

## **Эксперимент № 6. «Дальномер». Ультразвуковой датчик HC-SR04**

Вам понадобится

1. <http://tinkercad.com>

Список деталей для эксперимента

- Ультразвуковой датчик HC-SR04
- 1 безопасная макетная плата
- 1 сервопривод
- провода «папа-мама»
- Arduino UNO R3

Пошаговое руководство для проведения эксперимента

Модуль оборудован четырех-пиновым разъемом стандарта 2.54мм

- VCC: "+" питания
- TRIG (T): вывод входного сигнала
- ECHO (R): вывод выходного сигнала (Длина сигнала зависит от расстояния объекта до датчика)
- GND: "-" питания

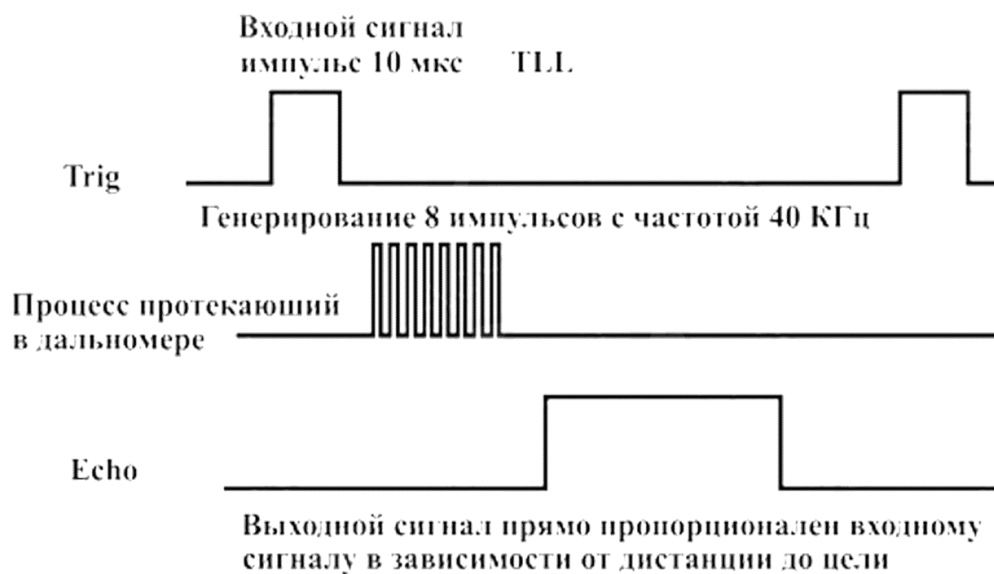
**Обратите внимание.** Работа ультразвукового дальномера основана на испускании ультразвука и его отражения от впереди находящихся предметов. Сенсор дальномера работает по принципу сонара, а точнее — посылает ультразвуковой пучок и по задержке отражённого от объекта сигнала определяет расстояние до цели. Поскольку в основе работы устройства используется ультразвук, сонар плохо подходит для определения расстояния до звукопоглощающих объектов. Поверхность объекта должна быть ровной и гладкой для идеального измерения расстояния.

### **Особенности ультразвукового дальномера HC-SR04**

- Точность измерения:  $\pm 1$  см (при максимальной дальности  $\pm 3$  см).
- Угол измерения:  $15^\circ$ .
- Ультразвуковой диапазон работы на частоте 40 кГц.
- Рабочее напряжение 4,8 В до 5,5 В ( $\pm 0.2$ В макс).
- Диапазон рабочих температур:  $0^\circ \text{C}$  до  $60^\circ \text{C}$  ( $\pm 10\%$ ).

**Обратите внимание.** Принцип работы датчика можно условно разделить на 4 этапа:

1. Подаем импульс продолжительностью 10 мкс, на вывод Trig.
2. Внутри дальномера входной импульс преобразуется в 8 импульсов частотой 40 КГц и посылается вперед через "Т глазик"
3. Дойдя до препятствия, посланные импульсы отражаются и принимаются "R глазиком". Получаем выходной сигнал на выводе Echo.
4. Исходя из времени возвращения звука, по простой формуле, можно рассчитать расстояние до объекта. Непосредственно на стороне контроллера переводим полученный сигнал в расстояние по формуле:



ширина импульса (мкс) / 58= дистанция (см)

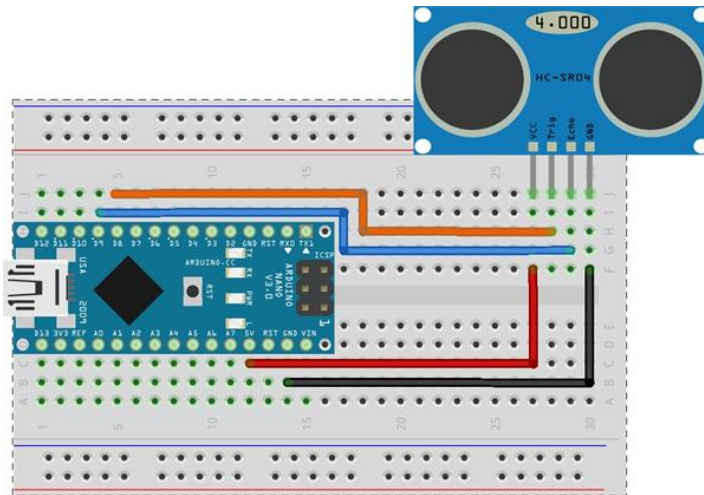
ширина импульса (мкс) / 148= дистанция (дюйм)

Подключение к Arduino

УЗ датчик	Arduino
VCC	+5V
TRIG	D8
ECHO	D9
GND	GND

Ну а теперь после удачного подключения ультразвуковой датчика HC-SR04 к Arduino разберемся с программной частью. Для начала мы используем простой скетч, который поможет нам определить расстояние до объекта в сантиметрах и воспроизвести звук.

**Пример 1. Подключение HC-SR04 к Arduino по 4 контактам (VCC, Trig, Echo, GND)**



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

```
void setup()
```

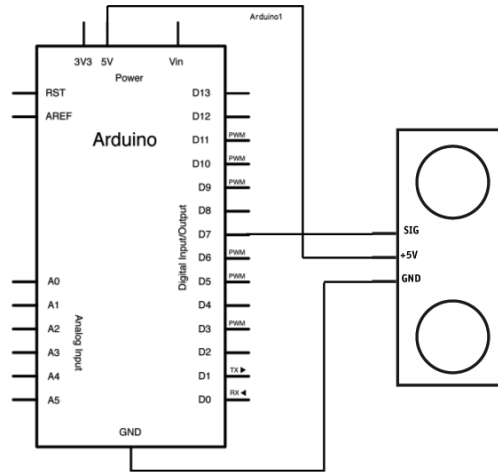
```
{
  pinMode(Trig, OUTPUT); //иницируем как выход
  pinMode(Echo, INPUT); //иницируем как вход
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  /* задаем скорость общения. В нашем случае с компьютером */
}
```

```
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. */
}
```

## Пример 2. Подключение HC-SR04 к Arduino по 3 контактам (VCC, Ping, GND)



```
/* Ping)) Sensor
```

В этом примере ультразвуковой дальномер считывает PING и возвращает расстояние до ближайшего объекта в диапазоне. Для этого он посылает импульс к датчику, чтобы инициировать чтение, а затем прослушивает пульс чтобы вернуться. Длина возвращающей импульса пропорциональна расстоянию объекта от датчика.

```
* VCC - +5V
```

```
* GND
```

```
* SIG - pin 7
```

```
*/
```

```
const int pingPin = 7;
```

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

```
void loop() {
```

```
  long duration, inches, cm;
```

```
  pinMode(pingPin, OUTPUT);  
  digitalWrite(pingPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(pingPin, HIGH);  
  delayMicroseconds(5);  
  digitalWrite(pingPin, LOW);
```

```
  pinMode(pingPin, INPUT);  
  duration = pulseIn(pingPin, HIGH);
```

```
  inches = microsecondsToInches(duration);  
  cm = microsecondsToCentimeters(duration);
```

```
  Serial.print(inches);  
  Serial.print("in, ");  
  Serial.print(cm);  
  Serial.print("cm");
```

```

Serial.println();

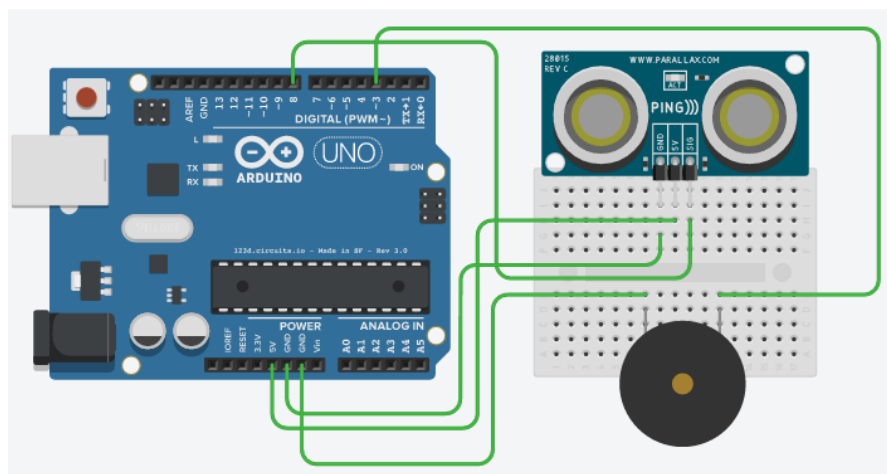
delay(100);
}

long microsecondsToInches(long microseconds) {
return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
return microseconds / 29 / 2;
}

```

**Пример 3. Обнаружение предмета с использованием HC-SR04 и пьезодинамика**



```

#define BUZZER_PIN 3
int echoPin = 9;
int trigPin = 8;

void setup() {
Serial.begin(9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
int duration, cm, frequency;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
cm = duration / 58;

frequency = map(cm, 0, 1023, 1500, 2500);
// заставляем пин с пищалкой «вибрировать», т.е. звучать
// (англ. tone) на заданной частоте 20 миллисекунд. При
// следующих проходах loop, tone будет вызван снова и снова,

```

```

// и на деле мы услышим непрерывный звук тональностью, которая
// зависит от количества света, попадающего на фоторезистор
if (cm>5){
if (cm<40) {
    tone(BUZZER_PIN, frequency, 50); }
    Serial.print(cm);
    Serial.println(" cm");
    delay(100);
}
}

```

**Задание.** Взаимодействие дальномера и сервопривода. Дистанция, измеряемая дальномером преобразуется в угол поворота сервопривода. Подключите устройства в соответствии с распиновкой на основе приведенного ниже скетча.

```

#include <Servo.h>
#define coef 10 //(коэффициент соответствия 10 градусов на 1см)
#define dead_zone 4
#define max_value 22
#define Trig 9
#define Echo 8
#define ledPin 13
#define servoPin 11

Servo myservo;

void setup()
{
    pinMode(Trig, OUTPUT); //иницируем как выход
    pinMode(Echo, INPUT); //иницируем 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.

```
*/  
}
```

