

**CURSO PROVINCIAL:  
"CONTROL Y ROBÓTICA  
EN TECNOLOGÍA"**



**CENTRO DE PROFESORES  
ALBACETE**

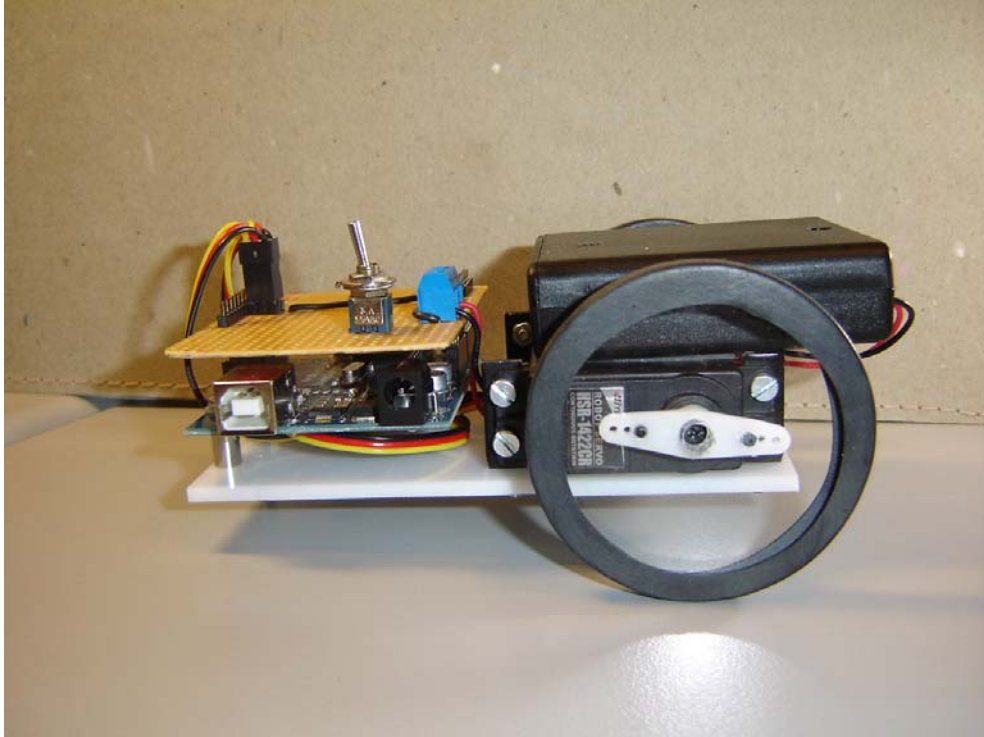
**ROBÓTICA EDUCATIVA  
EducaBot.  
Motores y movilidad.**

**Manuel Hidalgo Díaz  
Antonio Martínez Núñez**

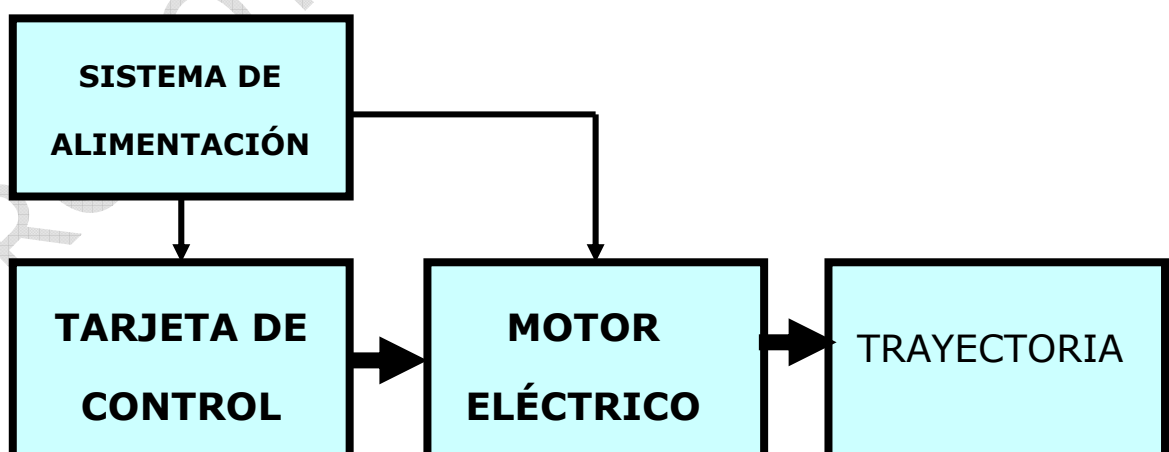
**Noviembre 2009**

## EducaBot 1ª fase.

Terminado la placa de conexi3n y montada junto con la placa de Arduino en la plataforma. El microbot EducaBot tiene que quedar como se ve en la siguiente figura:



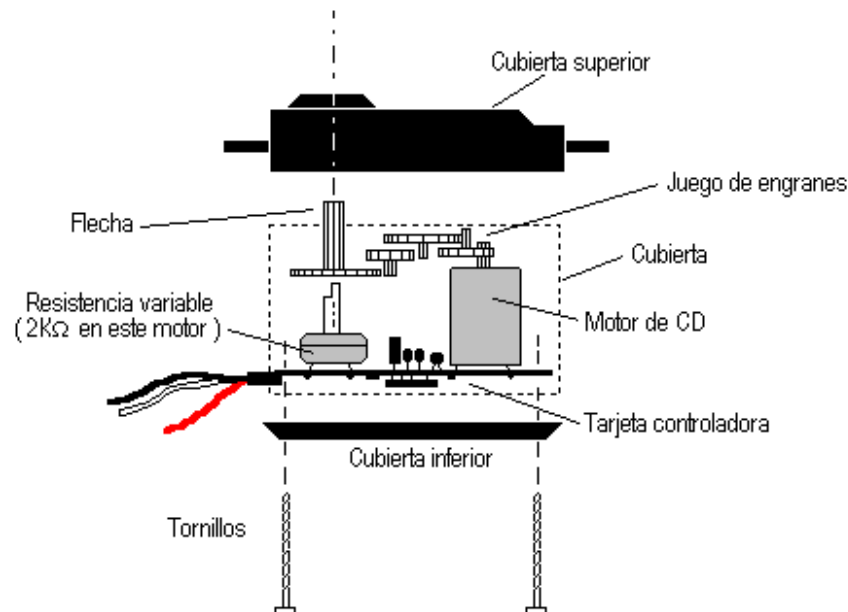
El diagrama de bloques correspondiente es el siguiente:



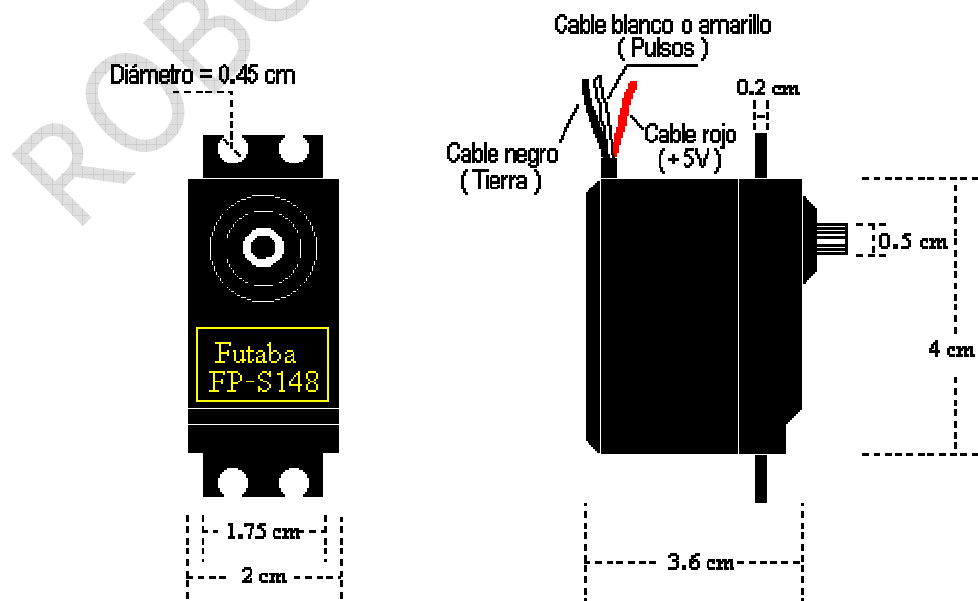
Con este microbot se puede controlar los motores de rotación continua y realizar una secuencia de movimientos.

## Servomotores de rotación continua.

Los motores eléctricos que van a mover a EducaBot son servomotores de rotación continua, giran 360° mientras que 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).



El servomotor de rotación continua 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 a un pin de la tarjeta de control (placa Arduino).

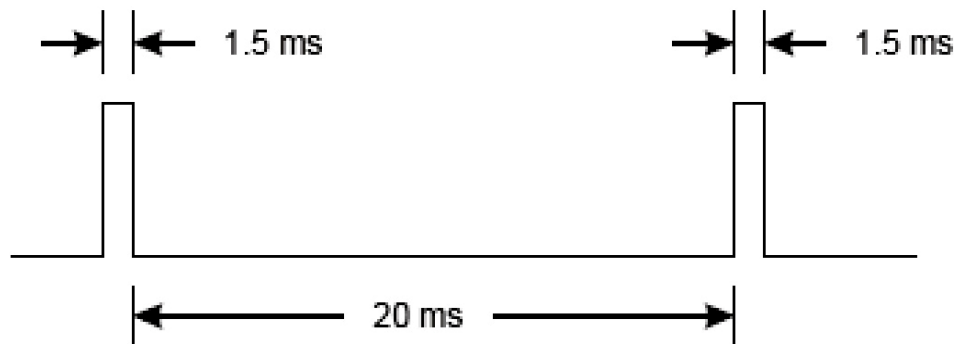


La señal de control de los servomotores de rotación continua es una señal de pulsos modulada en anchura PWM (Pulse Width Modulation). Este tipo de señal de control se utiliza en los servos estándar para realizar los giros desde  $0^\circ$  a  $180^\circ$ . Aquí, va a permitir controlar el giro y la velocidad del eje del servomotor.

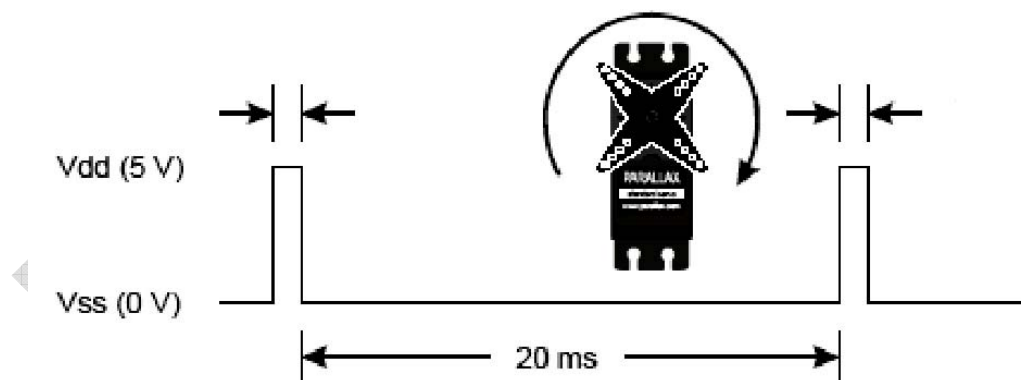
En teoría, porque en la fabricación de los motores eléctricos tiene una tolerancia, con el siguiente pulso de señal el servomotor de rotación continua tendría que estar parado. Para un servo la posición del eje sería de  $90^\circ$ .

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:

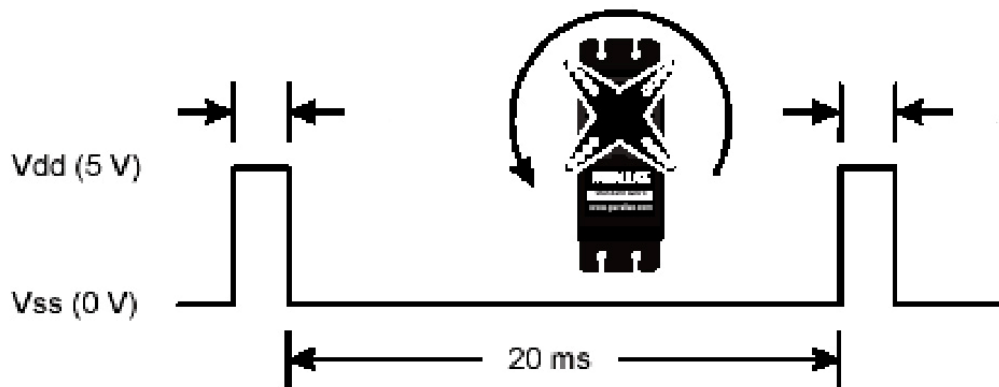


Señal con periodo de 20 ms.

Anchura del pulso entre 0.8 a 1.2 ms.

Aquí hay un rango de valores porque depende del fabricante del servomotor de rotación continua.

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

Igual que en el caso anterior la anchura del pulso depende del fabricante del servomotor de rotación continua. 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.
<b>Futaba</b>	0.9	1.5	2.1	50	rojo	negro	blanco
<b>Hitec</b>	0.9	1.5	2.1	50	rojo	negro	amarillo
<b>Graupner/Jr</b>	0.8	1.5	2.2	50	rojo	marrón	naranja
<b>Multiplex</b>	1.05	1.6	2.15	40	rojo	negro	amarillo
<b>Robbe</b>	0.65	1.3	1.95	50	rojo	negro	blanco
<b>Simprop</b>	1.2	1.7	2.2	50	rojo	azul	negro

En el microbot EducaBot se utiliza la librería **<Servo.h>** que tiene el entorno de programación de Arduino para controlar los servomotores de rotación continua.

A continuación vamos a realizar unos programas para ajustar la velocidad cero, realizar secuencia de movimientos y la curva que relaciona la señal PWM con la velocidad.

## Ajuste de velocidad.

El siguiente programa permite ajustar la velocidad de los servomotores de rotación continua a cero.

Cambiando los valores de las variables **vel\_Sder** y **vel\_Sizq** se ajusta la velocidad. En teoría la velocidad cero del servomotor sería con el valor de 90.

```
// PruebaServomotores01
// "Curso Control y Robotica"
// CEP de ALBACETE
// Manuel Hidalgo Diaz
// Noviembre 2009

// Programa para ajustar la velocidad de los servomotores de rotacion continua
#include <Servo.h>

Servo servo_der; // crea los objetos para controlar los servomotores
Servo servo_izq;

int vel_Sder = 90; // variable para la velocidad del servomotor (0 a 180)
int vel_Sizq = 90; // 0 --> velocidad maxima en un sentido
                    // 180 --> velocidad maxima en sentido contrario

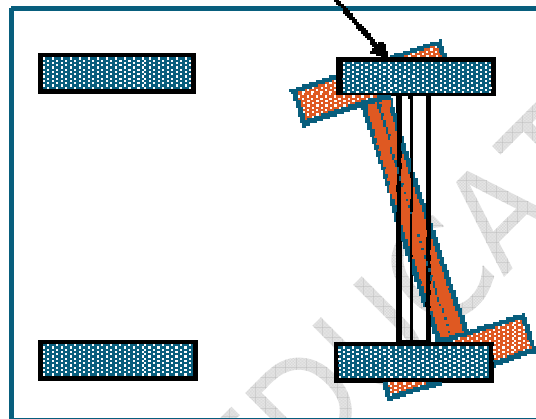
void setup()
{
    servo_der.attach(9); // asocia el pin 9 al servo_der
    servo_izq.attach(10); // asocia el pin 10 al servo_izq
}

void loop()
{
    servo_der.write(vel_servo_der); // Da al servo_der la velocidad en la variable vel_servo_der
    servo_izq.write(vel_servo_izq); // Da al servo_izq la velocidad en la variable vel_servo_izq
}
```

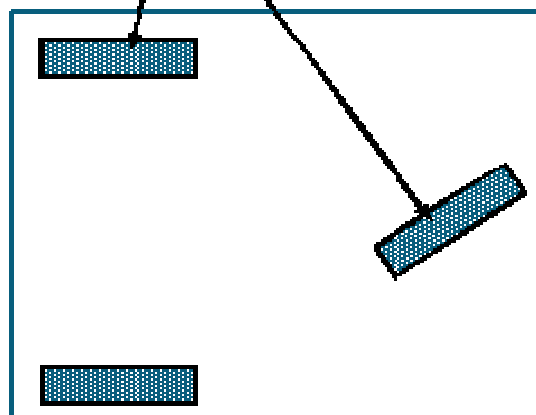
## Movilidad de EducaBot.

En la movilidad de los microbots se puede adoptar varias opciones de movilidad, entre ellas están las siguientes:

**ESTRUCTURA DE COCHILLO**  
Tracción y dirección en  
dos ruedas

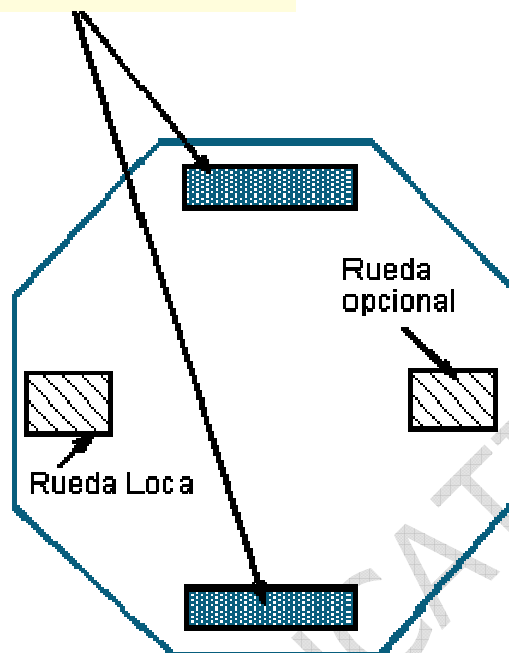


**ESTRUCTURA DE TRICICLO**  
Tracción y dirección  
independientes

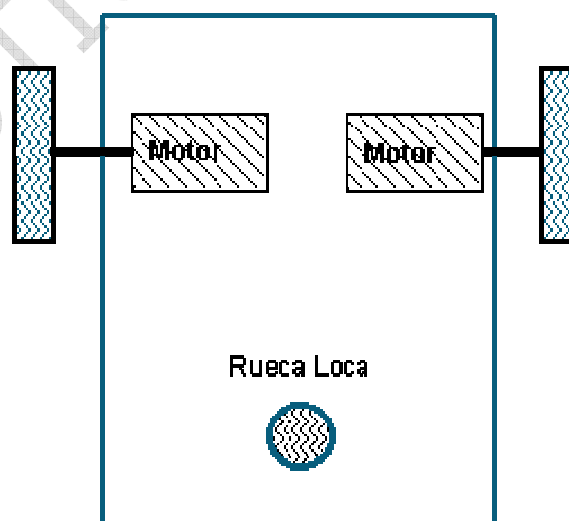


### DIRECCIÓN DIFERENCIAL

Ruedas de tracción

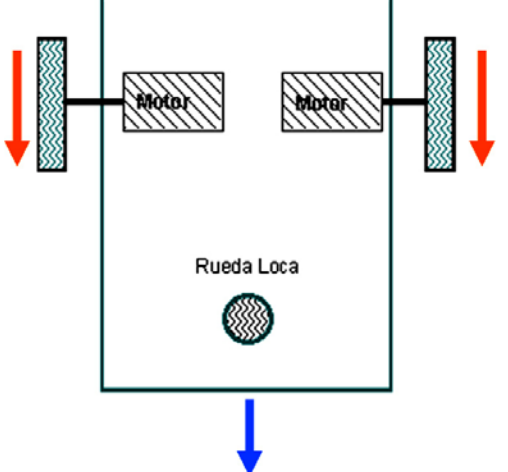
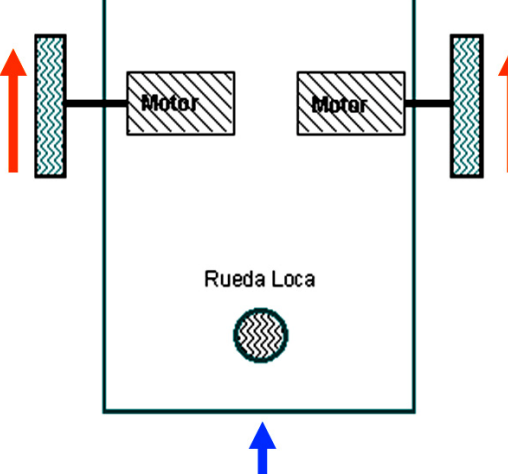
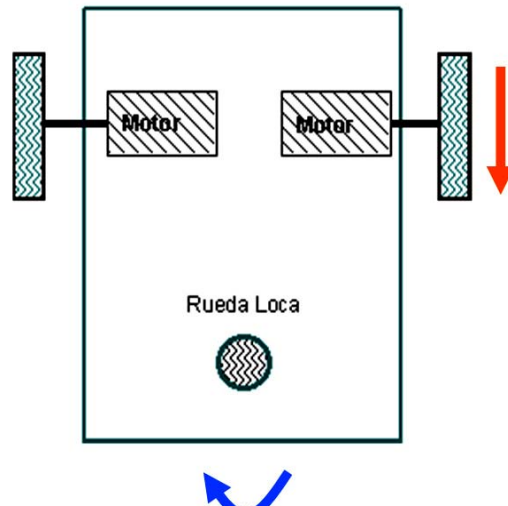


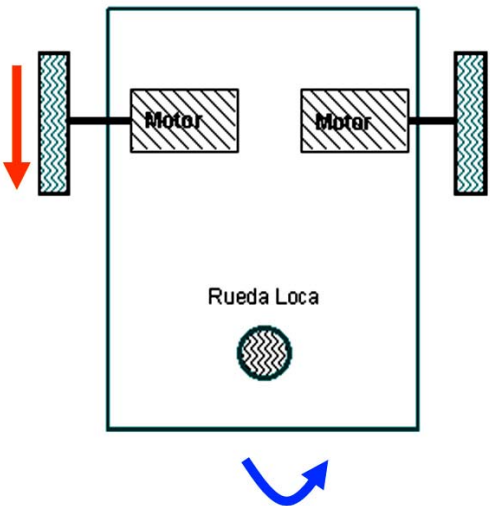
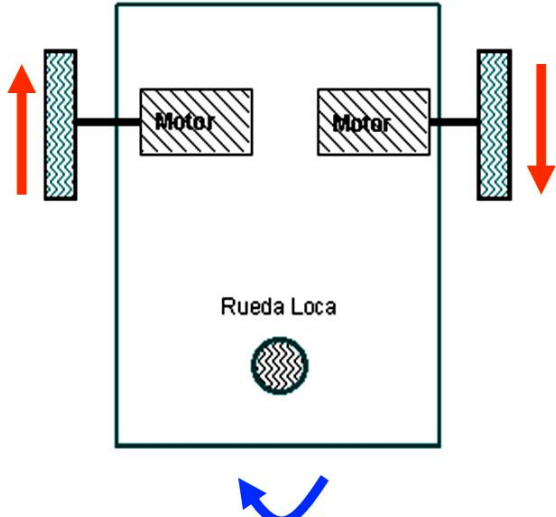
### ESTRUCTURA DE NUESTRO MICROBOT



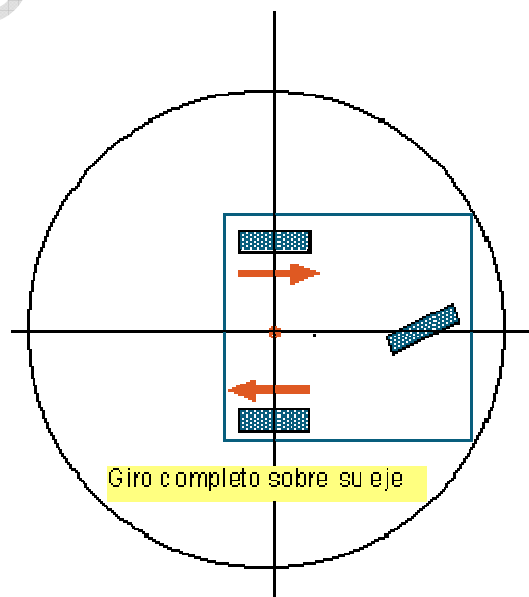


La estructura del microbot EducaBot permite realizar los siguientes movimientos:

	<p><b>1.-</b> Movimiento hacia delante, se hacen girar los dos motores en la misma dirección hacia delante (sentido horario) . Esto provoca un movimiento rectilínea suponiendo que los dos motores son exactamente iguales.</p>
	<p><b>2.-</b> Movimiento hacia atrás, se hacen girar los dos motores en la misma dirección hacia atrás (sentido contrario a la agujas del reloj). Esto provoca un movimiento rectilíneo suponiendo que los dos motores son exactamente iguales.</p>
	<p><b>3.-</b> Movimiento de giro a la derecha, se hacen girar el motor izquierdo en sentido horario (hacia delante) y el motor de la derecha en sentido antihorario (hacia atrás) o parado. Esto provoca un movimiento de giro a la derecha de la estructura</p>

	<p>4.- Movimiento de giro a la izquierda, se hacen girar el motor izquierdo en sentido antihorario (hacia atrás) o parado y el motor de la derecha en sentido horario (hacia adelante). Esto provoca un movimiento de giro a la izquierda de la estructura</p>
	<p>5.- El movimiento de giro completo sobre su propio eje, abarca una superficie muy grande que hace la estructura muy adecuada para moverse en recintos muy pequeños, como podría ser el caso de movimientos en pruebas de laberintos.</p>

En los movimientos de giro es muy importante conocer el punto sobre el cual gira la estructura del microbot.



El siguiente programa va a permitir mover el microbot con una secuencia de movimientos preprogramada.

Los pasos para adaptar este programa al microbot que se ha construido son los siguientes:

- 1.- Tener el valor de la velocidad cero de cada uno de los servomotores obtenido con el programa anterior.
- 2.- En el programa hay una nueva función denominada **mueve\_motores(int vel\_der, int vel\_izq)** que admite dos valores entre 100 y -100.

Estos valores corresponden a las velocidades de los servomotores según los valores asignados:

100 → corresponde a la máxima velocidad de avance.

0 → corresponde a parada.

-100 → corresponde a la máxima velocidad hacia atrás.

- 3.- Dentro de la función

**mueve\_motores(int vel\_der, int vel\_izq)**

está la instrucción de mapeo

**motor\_der.write(map(vel\_der, -100, 100, 0, 180))**

para realizar es escalado de valores correspondientes a la velocidad que hay que dar al servomotor.

En esta instrucción se tiene que ajustar los valores de 0 y 180 a la desviación que hemos tenido en el ajuste de velocidad cero de los servomotores.

En el caso que en el ajuste de velocidad cero se haya obtenido con el valor de 90, los valores están correctos.

- 4.- Realiza la comprobación de una trayectoria de 2 m a una velocidad del 50%.

```
// moverMotores

// "Curso Control y Robotica". CEP de ALBACETE

// Manuel Hidalgo Diaz. Noviembre 2009

// Programa para realizar una secuencia de movimientos

#include <Servo.h>

// Declaracion de funciones

void mueve_motores(int vel_a, int vel_b);

// Variables del programa

Servo motor_izq; // Servo de la rueda izquierda

Servo motor_der; // Servo de la rueda derecha

// Inicializacion

void setup()

{

    // Inicializa servos

    motor_der.attach(9); // Asocia el servo del motor derecho a la salida PWM 9

    motor_izq.attach(10); // Asocia el servo del motor izquierdo a la salida PWM 10

    // Detiene el robot

    mueve_motores(0,0);

}
```

```
// Programa principal

void loop()

{
    mueve_motores(50,50);
    //delay(10);
}

//Funciones

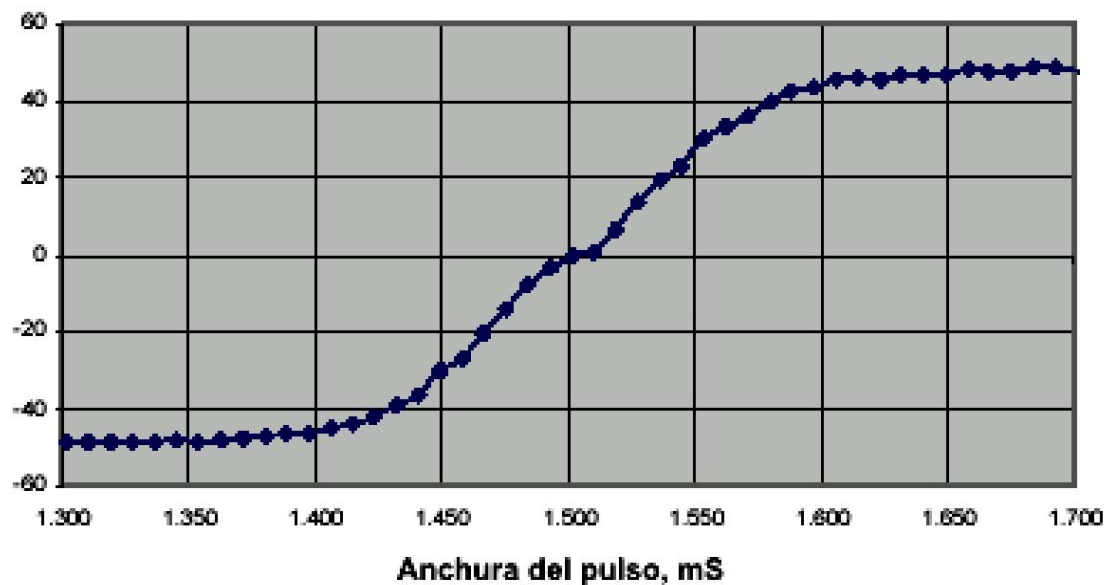
// Mueve los motores en el rango de -100 a 100
void mueve_motores(int vel_a, int vel_b)
{
    motor_der.write(map(vel_a, -100, 100, 0, 180));
    motor_izq.write(map(vel_b, 100, -100, 0, 180));
}
```

## Gráfica de respuesta velocidad-PWM.

En la programación de secuencias preprogramadas hay que tener un conocimiento de la velocidad en función de pulso PWM aplicado.

Una forma de conocer el valor de estos parámetros es realizando una gráfica para obtener la relación entre ambas.

**Control de velocidad mediante la anchura del pulso aplicado al servo**



Orientaciones para hacer una tabla de valores:

- Cambiar el valor de pulso (%) en incrementos de 10.
- La duración de cada medida 6 s.

PWM (%)	Velocidad (RPM)	PWM (%)	Velocidad (RPM)	PWM (%)	Velocidad (RPM)	PWM (%)	Velocidad (RPM)
100		50		-50		-100	
90		40		-40		-90	
80		30		-30		-80	
70		20		-20		-70	
60		10		-10		-60	
		0					

## Conclusiones de la actividad de movilidad.

El microbot EducaBot utiliza la librería **<Servo.h>** que tiene el entorno de programación de Arduino para controlar los servomotores de rotación continua. Esta librería está pensada para controlar servomotores estándar (giro-posicional desde 0° a 180°) con una señal de control PWM.

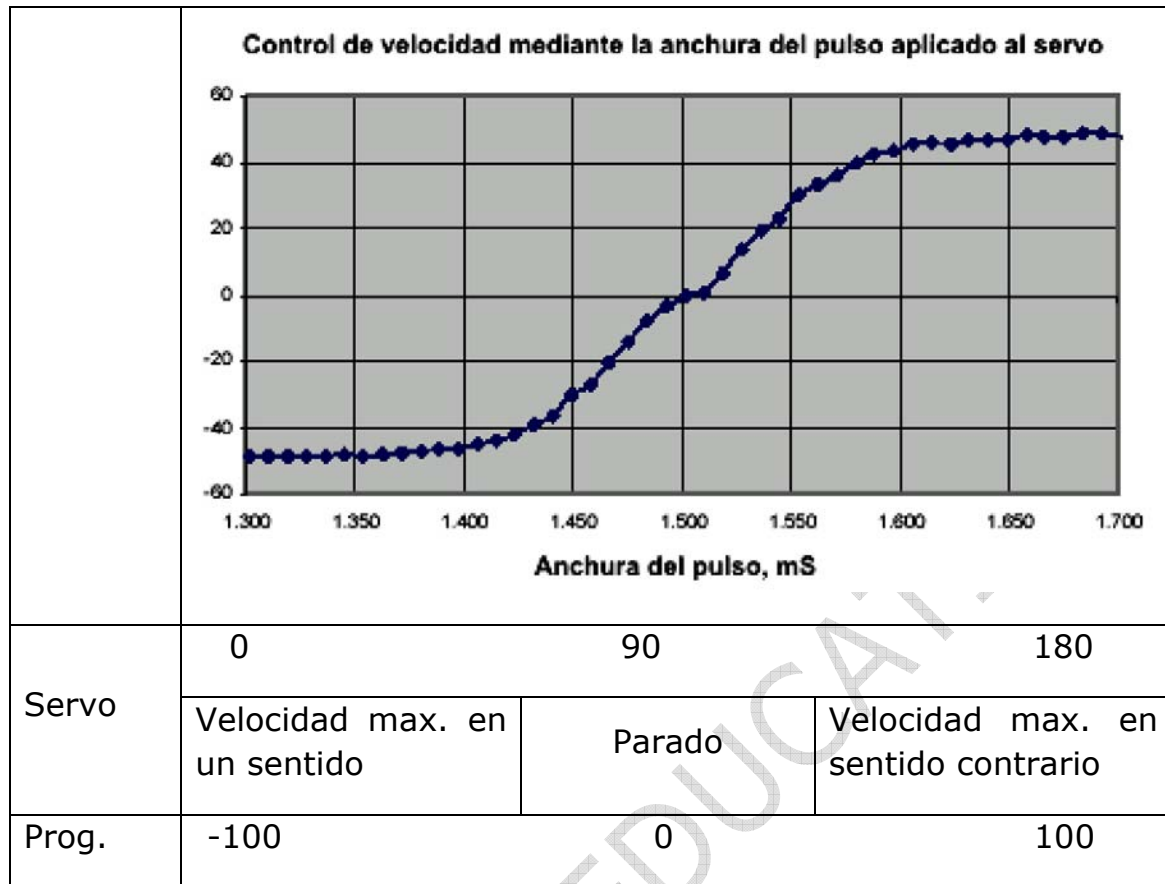
Conociendo el servomotor de rotación continua cuyo control también es con una señal PWM se establece una correspondencia en el control utilizando la librería **<Servo.h>** de la siguiente manera:

Servomotor estándar	0°	90°	180°
Servomotor rt. continua	Velocidad max. en un sentido	Parado	Velocidad max. en sentido contrario

También utilizamos en el programa **moverMotores** la instrucción de mapeado para adaptar el valor de la velocidad a un lenguaje natural.

Servomotor estándar	0°	90°	180°
Servomotor rt. continua	Velocidad max. en un sentido	Parado	Velocidad max. en sentido contrario
Programa	-100	0	100

Realizando las pruebas con el microbot comprobamos que estos valores no se cumplen, la curva de respuesta de los motores (velocidad-pulso) es la siguiente:



Se observa que a partir de un valor la velocidad se mantiene (en nuestra prueba es la distancia recorrida en 6s). Por tanto, es conveniente **adaptar la instrucción de mapeado** a los valores reales y siempre simétricos al valor de la velocidad cero.

NOTA.- si las pruebas de velocidad se hacen con incrementos de 5 o 1, la precisión del ajuste será mejor.