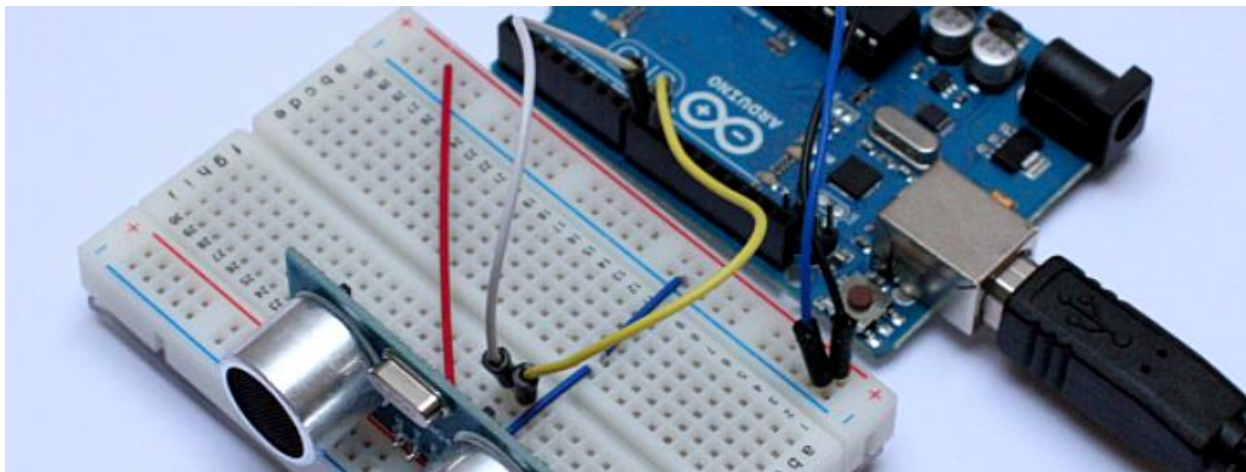
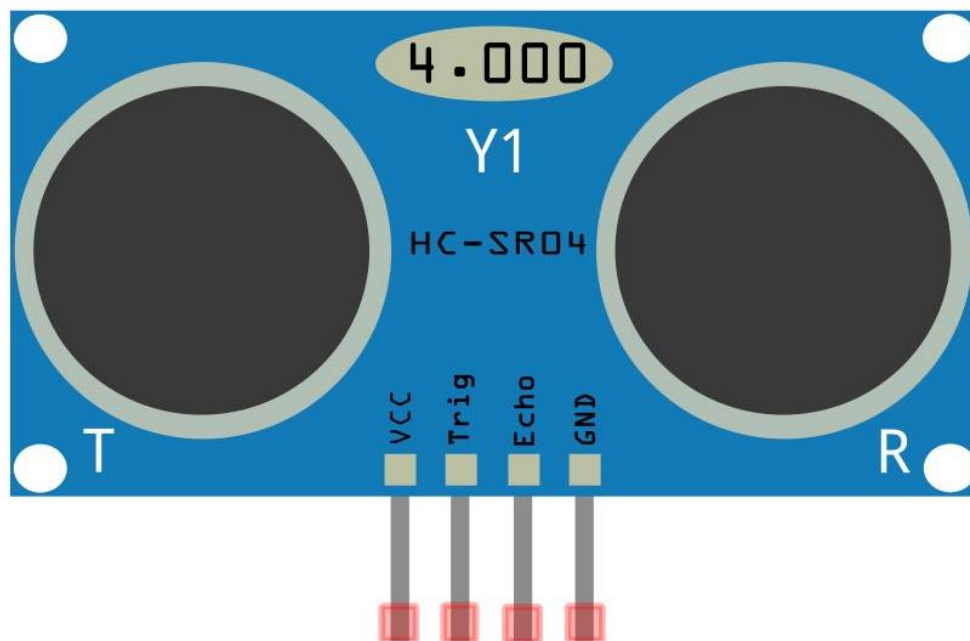


# Ультразвуковой дальномер HC-SR04 подключение к Arduino



В этой статье мы научимся работать с ультразвуковым дальномером HC-SR04. Как работает ультразвуковой дальномер? Устройство испускает ультразвук, который отражается от препятствий и в виде отражённой волны возвращается обратно. Измерив возвращения звука и зная скорость распространения звука, можно очень просто посчитать пройденное им расстояние. Модель дальномера HC-SR04 является самой простой и недорогой, однако обладает хорошими техническими характеристиками (измеряемый диапазон расстояний от 2 до 450 см). Именно поэтому данный модуль можно смело рекомендовать радиолюбителям.



Описание принципа работы ультразвукового дальномера HC-SR04 можно разделить на следующие шаги:

**Шаг 1:** На вход **Trig** подаётся импульс длительностью 10 микроСекунд. Для дальномера это команда начать измерение расстояния перед ним.

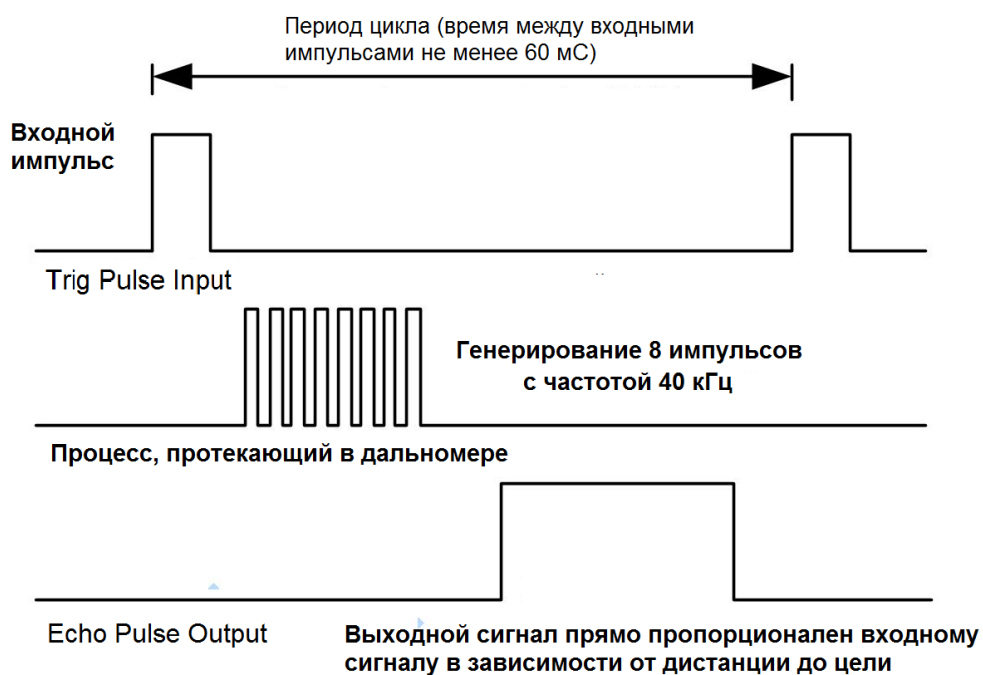
**Шаг 2:** Устройство генерирует 8 ультразвуковых импульсов с частотой 40 кГц через выходной сенсор Т.

**Шаг 3:** Звуковая волна отражается от препятствия и попадает в принимающий сенсор R.

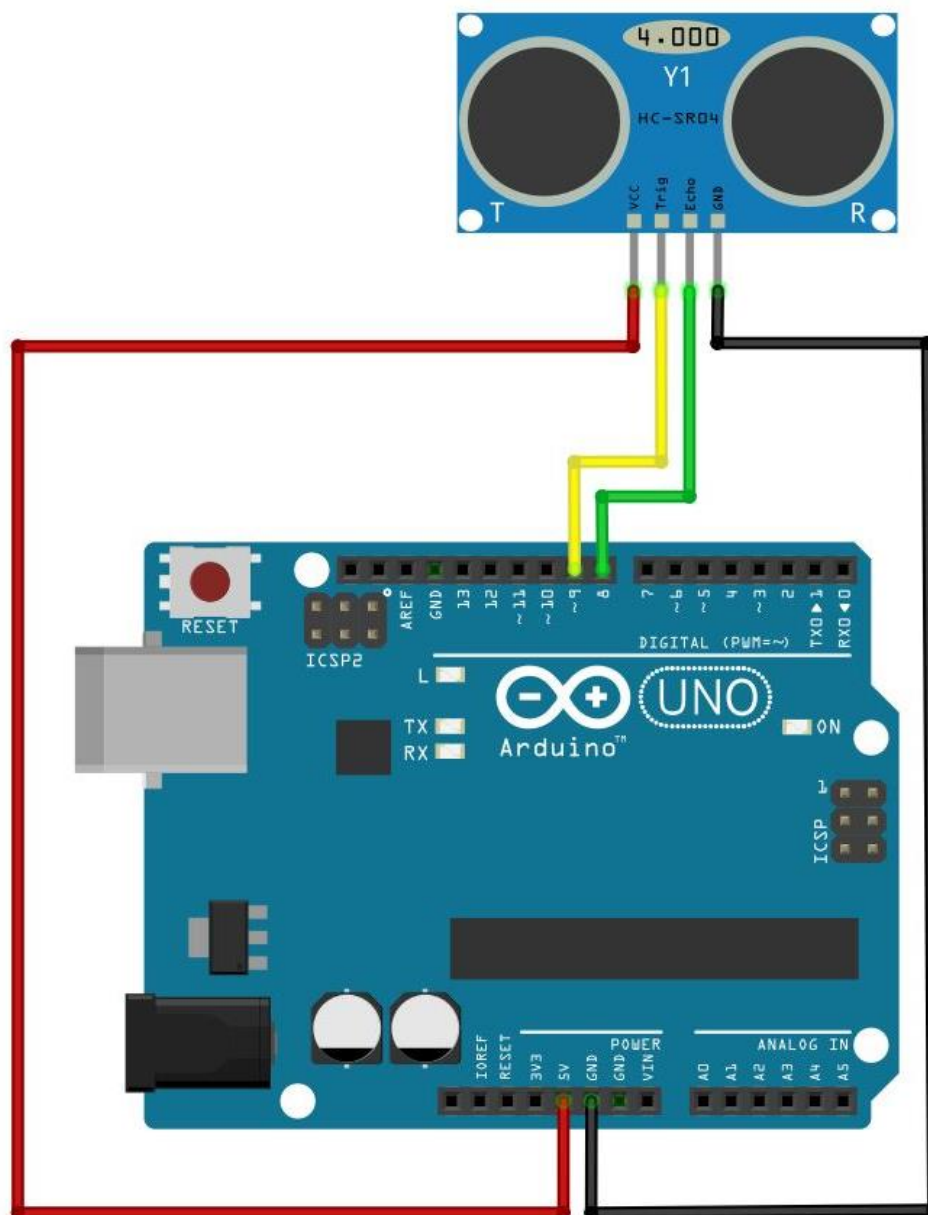
**Шаг 4:** На выходе **Echo** формируется импульс, длительность которого прямо пропорциональна измеренному расстоянию.

**Шаг 5:** На стороне управляющего контроллера переводим длительность импульса **Echo** в расстояние по формуле:  $\text{ширина импульса(мкс)} / 58 = \text{дистанция (см)}$ .

Ниже на рисунке приведены временные диаграммы, наглядно поясняющие перечисленные шаги.



Теперь, когда мы разобрались с устройством ультразвукового дальномера HC-SR04, подключим его к нашему Arduino по следующей схеме:



Собрав схему, нам остаётся только залить скетч для работы. Ниже приведён пример такого скетча. В нём информация о расстоянии будет выводиться в COM порт компьютера, а также при измеренной дистанции менее 30 см загорится светодиод, подключенный к 13 пину Arduino.

### Пример программного кода:

```
#define Trig 9
#define Echo 8
#define ledPin 13

void setup()
{
  pinMode(Trig, OUTPUT); //иницируем как выход
  pinMode(Echo, INPUT); //иницируем как вход
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  /* задаем скорость общения. В нашем случае с компьютером */
}

unsigned int impulseTime=0;
unsigned int distance_sm=0;

void loop()
{
  digitalWrite(Trig, HIGH);
  /* Подаем импульс на вход trig датчика */
  delayMicroseconds(10); // равный 10 микросекундам
  digitalWrite(Trig, LOW); // Отключаем
  impulseTime=pulseIn(Echo, HIGH); // Замеряем длину импульса
  distance_sm=impulseTime/58; // Пересчитываем в сантиметры
  Serial.println(distance_sm); // Выводим на порт
  if (distance_sm<30) // Если расстояние менее 30 сантиметром
  {
    digitalWrite(ledPin, HIGH); // Светодиод горит
  }
  else
  {
    digitalWrite(ledPin, LOW); // иначе не горит
  }
  delay(100);
  /* ждем 0.1 секунды, Следующий импульс может быть излучён, только после исчезновения эха от
  предыдущего.
  Это время называется периодом цикла (cycle period).
  Рекомендованный период между импульсами должен быть не менее 50 мс. */
}
```

### Дополнительный пример работы:

Взаимодействие дальномера и сервопривода. Дистанция, измеряемая дальномером, преобразуется в угол поворота сервопривода.

```
#include <Servo.h>

#define coef 10 //(коэффициент соответствия 10 градусов на 1см)
#define dead_zone 4
#define max_value 22
#define servoPin 11

#define Trig 9
#define Echo 8
#define ledPin 13

Servo myservo;

void setup()
{
  pinMode(Trig, OUTPUT); //инициуруем как выход
  pinMode(Echo, INPUT); //инициуруем как вход
  pinMode(ledPin, OUTPUT);
  myservo.attach(servoPin);
  myservo.write(0);
}

unsigned int impulseTime=0;
unsigned int distance_sm=0;

void loop()
{
  digitalWrite(Trig, HIGH); /* Подаем импульс на вход
  trig дальномера */
  delayMicroseconds(10); // равный 10 микросекундам
  digitalWrite(Trig, LOW); // Отключаем
  impulseTime = pulseIn(Echo, HIGH); // Замеряем длину импульса
  distance_sm = impulseTime/58; // Пересчитываем в сантиметры

  if (distance_sm >= dead_zone && distance_sm <= max_value)
  { myservo.write(coef * (distance_sm - dead_zone)); }
  else if (distance_sm < dead_zone)// если дистанция менее 4 см, серва в положении ноль градусов
  { myservo.write(0); }
  else
  { myservo.write(180); }
  delay(100); /* ждем 0.1 секунды,
  Следующий импульс может быть излучён, только после исчезновения эха от предыдущего. Это время
  называется периодом цикла (cycle period). Рекомендованный период между импульсами должен быть не
  менее 50 мс.
  */
}
```