



Proyecto

Sensor Ultrasónico HC-SR04

DIEGO MURACCIOLE

NOVIEMBRE 2014

Fundamento teórico

Fundamentos teóricos

- ▶ Ultrasonido: señal acústica cuya frecuencia están por encima del rango sensibles al oído humano
- ▶ Los sensores ultrasónicos miden la distancia de la que están respecto a un objeto.
- ▶ Están formados por un transductor que emite un pulso de energía ultrasónica y un receptor que captura el eco de la señal tras haber sido reflejado por el objeto.
- ▶ La distancia por lo tanto se calcula por la ecuación $d = \frac{1}{2} v \cdot t$, donde v es la velocidad del sonido en el aire y t el tiempo transcurrido entre emisión y recepción.

Aplicaciones

- ▶ Medición de nivel (ej. tanques)
- ▶ Control de colisiones
- ▶ Control de posición en campos como la robótica

Ventajas

- ▶ No necesitan contacto físico para detección de objetos
- ▶ Accesibilidad económica
- ▶ Compactos
- ▶ Livianos



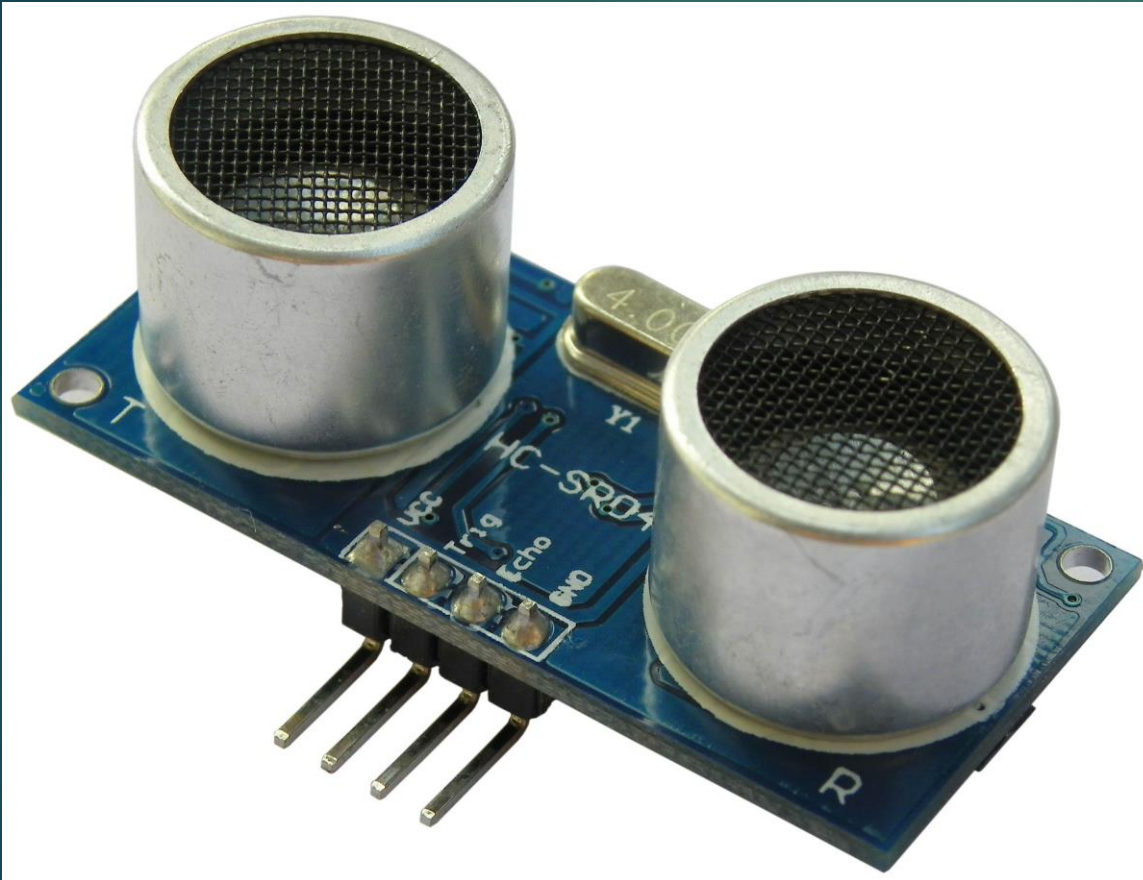
HC-SR04

HC-SR04



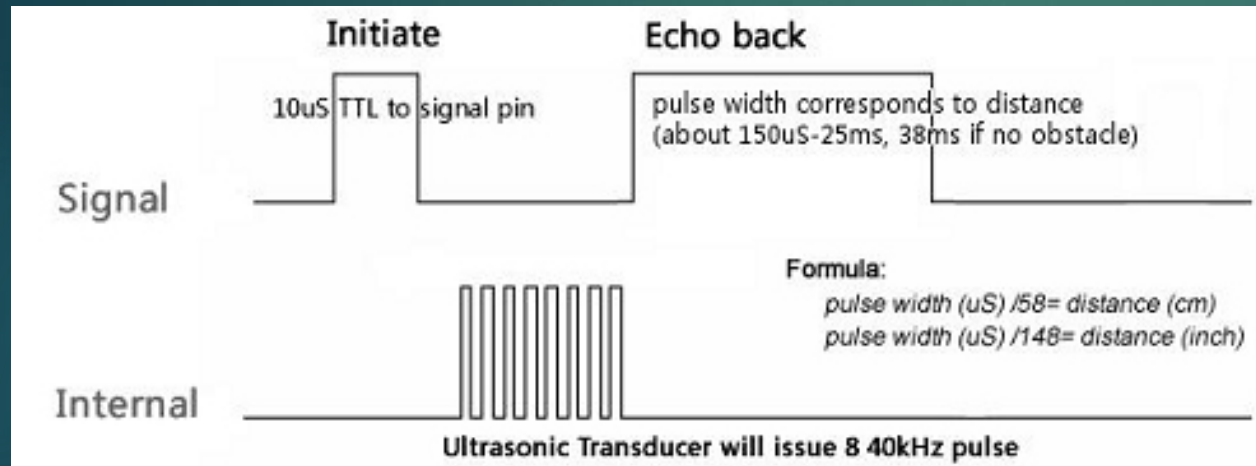
- Características
 - Working Voltage DC 5 V
 - Working Current 15mA
 - Working Frequency 40Hz
 - Max Range 4m
 - Min Range 2cm
 - Measuring Angle 15 degree
 - Trigger Input Signal 10uS TTL pulse
 - Echo Output Signal Input TTL level signal and the range in proportion
 - Dimension 45*20*15mm

HC-SR04



- Pines de conexión
 - VCC
 - Trig (Disparo del ultrasonido)
 - Echo (Recepción del ultrasonido)
 - GND
- Precio
 - UY: U\$S 9 [\(link\)](#)
 - EEUU: U\$S 2,89 [\(link\)](#)

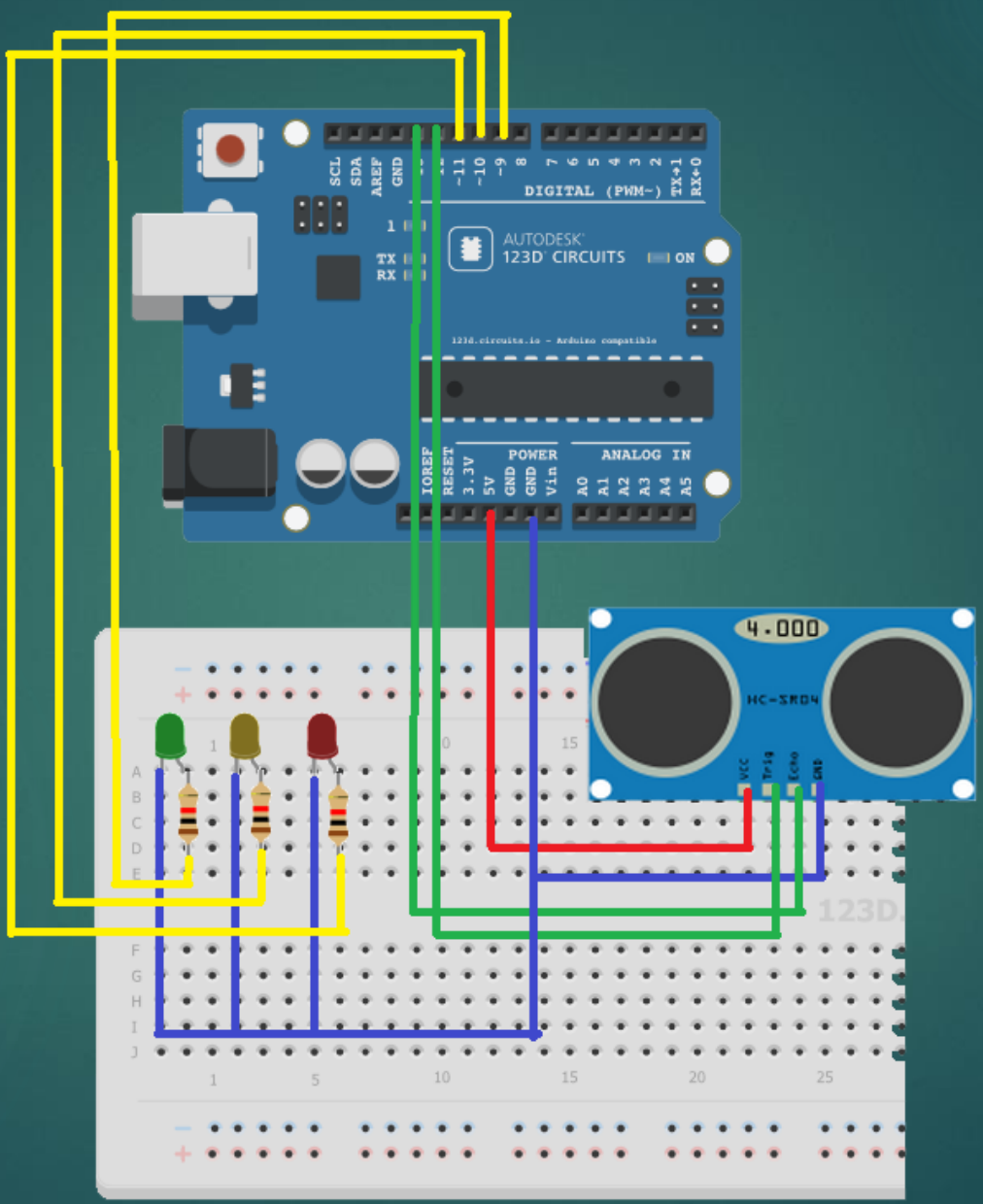
HC-SR04



- Funcionamiento
 - Para obtener una medida es necesario generar un pulso al pin Trig de un tiempo mínimo de 10µs
 - Esto genera que el módulo mande una señal de 8 ciclos de 40kHz
 - Monitorizar la señal que llega al pin Echo y calcular la distancia del objeto como $\text{distancia} = (\text{ancho del pulso} * \text{velocidad sonido}) / 2$



Circuito





Código

```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
int Pin_green = 9;

void setup() {
  pinMode(Pin_trig, OUTPUT);
  pinMode(Pin_echo, INPUT);
  pinMode(Pin_red, OUTPUT);
  pinMode(Pin_yellow, OUTPUT);
  pinMode(Pin_green, OUTPUT);
}

void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
digitalWrite(Pin_green, LOW);
digitalWrite(Pin_red, LOW);
digitalWrite(Pin_yellow, LOW);

if(cm>5){
  digitalWrite(Pin_green, HIGH);
}
if(cm>15){
  digitalWrite(Pin_yellow, HIGH);
}
if(cm>30){
  digitalWrite(Pin_red, HIGH);
}

delay(200);
}
```

Definición de pines

```
int Pin_echo = 13;  
int Pin_trig = 12;  
int Pin_red = 11;  
int Pin_yellow = 10;  
int Pin_green = 9;
```

```
void setup() {  
  pinMode(Pin_trig, OUTPUT);  
  pinMode(Pin_echo, INPUT);  
  pinMode(Pin_red, OUTPUT);  
  pinMode(Pin_yellow, OUTPUT);  
  pinMode(Pin_green, OUTPUT);  
}
```

```
void loop() {  
  int pulso, cm;  
  digitalWrite(Pin_trig, LOW);  
  delayMicroseconds(2);  
  digitalWrite(Pin_trig, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(Pin_trig, LOW);  
  pulso = pulseIn(Pin_echo, HIGH);  
  cm = pulso / 29 / 2;
```

```
  digitalWrite(Pin_green, LOW);  
  digitalWrite(Pin_red, LOW);  
  digitalWrite(Pin_yellow, LOW);  
  
  digitalWrite(Pin_green, HIGH);  
}  
if(cm>15){  
  digitalWrite(Pin_yellow, HIGH);  
}  
if(cm>30){  
  digitalWrite(Pin_red, HIGH);  
}  
  
  delay(200);  
}
```

```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
int Pin_green = 9;

void setup() {
  pinMode(Pin_trig, OUTPUT);
  pinMode(Pin_echo, INPUT);
  pinMode(Pin_red, OUTPUT);
  pinMode(Pin_yellow, OUTPUT);
  pinMode(Pin_green, OUTPUT);
}

void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
digitalWrite(Pin_green, LOW);
digitalWrite(Pin_red, LOW);
digitalWrite(Pin_yellow, LOW);

if(cm>5){
  digitalWrite(Pin_green, HIGH);
}

digitalWrite(Pin_red, HIGH);
digitalWrite(Pin_yellow, HIGH);
delay(200);
}
```

Configuración in/out de pines en método setup


```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
int Pin_green = 9;
```

```
void setup() {
  pinMode(Pin_trig, OUTPUT);
  pinMode(Pin_echo, INPUT);
  pinMode(Pin_red, OUTPUT);
  pinMode(Pin_yellow, OUTPUT);
  pinMode(Pin_green, OUTPUT);
}
```

```
void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
  digitalWrite(Pin_green, LOW);
  digitalWrite(Pin_red, LOW);
  digitalWrite(Pin_yellow, LOW);
```

```
  if(cm>5){
    digitalWrite(Pin_green, HIGH);
  }
  if(cm>15){
    digitalWrite(Pin_yellow, HIGH);
  }
  if(cm>30){
    digitalWrite(Pin_red, HIGH);
  }
}
```



Estabiliza el sensor

```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
int Pin_green = 9;

void setup() {
  pinMode(Pin_trig, OUTPUT);
  pinMode(Pin_echo, INPUT);
  pinMode(Pin_red, OUTPUT);
  pinMode(Pin_yellow, OUTPUT);
  pinMode(Pin_green, OUTPUT);
}

void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
digitalWrite(Pin_green, LOW);
digitalWrite(Pin_red, LOW);
digitalWrite(Pin_yellow, LOW);

if(cm>5){
  digitalWrite(Pin_green, HIGH);
}
if(cm>15){
  digitalWrite(Pin_yellow, HIGH);
}
if(cm>30){
  digitalWrite(Pin_red, HIGH);
}

delay(200);
}
```

Emite pulso de 10us por el
pin Trig

```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
int Pin_green = 9;

void setup() {
  pinMode(Pin_trig, OUTPUT);
  pinMode(Pin_echo, INPUT);
  pinMode(Pin_red, OUTPUT);
  pinMode(Pin_yellow, OUTPUT);
  pinMode(Pin_green, OUTPUT);
}

void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
digitalWrite(Pin_green, LOW);
digitalWrite(Pin_red, LOW);
digitalWrite(Pin_yellow, LOW);

if(cm>5){
  digitalWrite(Pin_green, HIGH);
}
if(cm>15){
  digitalWrite(Pin_yellow, HIGH);
}
if(cm>30){
  digitalWrite(Pin_red, HIGH);
}

delay(200);
}
```

Mide ancho del pulso por el
pin de Echo

```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
int Pin_green = 9;

void setup() {
  pinMode(Pin_trig, OUTPUT);
  pinMode(Pin_echo, INPUT);
  pinMode(Pin_red, OUTPUT);
  pinMode(Pin_yellow, OUTPUT);
  pinMode(Pin_green, OUTPUT);
}

void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
digitalWrite(Pin_green, LOW);
digitalWrite(Pin_red, LOW);
digitalWrite(Pin_yellow, LOW);

if(cm>5){
  digitalWrite(Pin_green, HIGH);
}
if(cm>15){
  digitalWrite(Pin_yellow, HIGH);
}
if(cm>30){
  digitalWrite(Pin_red, HIGH);
}

delay(200);
}
```

Calcula distancia en cm

```
int Pin_echo = 13;
int Pin_trig = 12;
int Pin_red = 11;
int Pin_yellow = 10;
```

Actua según la medición
obtenida

```
pinMode(Pin_yellow, OUTPUT);
pinMode(Pin_green, OUTPUT);
}
```

```
void loop() {
  int pulso, cm;
  digitalWrite(Pin_trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Pin_trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Pin_trig, LOW);
  pulso = pulseIn(Pin_echo, HIGH);
  cm = pulso / 29 / 2;
```

```
  digitalWrite(Pin_green, LOW);
  digitalWrite(Pin_red, LOW);
  digitalWrite(Pin_yellow, LOW);
```

```
  if(cm>5){
    digitalWrite(Pin_green, HIGH);
  }
  if(cm>15){
    digitalWrite(Pin_yellow, HIGH);
  }
  if(cm>30){
    digitalWrite(Pin_red, HIGH);
  }
```

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