**Урок №9. Работа с ультразвуковым дальномером**

**Вводный блок (5 минут)**

**Цели и задачи урока:**

1. Изучить принципы работы с ультразвуковым дальномером
2. Использовать ультразвуковой дальномер в проекте

**Вспомним прошлые занятия**

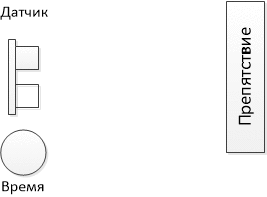
1. Какие компоненты мы уже изучили? Как их подключать?
2. Что такое циклы и условия? В чем их отличие?
3. Какие ключевые слова используются для создания условий и циклов?
4. Как создавать функцию? Какие у нее синтаксические особенности?

**Познание нового (15 минут)**

Дальномер — это устройство для измерения расстояния до некоторого предмета. Дальномер помогает роботам в разных ситуациях. Простой колесный робот может использовать этот прибор для обнаружения препятствий. Летающий дрон использует дальномер для барражирования над землей на заданной высоте. С помощью дальномера можно даже построить карту помещения, применив специальный алгоритм SLAM.

Принцип действия:

На этот раз мы разберем работу одного из самых популярных датчиков — ультразвукового (УЗ) дальномера. Существует много разных модификаций подобных устройств, но все они работают по принципу измерения времени прохождения отраженного звука. То есть датчик отправляет звуковой сигнал в заданном направлении, затем ловит отраженное эхо и вычисляет время полета звука от датчика до препятствия и обратно.

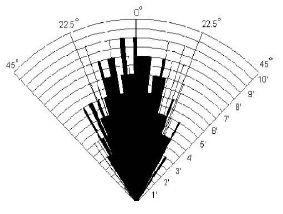


Из школьного курса физики мы знаем, что скорость звука в некоторой среде величина постоянная, но зависящая от плотности среды. Зная скорость звука в воздухе и время полета звука до цели, мы можем рассчитать пройденное звуком расстояние по формуле:

s = v\*t

где v — скорость звука в м/с, а t — время в секундах.  Скорость звука в воздухе, кстати, равна 340.29 м/с.

Чтобы справиться со своей задачей, дальномер имеет две важные конструктивные особенности. Во-первых, чтобы звук хорошо отражался от препятствий, датчик испускает ультразвук с частотой 40 кГц. Для этого в датчике имеется пьезокерамический излучатель, который способен генерировать звук такой высокой частоты. Во-вторых, излучатель устроен таким образом, что звук распространяется не во все стороны (как это бывает у обычных динамиков), а в узком направлении. На рисунке представлена диаграмма направленности типичного УЗ дальномера.



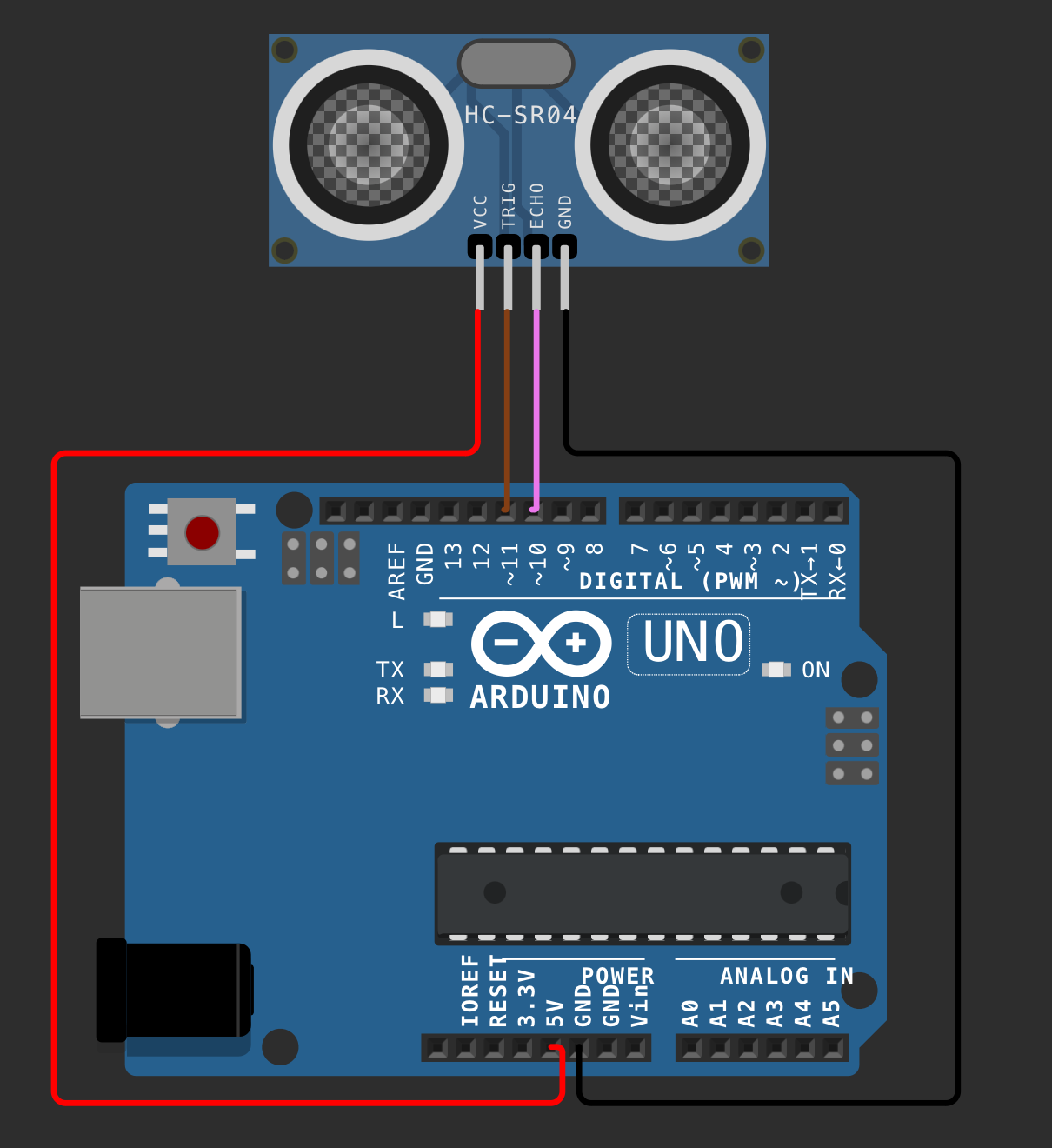
Как видно на диаграмме, угол обзора самого простого УЗ дальномера составляет примерно 50-60 градусов. Для типичного варианта использования, когда датчик детектирует препятствия перед собой, такой угол обзора вполне пригоден. Ультразвук сможет обнаружить даже ножку стула, тогда как лазерный дальномер, к примеру, может её не заметить.

**Работа над проектом (30 минут)**

Тема: Создание нового проекта и написание программы, которая определяет расстояние до объекта с индикацией опасности в виде светодиодов

[**https://wokwi.com/projects/404638775830324225**](https://wokwi.com/projects/404638775830324225)

Подключим ультразвуковой дальномер к Arduino.



Как можно заметить, из звукового дальномера выходит 4 контакта, 2 из которых подключаются к 5В (VCC) и GND (GND).

Остальные два контакта необходимы для считывания показаний

Ультразвуковой датчик работает по принципу отправки и принятия данных

То есть, сначала с trig отправляется сигнал, который должен оттолкнуться от стены или другого препятствия и приниматься на контакт echo. Полученное время и будет являться расстоянием до объекта. Расстояние нужно поделить на 2, чтобы найти только расстояние до объекта.

Инициализируем trig на отправку информации, а echo на принятие информации:

const int TRIG = 11;

const int ECHO = 10;

void setup() {

pinMode(TRIG, OUTPUT);

pinMode(ECHO. INPUT);

Serial.begin(9600);

}

Далее, напишем функцию getDistance для считывания расстояния до объекта:

Int getDistance(){

}

Функция имеет тип **int**. Это означает, что функция будет возвращать результат в виде числа с помощью **return.**

Создадим переменные **duration** и **cm** для определения времени и расстояния.

Для этого подается краткосрочный импульс с trig, то есть сначала там должен быть сигнал LOW, потом HIGH, потом снова LOW.

int getDistance() {

int duration, cm;

digitalWrite(TRIG, LOW);

delayMicroseconds(2);

digitalWrite(TRIG, HIGH);

delayMicroseconds(2);

digitalWrite(TRIG, LOW);

}

*delayMicroseconds* - задержка в микросекундах

Далее получаем сигнал с ECHO с помощью функции PulseIn и переведем полученный сигнал в сантиметры. У функции PulseIn есть два параметра:

номер контакта (ECHO)

активный сигнал (HIGH)

int getDistance() {

int duration, cm;

digitalWrite(TRIG, LOW);

delayMicroseconds(2);

digitalWrite(TRIG, HIGH);

delayMicroseconds(2);

digitalWrite(TRIG, LOW);

duration = pulseIn(ECHO, HIGH);

cm = duration / 58;

}

Возникает вопрос, откуда взялось число 58 и зачем оно используется для конвертирования в сантиметры?

Функция pulseIn замеряет длину положительного импульса на контакте echoPin в микросекундах. В программе мы записываем время полета звука в переменную duration. Поскольку это ультразвуковой дальномер, то в качестве скорости берется скорость звука и нам потребуется умножить время на скорость звука:

**s = duration \* v = duration \* 340 м/с**

Переводим скорость звука из м/с в см/мкс:

**s = duration \* 0.034 cм/мкс**

Для удобства преобразуем десятичную дробь в обыкновенную:

**s = duration \* 1/29 = duration / 29**

А теперь вспомним, что звук прошел два искомых расстояния: до цели и обратно. Поделим всё на 2:

**s = duration / 58**

Теперь мы знаем, откуда взялось число 58 в программе, осталось только вернуть полученный результат из функции и вызывать функцию:

int getDistance() {

int duration, cm;

digitalWrite(TRIG, LOW);

delayMicroseconds(2);

digitalWrite(TRIG, HIGH);

delayMicroseconds(2);

digitalWrite(TRIG, LOW);

duration = pulseIn(ECHO, HIGH);

cm = duration / 58;

**return cm;**

}

**void loop() {**

**int distance = getDistance();**

**Serial.println(distance);**

**delay(100);**

**}**

При запуске скетча и при нажатии на ультразвуковой дальномер в верхнем углу отображения схемы появится ползунок, который можно изменять.

Можно сравнить полученные данные с данными, которые выставляются. Как можно заметить, есть небольшая погрешность в 1-3 см, что является нормальным явлением.

**Некомпьютерная активность (15 минут)**

Расстояние до объекта

*Материалы*:

- Объекты разной дальности (например, стулья, двери, деревья)

- Бумага и ручки для записей

*Ход активности:*

1. Начните с объяснения ученикам, что определение расстояния до объектов - важный навык, который используется во многих областях, от повседневной жизни до научных исследований.

2. Разделите учеников на небольшие группы и выведите их в пространство, где есть различные объекты на разном расстоянии.

3. Предложите ученикам попробовать определить расстояние до выбранных объектов, не используя электронные устройства. Объясните, что для этого они могут использовать собственные наблюдения.

4. Попросите учеников записывать свои измерения и наблюдения в тетради. Например:

- Объект: дверь

- Расстояние: 5 метров

- Способ определения: "Отмерил 5 шагов по 1 метру каждый"

5. Затем предложите ученикам сравнить свои оценки расстояния друг с другом.

6. Обсудите с учениками, какие методы они использовали для определения расстояния и как можно повысить точность своих оценок.

**Работа над проектом (20 минут)**

Создадим схему, которая позволяет отслеживать расстояние до объекта с сигнализацией в виде светодиода.

Для реализации задания необходимо:

1. Подключить 2 светодиода - красный и зеленый
2. Реализовать следующий алгоритм: Если расстояние больше 20 см - светится зеленый, иначе начинает светится красный, а если расстояние будет меньше 5 см, то красный светодиод начнет мигать

[**https://wokwi.com/projects/404639131401414657**](https://wokwi.com/projects/404639131401414657)

**Рефлексия (5 минут)**

"Что нового вы узнали сегодня?"

"Что такое дальномер?"

“Где дальномер может использоваться? Какие роботы точно используют дальномеры”?