обратно. По времени полёта считаем расстояние.

ПРЕДСКАЖИ

Звук пролетает 1 см примерно за 29 микросекунд. Предмет в 10 см. Сколько времени пройдёт до возвращения эха? (Не забудь: туда и обратно.)

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить: время эха → расстояние. До предмета в 10 см и обратно — около 580 мкс.

2 Расстояние в мониторе порта

20 мин

1. Посмотри номера контактов TRIG и ECHO на карточке у робота. Карточки нет — проследи провода от выводов TRIG и ECHO датчика до платы и прочитай номера у гнёзд. Запиши в журнал.
2. Создай скетч `L51_Distance` и впиши номера контактов вместо `-1`.

Расстояние в мониторе порта L51_Distance.ino

```ino
// 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);
}

void setup() {
  Serial.begin(9600);
  setupSonar();
}

void loop() {
  int distance = readDistanceCm();
  if (distance < 0) {
    Serial.println("нет эха");
  } else {
    Serial.print(distance);
    Serial.println(" см");
  }
```

```
  delay(200);
}
```

3. Загрузи, открой монитор порта. Ставь груз на 5, 10, 20 и 30 см по линейке. Запиши показания.

По линейке	Показание датчика
5 см	
10 см	
20 см	
30 см	

ДОЛЖНО ПОЛУЧИТЬСЯ

Показания датчика близки к линейке, разница — 1–2 см.

Как это устроено: Как работает readDistanceCm

- На TRIG подаём короткий импульс 10 мкс — датчик щёлкает.
- `pulseIn(ECHO_PIN, HIGH, 25000)` ждёт эхо не дольше 25 000 мкс и возвращает время в микросекундах. Если эхо не вернулось — возвращает 0.
- Делим время на 58: это 29 мкс на сантиметр туда и 29 обратно.
- Если эха нет, функция возвращает −1: «измерения нет».

ЕСЛИ НЕ ПОЛУЧИЛОСЬ

Что видишь	Что сделать
Программа ничего не печатает	Вписаны ли номера TRIG и ECHO вместо −1?
Всегда «нет эха»	Перепутаны TRIG и ECHO или датчик не получает питание
Числа прыгают	Груз должна стоять прямо напротив датчика

3 «Нет эха» — не «рядом»

15 мин

1. Направь датчик в пустоту или на мягкую ткань под углом. Что печатает монитор?
2. Поднеси ладонь вплотную к датчику, ближе 2 см. Что печатает?

НАЙДИ ОШИБКУ · РОБОТ ЗАКРЫЛ ЗАХВАТ В ПУСТОТЕ

Соседи написали остановку так: `if (readDistanceCm() <= 5) { stopMotors(); }`. Робот иногда останавливается посреди пустой дороги. Почему? Что вернёт функция, если эха нет?

ДОЛЖНО ПОЛУЧИТЬСЯ

Ты можешь объяснить, почему условие остановки должно проверять, что расстояние больше нуля.

2

ЗАНЯТИЕ 2 · 45 МИН

Стоп перед грузом

4 Едем к грузу и останавливаемся

17 мин

1. Создай скетч `L51_Stop_At_Can`. Перенеси измерения, `kp`, `baseSpeed` и контакты датчика.

Стоп перед грузом L51_Stop_At_Can.ino

```
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);
}
```

```
42
43  void followLineStep() {
44    int error = leftLevel - rightLevel;
45    int correction = kp * error;
46    drive(FORWARD, constrain(baseSpeed + correction, 0, 255),
47          FORWARD, constrain(baseSpeed - correction, 0, 255));
48  }
49
50  void setupRobot() {
51    pinMode(LEFT_DIR_PIN, OUTPUT);
52    pinMode(LEFT_SPEED_PIN, OUTPUT);
53    pinMode(RIGHT_DIR_PIN, OUTPUT);
54    pinMode(RIGHT_SPEED_PIN, OUTPUT);
55    stopMotors();
56    // Без измерений робот не поедет.
57    if (LEFT_WHITE == LEFT_BLACK || RIGHT_WHITE == RIGHT_BLACK) {
58      while (true) {
59      }
60    }
61    delay(2000);
62  }
63
64  // Контакты ультразвукового датчика: впиши номера из таблицы проводки своего
65  const int TRIG_PIN = -1;
66  const int ECHO_PIN = -1;
67
68  // Расстояние в сантиметрах или -1, если эхо не вернулось.
69  int readDistanceCm() {
70    digitalWrite(TRIG_PIN, LOW);
71    delayMicroseconds(2);
72    digitalWrite(TRIG_PIN, HIGH);
73    delayMicroseconds(10);
74    digitalWrite(TRIG_PIN, LOW);
75    unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000);
76    if (echoUs == 0) {
77      return -1;
78    }
79    return echoUs / 58;
80  }
81
82  void setupSonar() {
83    // Без контактов датчика программа дальше не пойдёт.
84    if (TRIG_PIN < 0 || ECHO_PIN < 0) {
```

```
    while (true) {
    }
  }
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

const int STOP_CM = 5; // на каком расстоянии от груза остановиться

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

  while (true) {
    int distance = readDistanceCm();
    if (distance > 0 && distance <= STOP_CM) {
      break;
    }
    readSensors();
    followLineStep();
  }
  stopMotors();
}

void loop() {
}
```

2. Поставь груз в конце линии. Запусти робота с другого конца.

✓ **ДОЛЖНО ПОЛУЧИТЬСЯ**

Робот едет по линии и останавливается перед грузом, не касаясь его.

ⓘ **Как это устроено: Что делает break**

`break` немедленно выходит из цикла `while`. После цикла программа выполняет `stopMotors()`. Так «стоп по расстоянию» оказывается в одном месте программы.

5 Точная остановка

15 мин

1. Измерь линейкой, на каком расстоянии от груза остановился робот.
2. Подбери `STOP_CM` и `baseSpeed`, чтобы робот останавливался там, где удобно захвату.
3. Повтори три раза. Запиши в журнал `STOP_CM` и фактическое расстояние.

Подсказка: Робот останавливается дальше или ближе, чем STOP_CM

Робот не останавливается мгновенно: он ещё немного катится. Чем выше скорость, тем дальше. Можно перед грузом ехать медленнее или поставить `STOP_CM` чуть больше.

ДОЛЖНО ПОЛУЧИТЬСЯ

Робот три раза подряд останавливается на нужном расстоянии от груза.

6 Задания по уровням

13 мин

ОСНОВНОЕ · СТОП В 5 СМ

Остановиться в 5 см от груза, стоящего в конце линии.

ПРОВЕРЬ СЕБЯ

- [ ] Я могу объяснить, как датчик измеряет расстояние.
- [ ] Я знаю, почему −1 нельзя принимать за «груз рядом».
- [ ] Мой робот останавливается перед грузом по измерению.

Проверь себя

Основное

1. Как ультразвуковой датчик узнаёт расстояние?

- ○ По времени, за которое звук долетает до предмета и возвращается
- ○ По яркости отражённого света
- ○ По громкости эха

2. readDistanceCm() вернула −1. Что это значит?

- ○ Груз вплотную к датчику
- ○ Эхо не вернулось — измерения нет
- ○ Датчик измерил 1 см

3. Какое условие остановки правильное?

- ○ distance <= 5
- ○ distance > 0 && distance <= 5
- ○ distance == 5

4. Робот всегда останавливается на 3 см ближе, чем STOP_CM. Почему?

- ○ После команды остановки робот ещё немного катится
- ○ Датчик врёт
- ○ Не хватает памяти

5. Три измерения: 12, 85, 13. Какая медиана?

- ○ 13
- ○ 36
- ○ 85

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 видит груз и останавливается перед ним по измерению. Осталось научиться его брать — в следующем уроке.

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

ультразвук	Звук такой высоты, что человек его не слышит. Датчик посылает щелчок и ждёт эхо.
эхо	Звук, отразившийся от предмета и вернувшийся к датчику.
pulseIn	Команда, которая измеряет, сколько микросекунд длился сигнал на входе.
мкс	Микросекунда — миллионная доля секунды.
медиана	Среднее по величине из нескольких чисел: из 4, 25 и 5 это 5.