

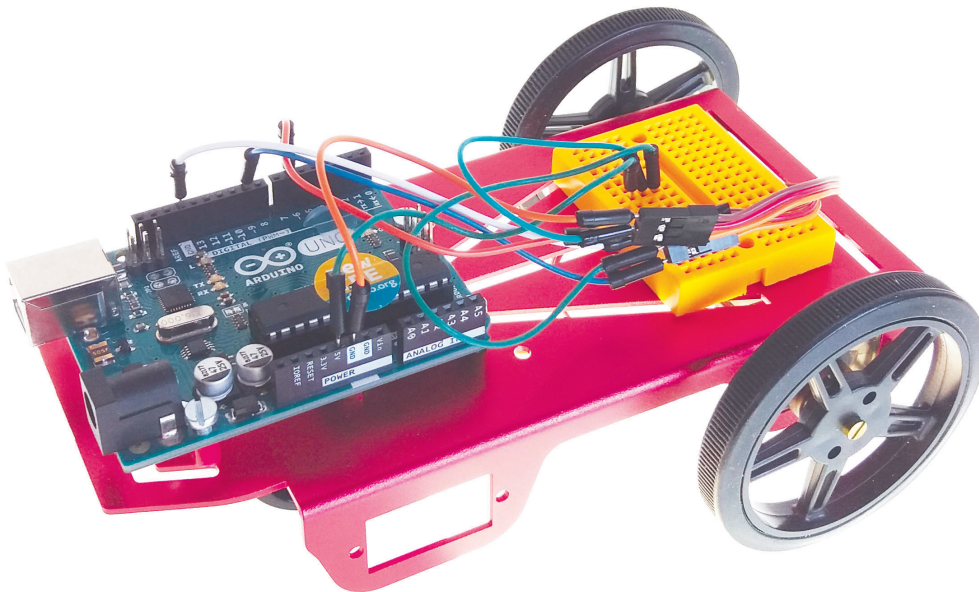
SERVO COCHE

MICRO-LOG[®]
LOGKIT
4208A

1. OBJETIVOS

Construir un coche robótico con servomotores de rotación continua y aprender a controlar los servomotores.

2. FOTOGRAFÍA



3. FUNCIONAMIENTO

El coche se programa con processing o mBlock, accionando los miniservos para que siga una ruta preestablecida.

4. LISTA DE MATERIALES

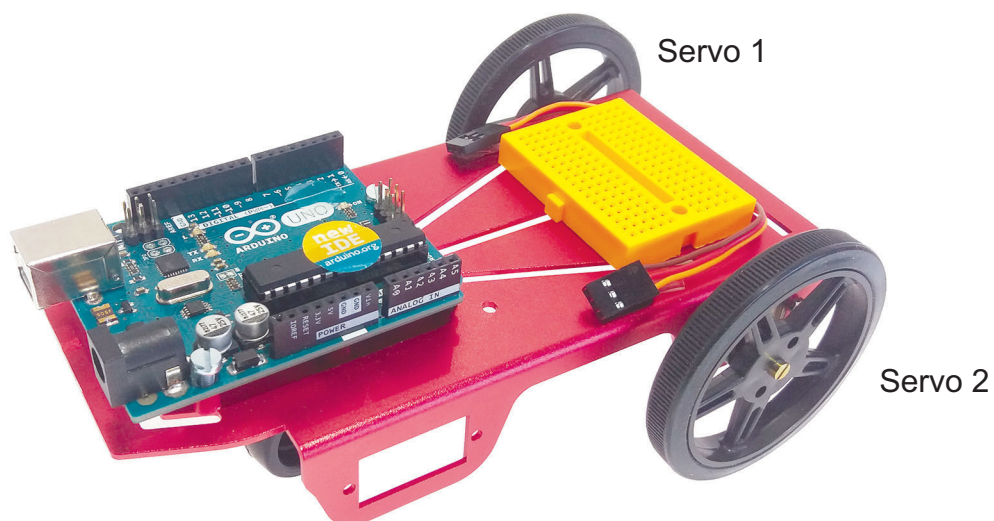
- 1 Arduino UNO REV-3 LOG 4031
- 1 Cable USB A-B LOG 4009
- 1 Plataforma móvil LOG 4080E
- 1 Miniboard LOG 885
- 1 Conector 9V para Arduino LOG 7734
- 1 Conjuntos de 10 latiguillos LOG 9519
- 4 Tornillos M3x16 mm LOG 464M
- 4 Tuercas M3 LOG 480
- 4 Bulones de plástico S 220P
- 1 Hoja Técnica H4208A

Leer todas las instrucciones y comprobar el listado de materiales antes de empezar el proyecto.

SERVO COCHE

5. CONSTRUCCIÓN

- a) Montar la plataforma móvil según sus instrucciones.
- b) Atornillar la placa Arduino UNO en la plataforma, utilizando los bulones de plástico, los tornillos y las tuercas.
- c) Pegar la miniboard en la parte trasera.



6. CONEXIÓN DEL CONTROL DE MOTORES

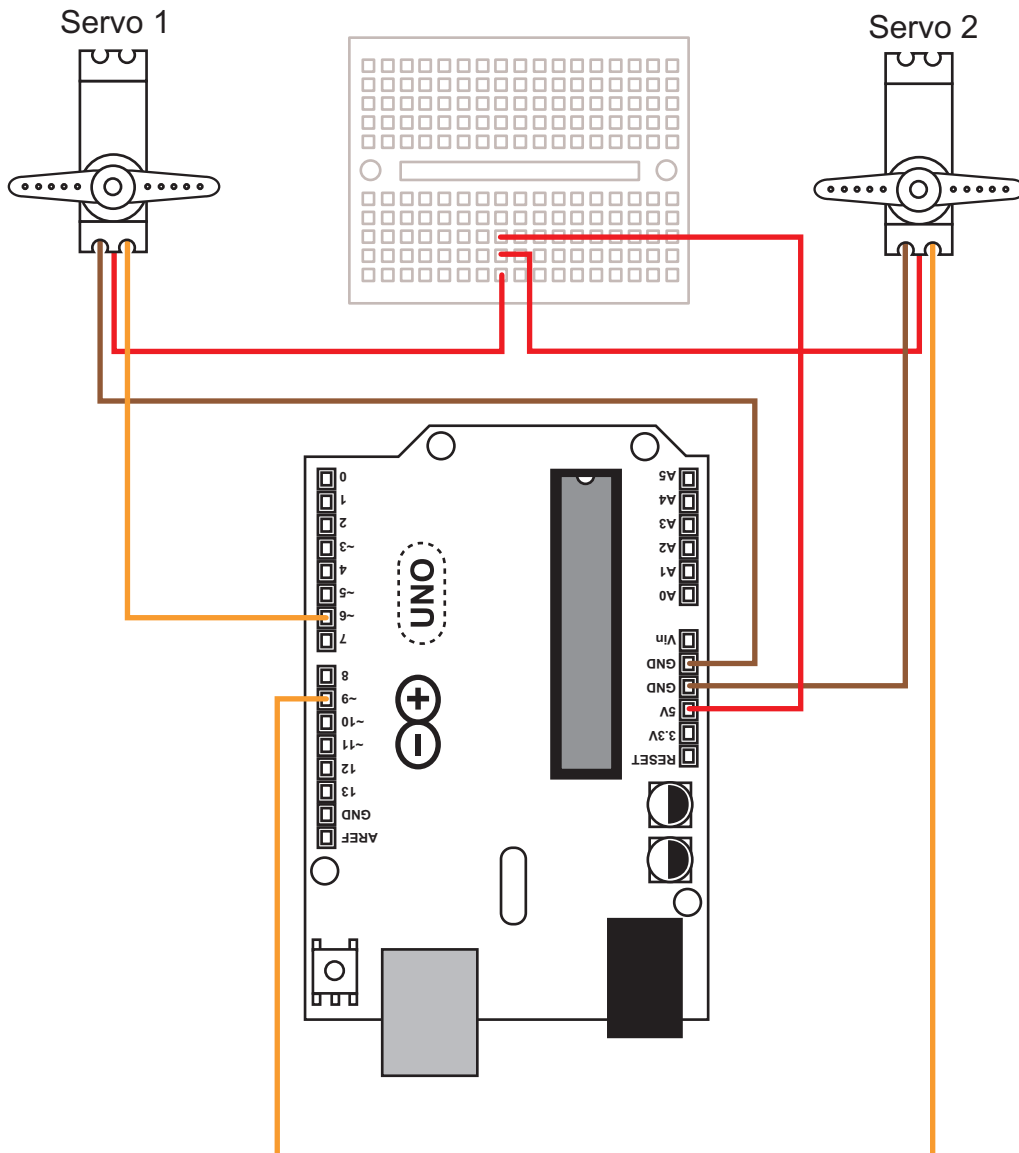
Cada miniservo lleva 3 cables que conectaremos de la siguiente manera:

Marrón ó negro (GND): debemos conectar el cable utilizando un latiguillo a GND de la placa Arduino.

Rojo (5V): debemos conectar el cable rojo a un pin de la placa board. Los cables rojos de los motores deben quedar conectados con latiguillos a una misma columna de la placa board.

Naranja (datos): conectamos el cable utilizando un latiguillo del miniservo 1 al pin 6 de arduino y el otro al pin 9.

A continuación conectamos un último latiguillo desde la columna de los latiguillos del cable rojo hasta el pin 5V de la placa arduino.



7. PROGRAMACIÓN EN MBLOCK

Makeblock es una herramienta de programación por bloques basado en scratch.

Para instalar el software de Makeblock, entramos en makeblock.micro-log.com y desde el menu de "descargas" descargar el software mBlock.

Una vez instalado, abrir "mBlock":

- Pinchar en la pestaña "Placas", y pinchar donde pone "Arduino UNO".
- Pinchar en la pestaña "Conectar", seleccionar donde pone "Puerto serie" y pinchar donde pone "COM*Numero*".

Una vez realizada la conexión con la tarjeta ya podemos programar.

Los motores se han conectado a los pines 6 y 9 que son salidas de tipo PWM. Esto significa que la señal que les podemos enviar varía. En el caso de estos motores, podemos enviar una señal que va desde 0 hasta 254, y en función de este valor cambiará el comportamiento del motor.

Para programar el movimiento del motor con mBlock utilizamos la siguiente instrucción:

fijar pin PWM (Pin del motor) a (velocidad)

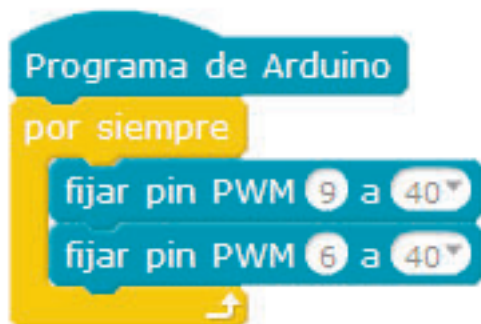
Donde velocidad puede adquirir los siguientes valores:

0 / 180 / 254 - Con cualquiera de estos valores el motor permanece parado.

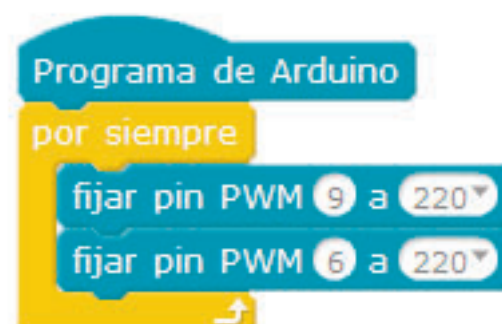
De 1 a 179 - El motor avanza en un sentido de giro siendo 1 la mayor velocidad y 179 la menor.

De 181 a 254 - El motor avanza en sentido contrario, siendo 181 la menor velocidad y 253 la mayor.

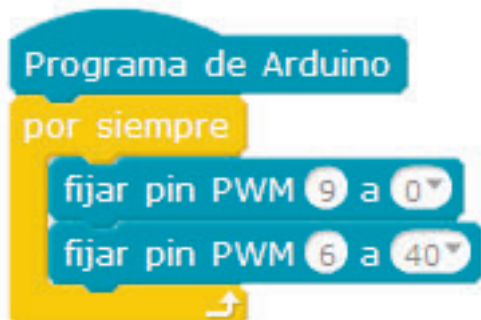
AVANCE HACIA DELANTE:



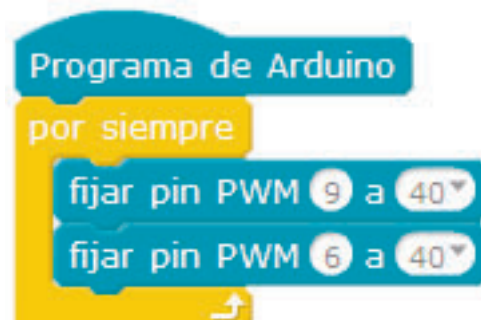
AVANCE HACIA ATRÁS:



AVANCE HACIA LA IZQUIERDA:



AVANCE HACIA LA DERECHA:



8. PROGRAMACIÓN EN ARDUINO IDE

Arduino IDE es una herramienta de programación por códigos basado en C++.

Para instalar el software de Arduino, entramos en www.arduino.cc/en/Main/Software y descargamos el software Arduino IDE.

Una vez instalado, abrir "Arduino IDE":

- Pinchar en la pestaña "Herramientas", seleccionar donde pone "Placa" y pinchar en "Arduino/Genuino UNO".
- Pinchar en la pestaña "Herramientas", seleccionar donde pone "Puerto" y pinchar en "COM*Numero* Arduino/Genuino UNO".

Una vez realizado la conexión con la tarjeta ya podemos programar.

Los motores se han conectado a los pines 6 y 9 que son salidas de tipo PWM. Esto significa que la señal que les podemos enviar varía. En el caso de estos motores, podemos enviar una señal que va desde 0 hasta 254, y en función de este valor cambiará el comportamiento del motor.

Para programar el movimiento del motor con arduino IDE utilizamos la siguiente instrucción:

```
analogWrite (PIN, Velocidad);
```

Donde velocidad puede adquirir los siguientes valores:

0 / 180 / 254 - Con cualquiera de estos valores el motor permanece parado.

De 1 a 179 - El motor avanza en un sentido de giro siendo 1 la mayor velocidad y 179 la menor.

De 181 a 254 - El motor avanza en sentido contrario, siendo 181 la menor velocidad y 253 la mayor.

AVANCE HACIA DELANTE:

```
void setup() {  
  
    pinMode(9, OUTPUT);  
    pinMode(6, OUTPUT);  
  
}  
  
void loop() {  
  
    analogWrite(6, 40);  
    analogWrite(9, 40);  
  
}
```

Si el vehículo no avanza recto, jugaremos con la velocidad del los motores variando el parámetro que en este programa hemos marcado como 40.

SERVO COCHE**AVANCE HACIA ATRÁS:**

```
void setup() {  
    pinMode(9, OUTPUT);  
    pinMode(6, OUTPUT);  
}
```

```
void loop() {  
    analogWrite(6, 220);  
    analogWrite(9, 220);  
}
```

Si el vehículo no avanza recto, jugaremos con la velocidad de los motores variando el parámetro que en este programa hemos marcado como 220.

GIROS: Para que el vehículo gire, paramos un motor y hacemos girar el otro motor. Otra opción es invertir los sentidos de giro o variar las velocidades para que un motor avance más rápido que el otro

AVANCE HACIA LA IZQUIERDA:

```
void setup() {  
    pinMode(9, OUTPUT);  
    pinMode(6, OUTPUT);  
}
```

```
void loop() {  
    analogWrite(6, 40);  
    analogWrite(9, 0);  
}
```

AVANCE HACIA LA DERECHA:

```
void setup() {  
    pinMode(9, OUTPUT);  
    pinMode(6, OUTPUT);  
}
```

```
void loop() {  
    analogWrite(6, 0);  
    analogWrite(9, 40);  
}
```

SERVO COCHE



LOGKIT
4208A

9. CALIBRACIÓN

- Aunque ambos motores reciban la misma señal, es muy habitual observar que el vehículo no vanza recto. Es preciso ajustar la velocidad afinándola dentro del programa. Ajustamos manualmente hasta dar con la combinación de velocidad que genere un movimiento rectilíneo.

mBlock:

fijar pin PWM (Pin del motor) a (velocidad)

Arduino IDE:

analogwrite (motor, velocidad);

10. DETALLES DE TIPO PRÁCTICO

- Desconectar la alimentación cuando no se este utilizando para no gastar las pilas.
- Necesita 1 pila 6F22 de 9 V.
- Tiempo de construcción: 1 H.
- Nivel: Fácil

11. PRUEBAS

- Si el coche no avanza correctamente, comprobar la carga de la pila.
- Realizar combinaciones de movimiento para tener un recorrido más completo.