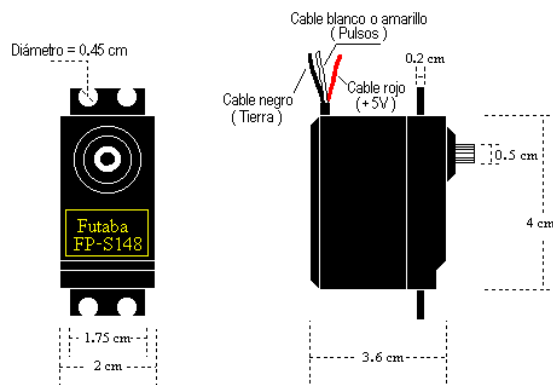


## SERVOMOTOR

Un **servomotor** (también llamado **servo**) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

En otras palabras, un servomotor es un motor especial al que se ha añadido un sistema de control (tarjeta electrónica), un potenciómetro y un conjunto de engranajes. Con anterioridad los servomotores no permitían que el motor girara 360 grados, solo aproximadamente 180; sin embargo, hoy en día existen servomotores en los que puede ser controlada su posición y velocidad en los 360 grados. Los servomotores son comúnmente usados en modelismo como aviones, barcos, helicópteros y trenes para controlar de manera eficaz los sistemas motores y los de dirección.



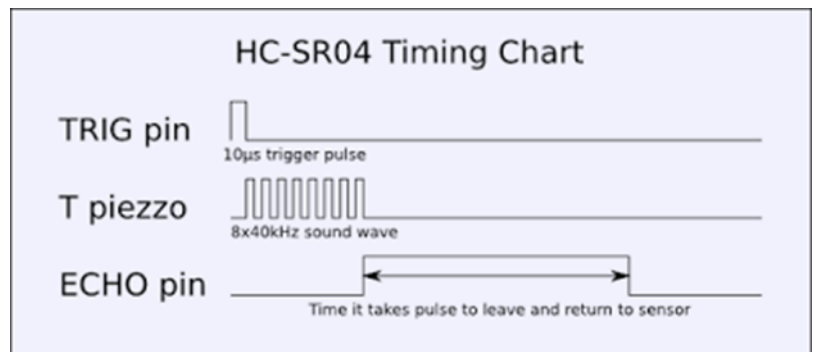
## PROGRAMACIÓN

```
#include <Servo.h> // cargar la biblioteca servo
Servo myServo; // crear el objeto servo (uno por cada servo que queramos
controlar. myServo es el nombre que le he asignado al servo sin hubiese más
de uno tendría que crear otro objeto servo con un nombre diferente por
ejemplo myServo2
void setup()
{
  myServo.attach(nº pin); // defino en que pin está el control del servo
  (generalmente cable de color blanco o amarillo) tiene que ser un pin ~

}

myServo.write(ángulo de giro); // ira dentro del void loop
```

## SENSOR DE ULTRASONIDOS



### Funcionamiento:

1. Enviar un Pulso "1" de al menos de 10µs por el Pin Trigger (Disparador).
2. El sensor enviará 8 Pulsos de 40KHz (Ultrasonido) y coloca su salida Echo a nivel alto (1), se debe detectar este evento e iniciar un conteo de tiempo.
3. La salida Echo se mantendrá en alto hasta recibir el eco reflejado por el obstáculo a lo cual el sensor pondrá su pin Echo a bajo, es decir, terminar de contar el tiempo.
4. Se recomienda dar un tiempo de aproximadamente 50ms de espera después de terminar la cuenta.

La distancia es proporcional a la duración del pulso y puedes calcularla con la siguiente formula (Utilizando la velocidad del sonido = 340m/s):

$$\text{Distancia en cm (centímetros)} = \text{Tiempo medido en } \mu\text{s} \times 0.017$$

### PROGRAMACION

Deberemos utilizar una nueva definición de variables donde guardar la duración del pulso echo

`long duracion;` sirve para guardar números desde -2,147,483,648 a 2,147,483,647.

otras ordenes

`delayMicroseconds(10);` // funciona igual que delay, pero en vez de ms en µs

`pulseIn(Pin, HIGH);` // Pin es donde está conectada la patilla echo, HIGH mira cuando pin se ha puesto a 1, si ponemos LOW mira cuando se pone a 0. Esta orden nos da el tiempo en µs que la patilla echo ha estado en 1 o 0 dependiendo si puse HIGH o LOW. En nuestro caso deberemos mirar el HIGH