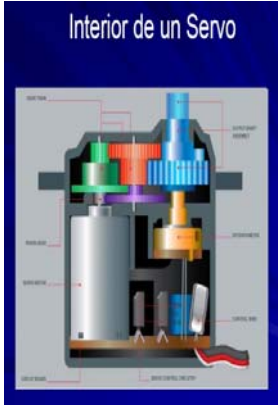


# Arduino Servomotores.

## Arduino Servomotores.

Los servos estándar sólo giran 180°. Estos servomotores son motores de corriente continua que incorporan un circuito electrónico que permite controlar de forma sencilla la dirección y la velocidad de giro de sus ejes mediante impulsos eléctricos (PWM).



Hay servomotores de rotación continua que puede girar 360°.

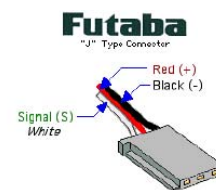
Se puede convertir un servo en uno de rotación continua (<http://www.youtube.com/watch?v=q6UGT1eWCRC&feature=fvw> o <http://www.youtube.com/watch?v=CtLN8BoMIQ> )



El servomotor tiene 3 cables. Dos de alimentación (rojo (+) y negro (-)), entre 5V y 7.5V, y uno de señal de control (amarillo o blanco), que se conecta directamente (los de pequeño consumo) a un pin PWM de la placa Arduino.

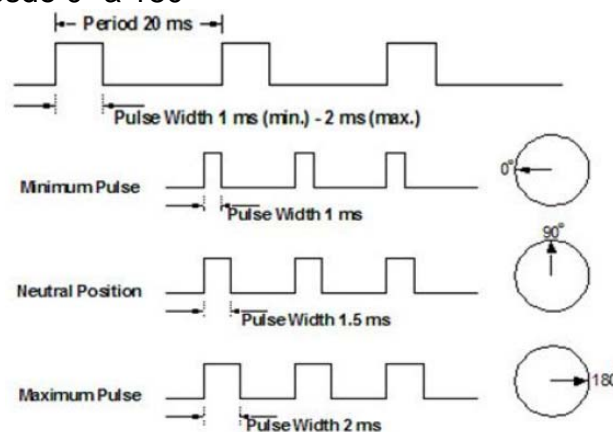
### Ejemplo de conexiones:

Fabricante	Voltaje positivo	Tierra	Señal de control
Futaba	Rojo	Negro	Blanco
Dong Yang	Rojo	Marrón	Naranja
Hitec	Rojo	Negro	Amarillo
JR	Rojo	Marrón	Naranja
Airtronics	Rojo	Negro	Naranja
Fleet	Rojo	Negro	Blanco
Krafr	Rojo	Negro	Naranja
E-Sky	Rojo	Negro	Blanco



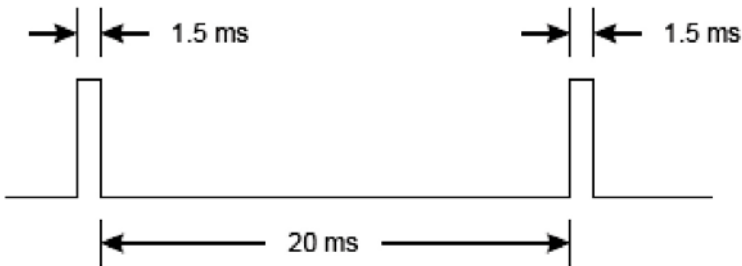
### Funcionamiento servos:

La señal de control de los servomotores es una señal de pulsos modulada en anchura PWM (Pulse Width Modulation). Este tipo de señal de control se utiliza en los servos estándar para realizar los giros desde 0° a 180°



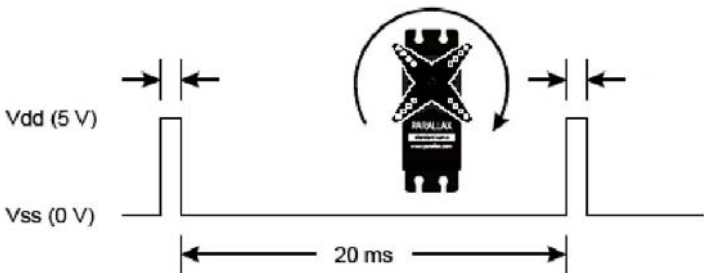
La señal de control de los servomotores de rotación continua es una señal de pulsos modulada en anchura PWM (Pulse Width Modulation). Permitir controlar el giro y la velocidad del eje del servomotor.

En teoría, porque en la fabricación de los motores eléctricos tiene una tolerancia, con el siguiente pulso de señal el servomotor de rotación continua tendría que estar parado.



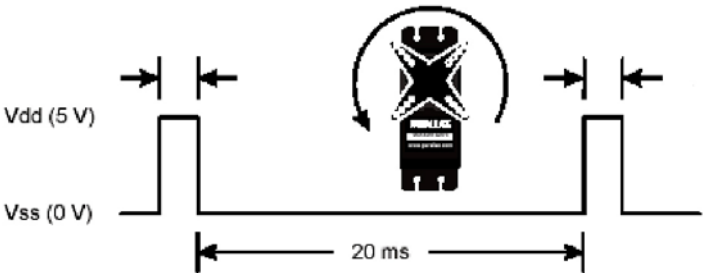
Señal con periodo de 20 ms.  
Anchura del pulso 1.5 ms.

Para el giro a máxima velocidad en el sentido de las agujas del reloj el pulso de la señal sería:



Señal con periodo de 20 ms.  
Anchura del pulso entre 0.8 a 1.2 ms.

Para el giro a máxima velocidad en el sentido contrario a las agujas del reloj el pulso de la señal sería:



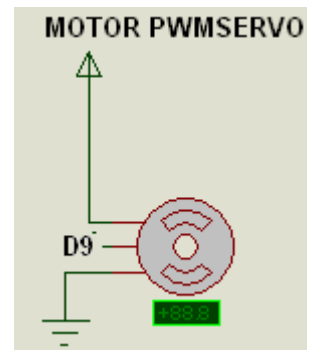
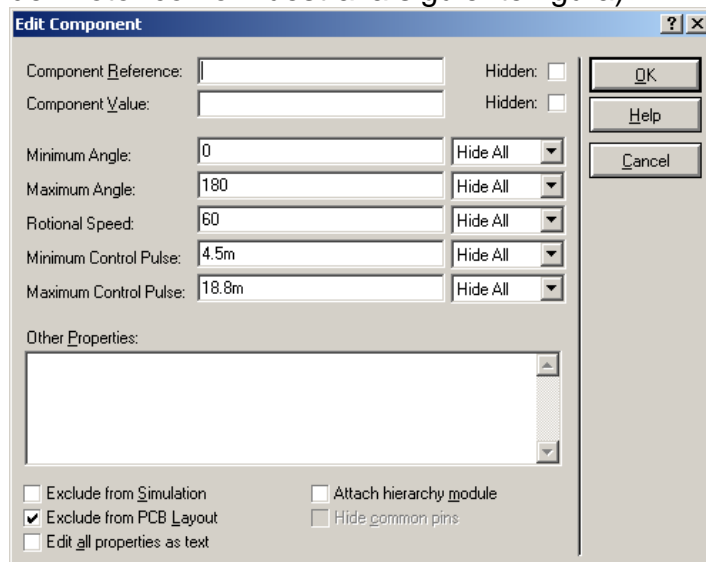
Señal con período de 20 ms.  
Anchura del pulso entre 1.9 a 2.2 ms

En la siguiente tabla se reflejan los tiempos de anchura de pulsos de diferentes fabricantes:

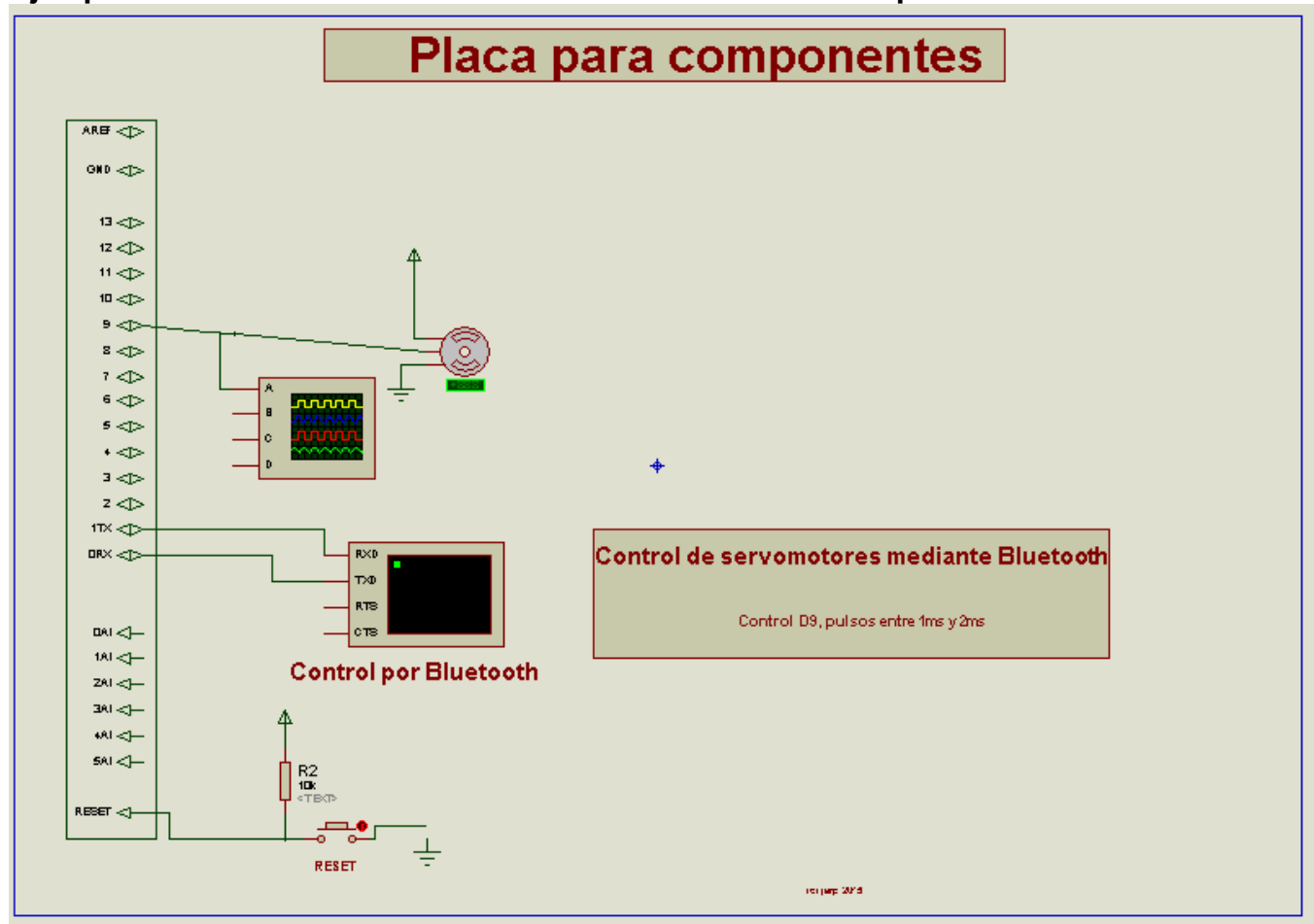
Fabricante	Duración pulso (ms)				disposición de cables		
	min.	neutral.	máx..	Hz	+ batt	-batt	pwm.
Futaba	0.9	1.5	2.1	50	rojo	negro	blanco
Hitech	0.9	1.5	2.1	50	rojo	negro	amarillo
Graupner/Jr	0.8	1.5	2.2	50	rojo	marrón	naranja
Multiplex	1.05	1.6	2.15	40	rojo	negro	amarillo
Robbe	0.65	1.3	1.95	50	rojo	negro	blanco
Simprop	1.2	1.7	2.2	50	rojo	azul	negro

Para el control con arduino utilizamos la librería `<servo.h>`.

Al simular con **proteus** utilizamos el **MOTOR-PWMSERVO** (configurar propiedades del motor como muestra la siguiente figura).



**Ejemplo de simulación con Proteus de un servo controlado por Bluetooth:**



Con el módulo bluetooth podemos controlarlo. Para simular **VIRTUAL TERMINAL** (configurar velocidad del simulador a 1200). Simulamos la información recibida por bluetooth con ayuda del teclado.

## Programas arduino:

```
// programa para girar desde 0 a 180º
#include <Servo.h>

Servo myservo;           // crea el objeto servo para controlar el servo
int pos = 0;             // posición servo

void setup()
{
  myservo.attach(9);      // asigna pin 9 para el control de servo (solo pines PWM)
}

void loop()
{
  for(pos = 0; pos < 179; pos += 1) // envía posición desde 0 a 180 de grado en grado
  {
    myservo.write(pos);      // envía posición
    delay(15);               // espera 15ms para posicionar el servo
  }
}
```

```
//programa para posicionar el motor mediante bluetooth
#include <Servo.h>

Servo myservo;           // crea el objeto servo para controlar el servo

int val;                 // variable para mandar señal al servo
char valor;              //almacena valor leído puerto serie

void setup()
{
  myservo.attach(9);      // asigna pin 9 para el control de servo (solo pines PWM)
  Serial.begin(9600);
  Serial.println("teclas 4:0, 7:45, 8:90, 9:135, 6:180");
}

void loop()
{
  if(Serial.available()>0){
    valor=Serial.read();
    Serial.println(valor);
    switch (valor) {
    case '8':
      Serial.println("90");
      val= 90;
      break;
    case '7':
      Serial.println("45");
      val= 45;
    }
  }
}
```

```
break;
case '4':
  Serial.println("0");
  val= 1;
  break;
case '6':
  Serial.println("180");
  val= 179;
  break;
case '9':
  Serial.println("135");
  val= 135;
  break;
} // switch
Serial.println(val);
myservo.write(val);           // envía posición
} //serial
delay(100);                  //retardo para leer tecla
} //loop
```

**Ver ejemplo de arduino Servo:**

- Knob.
- Sweep.