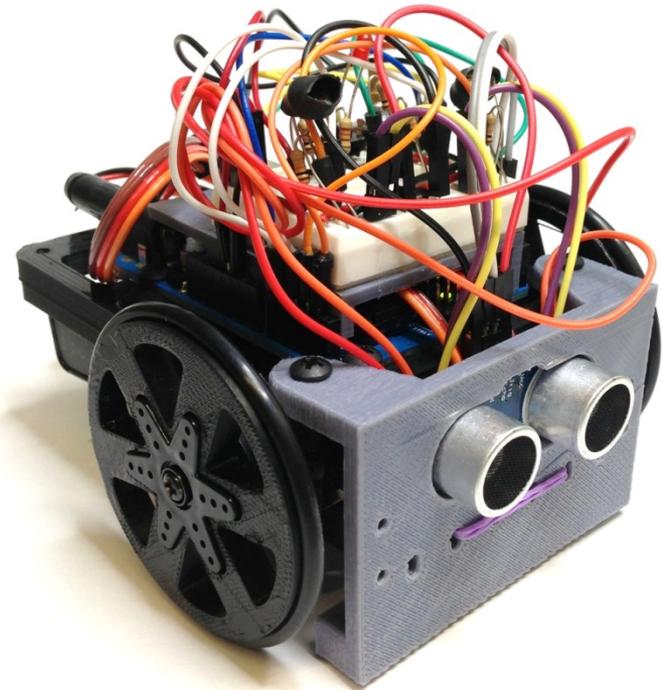
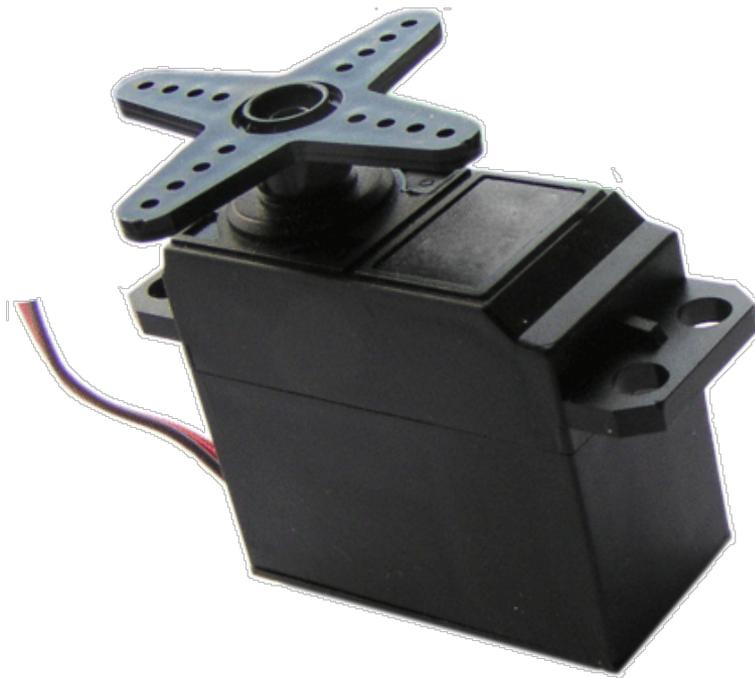


GUIA DE MONTAJE Y PROGRAMACIÓN

MOTORES

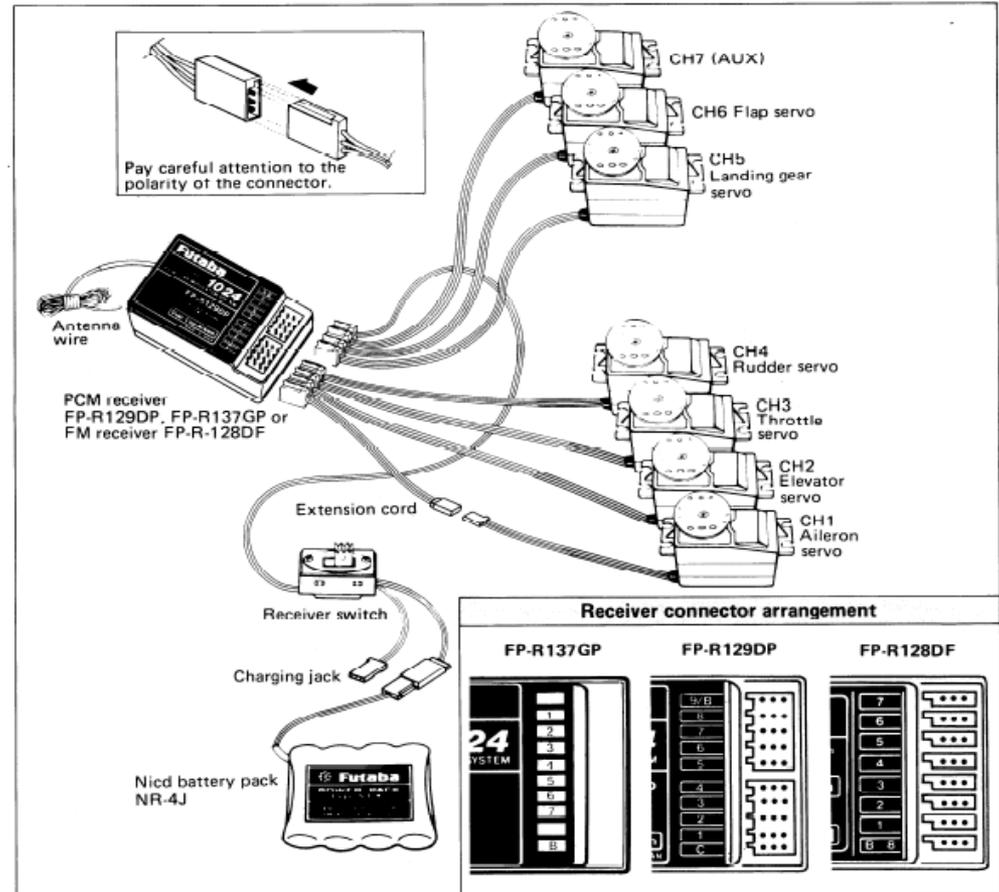


Conectamos los motores



Servomotores de radiocontrol (RC-Servos)

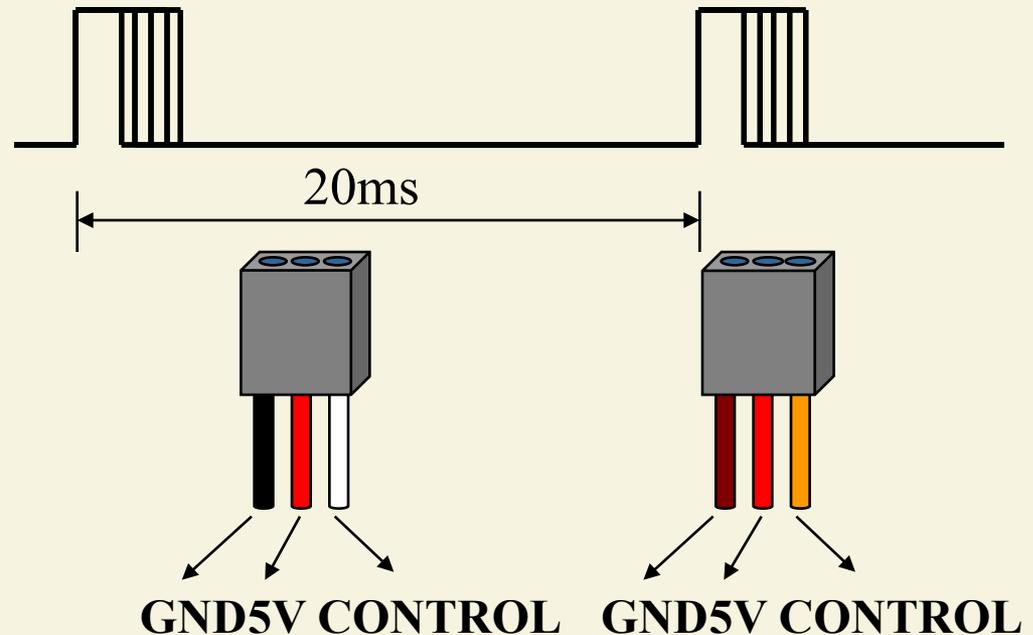
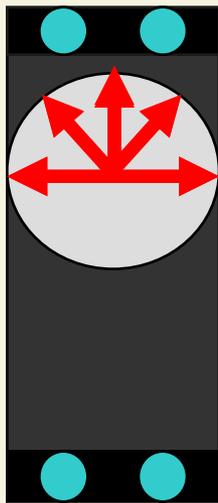
- El robot TuBot utiliza como motores dos servomotores de radiocontrol modificados.
- Son utilizados normalmente en coches y aviones de radiocontrol



Servomotores de radiocontrol (RC-Servos)

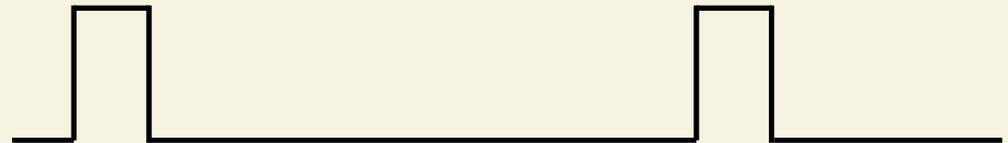
- Funcionamiento de un servomotor de radiocontrol
 - El servomotor está diseñado para mantener una posición. No para girar continuamente.
 - Utiliza 3 cables: 5V, 0V y señal de control.
 - Se controla con una señal periódica como la de la figura
 - La posición del eje del servomotor depende del ancho del pulso.

Tiempo a nivel alto: de 1 a 2ms (depende de las marcas)



Servomotores de radiocontrol (RC-Servos)

- Librería Servo de Arduino para controlar servomotores
 - Enlace a la documentación original de la librería servo
 - <https://www.arduino.cc/en/Reference/Servo>
 - Para utilizar la librería hay que crear un objeto Servo
Servo miServo;
 - Hay tres métodos que destacar:
 - **miServo.attach(pin);**
 - Sirve para relacionar el objeto Servo con un pin concreto por que genera la señal de control
 - **miServo.write(grados)**
 - Mueve el servo a una posición determinada en grados de 0° a 180°
 - Genera una señal periódica como la de la figura variando el ancho del pulso dependiendo del ángulo deseado.
 - **miServo.writeMicroseconds(tiempo)**
 - Genera una señal periódica de 20ms de periodo con un ancho de pulso de "tiempo" microsegundos.



- Con estas funciones se puede controlar la posición de un servomotor de radiocontrol.



□ Elementos internos de un servomotor de radiocontrol

■ El servomotor internamente tiene cuatro partes:

□ Motor de corriente continua

- Es el causante del movimiento

□ Reductora

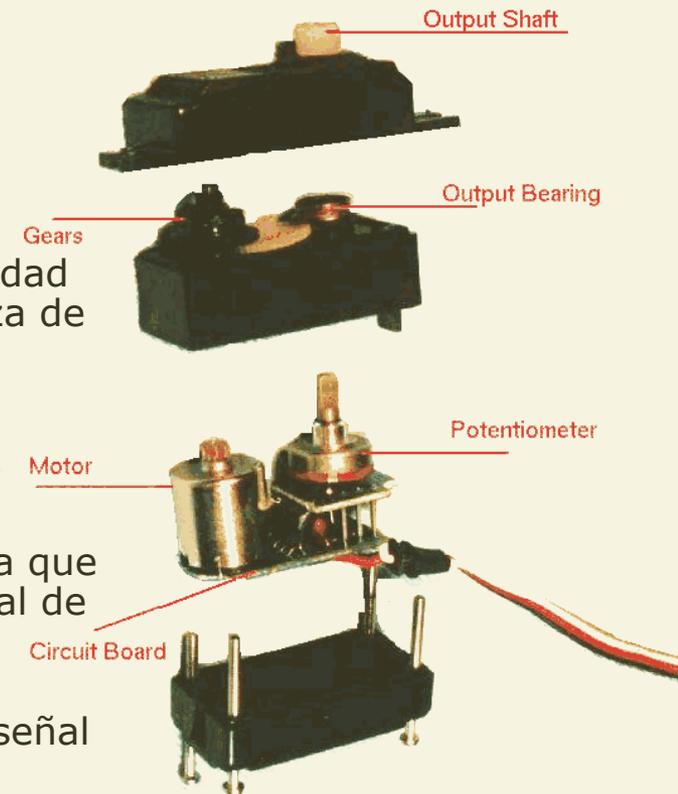
- Conjunto de engranajes que reduce la velocidad de giro del eje a la vez que aumenta la fuerza de giro (par)

□ Potenciómetro

- Mide la colocación del eje en cada momento.

□ Electrónica de control y potencia

- Da las órdenes a los motores necesarias para que el eje alcance la posición indicada por la señal de control de entrada.
- Da orden de girar a derecha o izquierda dependiendo de dónde se le dice que vaya (señal de control) y dónde está (potenciómetro)
- Para evitar pasarse al alcanzar la posición, cuanto más cerca está de su objetivo más despacio va.



Servomotores de radiocontrol (RC-Servos)

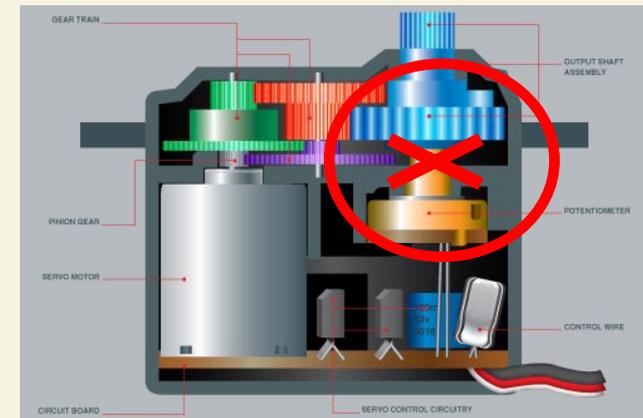
- Forma de convertir un servomotor de radiocontrol en un motor de corriente continua de rotación libre

- Pasos

- Se elimina un tetón que limita el movimiento sobre de los engranajes
 - Con esto permite girar al servo 360°
- Se desconecta el potenciómetro del eje de salida
 - Para que no gire con el movimiento del eje
- Se coloca el potenciómetro en el centro
 - Para que el sistema de control interno piense que siempre está en 90°

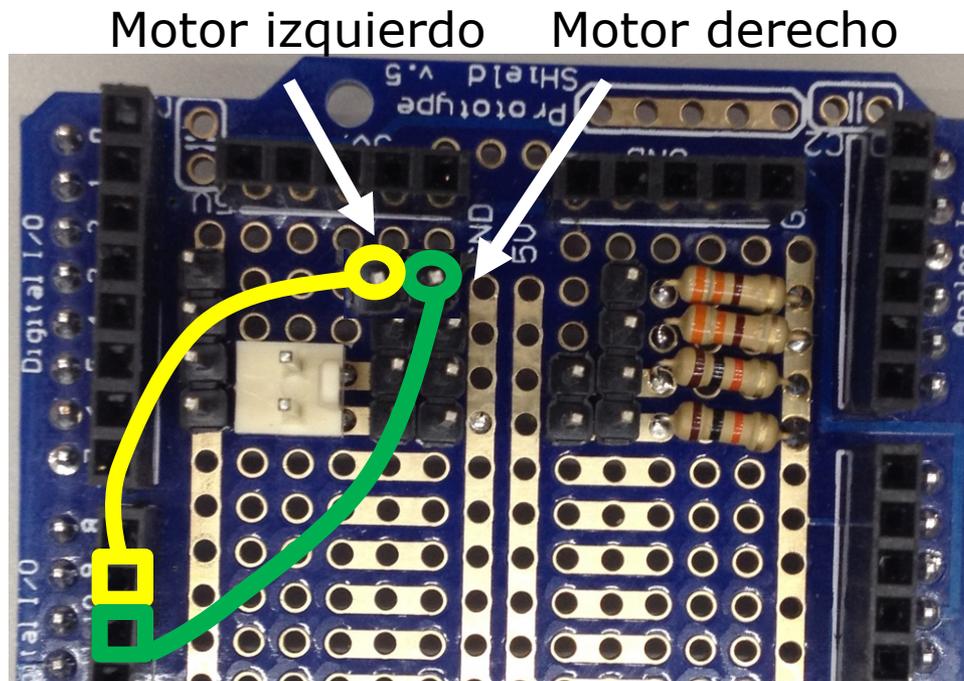
- ¿Qué conseguimos con esto?

- Que el motor se pare cuando le decimos que vaya a 90°
 - Porque el cree que ya está ahí
- Que gire en un sentido cuando se le dice que vaya a un ángulo mayor que 90° y en el otro cuando le decimos que vaya a una posición menor de 90°
 - El control cree que siempre está en 90° y da orden girar a un lado o a otro.
- Cuanto más lejos estamos de 90°, más rápido gira



Conectamos los motores

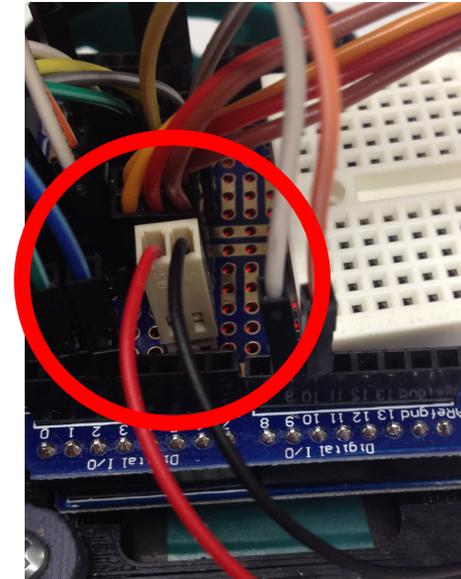
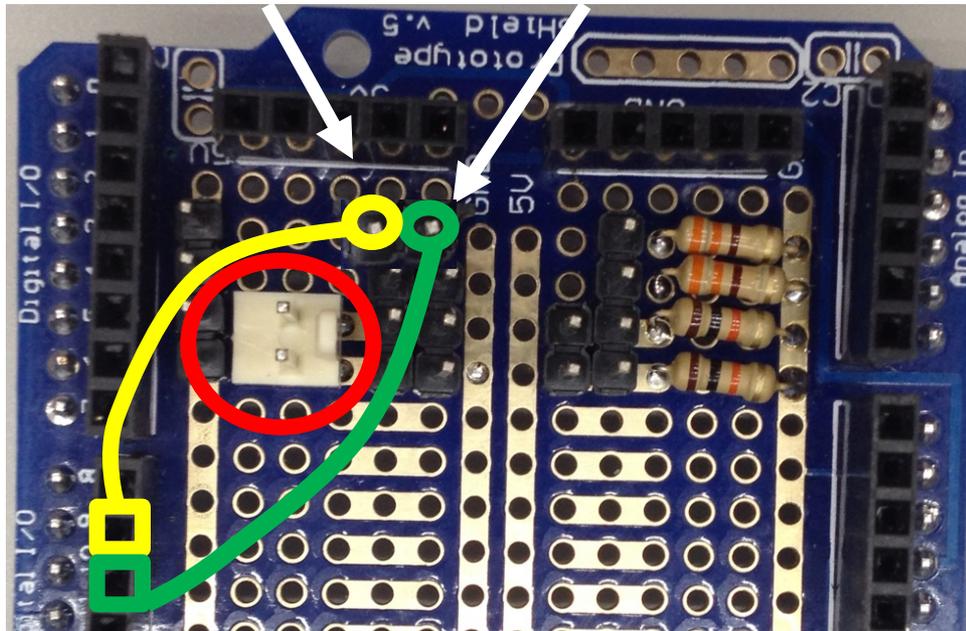
- ❑ Conecta las señales de control de los motores a la tarjeta de control.
- Conecta motor derecho al pin 10 de la tarjeta de control
 - ❑ Conecta el cable **verde** macho-hembra desde los conectores macho marcados en la imagen con círculos el pin 10 marcado con cuadrado
- Conecta el motor izquierdo al pin 9 de la tarjeta de control
 - ❑ Conecta el cable **amarillo** macho - hembra



Conectamos los motores

- ❑ Conecta las baterías de los motores (pilas AAA) a la tarjeta de expansión.
- ❑ **¡CUIDADO CON LA COLOCACIÓN! Solo debe encajar en una posición.**
- ❑ No pongas las pilas hasta que no programes los motores. Pueden descargarse.

Motor izquierdo Motor derecho

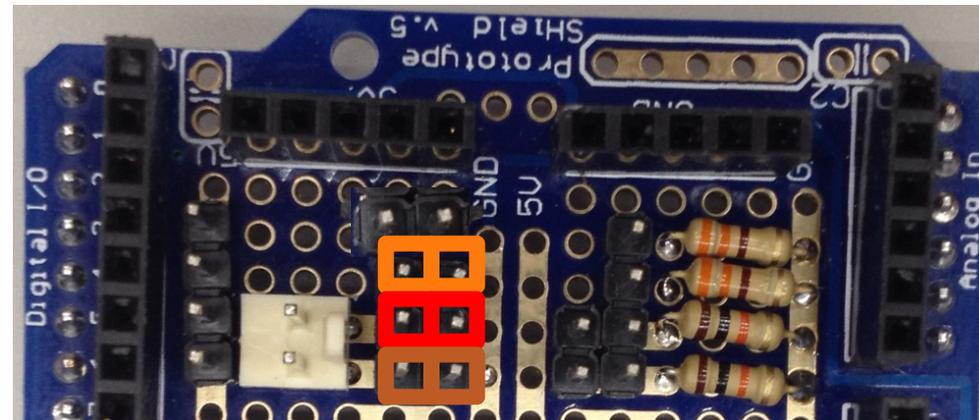
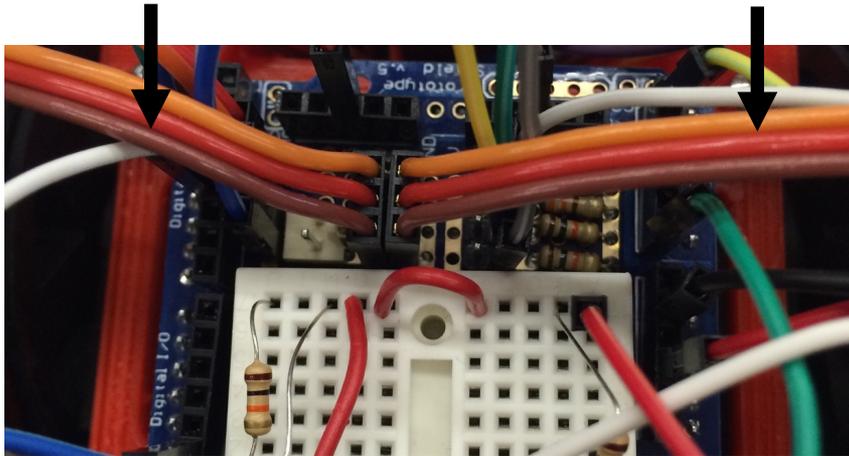


Conectamos los motores

- Conecta los cables que vienen de los motores a la tarjeta de conexiones.
- **¡CUIDADO CON LA COLOCACIÓN! Podemos romper los motores.**
- Los cables **marrón**, **rojo** y **negro** de los motores deben colocarse como se indica en las figuras.

Motor izquierdo

Motor derecho



Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)
 - Abre el ejemplo del botón que se llama PulsadorLED.
 - Archivo->Ejemplos->TuBot2016Lib->7_Motor->MotorPotenciometroGrados
 - Verifica y carga el programa en la tarjeta de control
 - Abre el Monitor Serie
 - Mueve el potenciómetro y comprobarás cómo cambia la velocidad de uno de los motores.

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacenará el valor de entrada analógica

void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo
}

void loop()
{
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que de valores de 0 a 180
  myservo.write(val); // Configura la posición del servo
  Serial.println(val); // Envá el valor enviado al motor
  delay(15); // Espera un poco a que el servomotor alcance la posición
}
```

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>
```



Es necesario indicar que se va a usar la librería Servo.h

```
Servo myservo; // Crea un objeto de tipo Servo
```

```
int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro  
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor  
int val; // Variable donde se almacenará el valor de entrada analógica
```

```
void setup()
```

```
{  
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado  
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo  
}
```

```
void loop()
```

```
{  
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023  
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que de valores de 0 a 180  
  myservo.write(val); // Configura la posición del servo  
  Serial.println(val); // Envá el valor enviado al motor  
  delay(15); // Espera un poco a que el servomotor alcance la posición  
}
```

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico
int pinServo = 10; // Indica el pin digital
int val; // Variable donde se almacenará el valor de entrada analógica
```

Configuración de los pines donde están conectados el potenciómetro y el motor

```
void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo
}
```

```
void loop()
{
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que de valores de 0 a 180
  myservo.write(val); // Configura la posición del servo
  Serial.println(val); // Envá el valor enviado al motor
  delay(15); // Espera un poco a que el servomotor alcance la posición
}
```

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacenará el valor de entrada analógica

void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin correspondiente
  Serial.begin(19200); // Se configura la comunicación serial
}

void loop()
{
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que de valores de 0 a 180
  myservo.write(val); // Configura la posición del servo
  Serial.println(val); // Envía el valor enviado al motor
  delay(15); // Espera un poco a que el servomotor alcance la posición
}
```

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacenará el valor de entrada analógica

void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo
}

void loop()
{
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que de valores de 0 a 180
  myservo.write(val); // Configura la posición del servo
  Serial.println(val); // Envá el valor enviado al motor
  delay(15); // Espera un poco a que el servomotor alcance la posición
}
```

Se lee el valor del potenciómetro

Puedes encontrar más información sobre la sentencia la función map en <https://www.arduino.cc/en/Reference/Map>

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacenará el valor de entrada analógica

void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo
}

void loop()
{
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que los valores de 0 a 1023
  myservo.write(val); // C
  Serial.println(val); // E
  delay(15); // E
}
```

La función **"map"** realiza una especie de regla de tres. Calcula la proporción de "val" en el rango de 0 a 1023 y devuelve el valor proporcional en el rango de 0 a 180

Puedes encontrar más información sobre la sentencia la función map en <https://www.arduino.cc/en/Reference/Map>

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacenará el valor de entrada analógica

void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo
}

void loop()
{
  val = analogRead(pinPot); // Lee el valor del potenciómetro
  val = map(val, 0, 1023, 0, 180); // Mapea el valor del potenciómetro a un valor de 0 a 180
  myservo.write(val); // Configura la posición del servo
  Serial.println(val); // Envá el valor enviado al motor
  delay(15); // Espera un poco a que el servomotor alcance la posición
}
```

Configura el valor del motor de 0 a 180.

Puedes encontrar más información sobre la sentencia la función map en <https://www.arduino.cc/en/Reference/Map>

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroGrados)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int pinPot = 0; // Indica el pin analógico donde va conectado el potenciómetro
int pinServo = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacenará el valor de entrada analógica

void setup()
{
  myservo.attach(pinServo); // Se relaciona el objeto Servo con el pin al que está conectado
  Serial.begin(19200); // Se configura la comunicación serie a 19200 bits por segundo
}

void loop()
{
  val = analogRead(pinPot); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, 0, 180); // Realiza un escalado para que de valores de 0 a 180
  myservo.write(val); // Configura la posición del servo
  Serial.println(val); // Envía el valor por el puerto serie
  delay(15); // Espera 15 ms
}
```

Configura el pulso de control del servo y envía el valor por el puerto serie

Puedes encontrar más información sobre la sentencia la función map en <https://www.arduino.cc/en/Reference/Map>

Moviendo los motores

Ejercicios:

1. Mueve el potenciómetro y apunta los valores entre los que el motor está parado. Calcula el valor medio.
 - Este valor corresponde con el valor en el que servomotor está parado.
2. Haz lo mismo con el otro motor.
 - Escribe, verifica el programa y cárgalo en la placa

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometroMicros)
 - Abre el ejemplo del botón que se llama PulsadorLED.
 - Archivo->Ejemplos->TuBot2016Lib->7_Motor->MotorPotenciometroMicros
 - Verifica y carga el programa en la tarjeta de control
 - Abre el Monitor Serie
 - Mueve el potenciómetro y comprobarás cómo cambia la velocidad de uno de los motores.
 - Este programa es muy parecido al anterior. La diferencia es que el control de la señal de control del servo se hace directamente con los microsegundos de nivel alto.

Moviendo los motores

- Ejemplo: Moviendo los motores como si fueran servomotores (MotorPotenciometro)

```
#include <Servo.h>

Servo myservo; // Crea un objeto de tipo Servo

int potPin = 0; // Indica el pin analógico donde va conectado el potenciómetro
int servoPin = 10; // Indica el pin digital donde va conectado el servomotor
int val; // Variable donde se almacena el valor
```

```
void setup()
{
  myservo.attach(servoPin); // Se relaciona el servo con el pin
  Serial.begin(19200); // Se configura el puerto serial
}
```

```
void loop()
{
  val = analogRead(potPin); // Lee valor del potenciómetro de 0 a 1023
  val = map(val, 0, 1023, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH); // Realiza un escalado para que de valores de 0 a 180
  myservo.writeMicroseconds(val); // Configura la posición del servo
  Serial.println(val); // Envá el valor al puerto serial
  delay(15); // Espera 15 ms
}
```

La función **"map"** realiza una especie de regla de tres. Calcula la proporción de "val" en el rango de 0 a 1023 y devuelve el valor proporcional en el rango de MIN_... A MAX_...

Utiliza la función **Servo.writeMicroseconds()**

Puedes encontrar más información sobre la sentencia la función map en <https://www.arduino.cc/en/Reference/Map>

Moviendo los motores

Ejercicios:

1. Repite lo que hiciste en el ejercicio anterior. Mueve el potenciómetro y apunta los valores entre los que el motor está parado. Calcula el valor medio.
 - Este valor corresponde con el valor en el que servomotor está parado. Este valor se utilizará para calibrar el motor más adelante.
2. Haz lo mismo con el otro motor.
 - Escribe, verifica el programa y cárgalo en la placa

Objeto MotorTubot

- En la librería se define el objeto "MotorTubot" que tiene, entre otros, los siguientes métodos:
 - **MotorTuBot.begin(pinMotor)**
 - Indicar dónde se conecta el motor
 - **MotorTuBot.setZero(zeroValue)**
 - Permite introducir el valor en microsegundos para el que el motor está parado
 - **MotorTubot.setSpeed(velocidad)**
 - Configurar la velocidad de giro del motor entre -100 y 100
 - El motor se para con MotorTubot.setSpeed(0)

Moviendo el Robot

- Ejemplo: Programa que mueve un motor (PruebaMotor)
 - Como siempre el programa tiene tres partes

Es necesario ponerlo para poder utilizar las librerías de los motores

```
#include <Servo.h>
#include <MotorTubot.h>
```

```
// Pines que controlan los motores
int pinMotorIzda = 9;
int pinMotorDcha = 10;
int pinLed = 13;
```

Configuración de los pines donde están conectados los motores y el LED

```
// Declaración del objeto motor
MotorTubot motor;
```

Es necesario definir un objeto de tipo "MotorTubot"

```
void setup(){
  // Inicializa el pin digital como salida.
  pinMode(pinLed, OUTPUT);

  // Inicializa el objeto motor
  motor.begin(pinMotorDcha);
}
```

Moviendo el Robot

- Ejemplo: Programa que mueve un motor (PruebaMotor)
 - Esta vez el programa tiene cuatro partes

```
#include <Servo.h>
#include <MotorTubot.h>

// Pines que controlan los motores.
int pinMotorIzda = 9;
int pinMotorDcha = 10;
int pinLed = 13;

// Declaración del objeto motor
MotorTubot motor;

void setup(){
  // Inicializa el pin digital como salida.
  pinMode(pinLed, OUTPUT);

  // Inicializa el objeto motor
  motor.begin(pinMotorDcha);
}
```

Conecta el objeto "MotorTubot" con un pin concreto.

Moviendo el Robot

- Ejemplo: Programa que mueve un motor (PruebaMotor)
 - Hay una función declarada en una zona para funciones
 - Las funciones pueden ser llamadas desde varios sitios diferentes del programa.
 - Las funciones pueden tener parámetros y devolver valores de retorno.
 - Más información sobre funciones puede encontrarse en:
 - Puedes encontrar más información sobre funciones en este enlace:
 - https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Usos_de_funciones

```
void pulso(void){ // Esta función genera un pulso corto en el led conectado a pinLed
  digitalWrite(pinLed, HIGH); // Enciende el LED
  delay(20);
  digitalWrite(pinLed, LOW); // Apaga el LED
}
```

Moviendo el Robot

- Ejemplo: Programa que mueve un motor (PruebaMotor)
 - Analiza en detalle el movimiento que debería hacer el motor

```
void loop(){
  digitalWrite(pinLed, HIGH); // Motor parado incrementando la velocidad cada 2
  motor.setSpeed(0);          // segundos
  delay(2000);
  digitalWrite(pinLed, LOW);
  motor.setSpeed(50);         // Configura el motor a 50 y hace parpadear el LED
  delay(2000);
  pulso();
  motor.setSpeed(100);        // Configura el motor a 100 y hace parpadear el LED
  delay(2000);

  digitalWrite(pinLed, HIGH); // Motor parado incrementando la velocidad cada 2
  motor.setSpeed(0);          // segundos en sentido contrario
  delay(2000);
  digitalWrite(pinLed, LOW);

  motor.setSpeed(-50);        // Configura el motor a -50 y hace parpadear el LED
  delay(2000);
  pulso();
  motor.setSpeed(-100);       // Configura el motor a 100 y hace parpadear el LED
  delay(2000);
  motor.setSpeed(0);
}
```

Moviendo los motores

□ Ejercicios:

1. Cambia los valores de velocidad del motor para que se aprecien tres velocidades diferentes.
 - Escribe, verifica el programa y cárgalo en la placa
2. Haz un programa con dos objetos del tipo MotorTuBot diferentes y haz que el robot avance durante 2 segundos.
 - Escribe, verifica el programa y cárgalo en la placa
3. Realiza un programa que haga avanzar al robot hasta que se encuentra con un obstáculo, momento en que gira en el sitio durante un rato y sigue avanzando..
 - Escribe, verifica el programa y cárgalo en la placa

Objeto MoverRobot

- En la librería se define el objeto “MoverTubot” que tiene, entre otros, los siguientes métodos:
 - `MoverTubot.begin(int pin1, int pin2);`
 - Indicar dónde se conectan los dos motores
 - `MoverTubot.setZero(int zeroValue1, int zeroValue2);`
 - Permite introducir el valor en microsegundos para el que los motores están parados
 - `MoverTubot.recto(int speedo);`
 - Mueve el robot recto a la velocidad indicada (+100 a -100)
 - `MoverTubot.para(void);`
 - Detiene los motores configurándolos a los valores zeroValue correspondientes.
 - `MoverTubot.dcha(int giro);` `MoverTubot.izda(int giro)`
 - Realiza un giro en el sitio.
 - Puede introducirse un valor entre +100 y -100
 - `MoverTubot.dcha(int giro, int speed);`
 - `MoverTubot.izda(int giro, int speed);`
 - Permite configurar giros y rotaciones

Moviendo el Robot

- Ejemplo: Programa que mueve un motor (MueveTubot)
 - Como siempre el programa tiene tres partes

Es necesario ponerlo para poder utilizar las librerías de los motores

```
//ejemplo que mueve el robot en varias direcciones
```

```
#include <Servo.h>  
#include <MotorTubot.h>
```

```
//Pines de conexión de los motores
```

```
int pinmotorIzda = 9;  
int pinmotorDcha = 10;  
MoverTubot tubot;
```

Configuración de los pines donde están conectados los motores

Es necesario definir un objeto de tipo "MoverTubot"

```
void setup(){
```

```
  // Configuramos los motores y sus pines
```

```
  // pin motor izq || pin motor dcha
```

```
  tubot.begin(pinmotorIzda , pinmotorDcha);
```

```
}
```

Conecta el objeto a los pines correspondientes

Moviendo el Robot

- Ejemplo: Programa que mueve un motor (MueveTubot)
 - Proporciona órdenes de movimiento del robot

```
void loop(){

  tubot.para(); // Parar robot
  delay(1000); // El robot estara parado 1s

  tubot.recto(50); // Mover robot hacia delante
  delay(1000); // Durante 1s

  tubot.dcha(20); // Girar el robot hacia la dcha
  delay(700); // Durante 0.7s

  tubot.recto(-20); // Mover robot hacia atras
  delay(1500); // Durante 1.5s

  tubot.izda(40); // Girar el robot hacia la izda
  delay(500); // Durante 0.5s

}
```

Moviendo el Robot

Ejercicios:

1. Realiza un programa que haga que el robot realice un cuadrado de unos 30cm
 - Escribe, verifica el programa y cárgalo en la placa

Calibración de motores

- Ejecuta el programa de calibración (GetMotorsZero)
 - Abre el monitor serie de Arduino
 - Pulsando las teclas A y D para el motor izquierdo y J y L para el motor derecho, ajusta los motores hasta que se paren.
 - Apunta los valores que se muestran en pantalla cuando estén los motores completamente parados. Estos son los valores de calibración.
 - Vuelve a cargar el ejemplo (motorSimple → motorLeft) añadiendo el método `motorLeft.setZero(valorZero)` siendo `valorZero` el valor obtenido en la calibración.
 - Por ejemplo: **`motorLeft.setZero(1250);`**
- Ejecuta el programa y mira a ver si se comporta de una manera más lógica que antes.
 - Pregunta al profesor si tienes dudas.

Moviendo el Robot

- **Ejercicios para mover un robot en un tatami de Sumo:**
 1. Realiza un programa que haga que el robot avance y se detenga cuando llegue a la línea blanca.
 2. Realiza un programa que haga para que el robot avance hasta que detecte blanco y en ese momento, retroceda durante un cierto tiempo, gire, y siga avanzando. El robot debería moverse dentro del Tatami sin salirse.
 3. Modifica el programa para que, si mientras está girando detecta un oponente, vaya a por él. Este ya es un programa que detecta al oponente.
 4. Modifica el programa para que el robot se mueva por el tatami y si detecta un oponente vaya a por él.

Paso 10.a: Nos movemos sin salirnos

- Teniendo en cuenta todo lo visto anteriormente, realiza los siguientes ejercicios:

□ Ejercicio:

- Realiza un programa que haga que el robot avance y se detenga cuando llegue a la línea blanca.

□ Ejercicio:

- Realiza un programa que haga para que el robot avance hasta que detecte blanco y en ese momento, retroceda durante un cierto tiempo, gire, y siga avanzando.
- El robot debería moverse dentro del Tatami sin salirse.

□ Ejercicio:

- Modifica el programa para que, si mientras está girando detecta un oponente, vaya a por él.
- Este ya es un programa que detecta al oponente