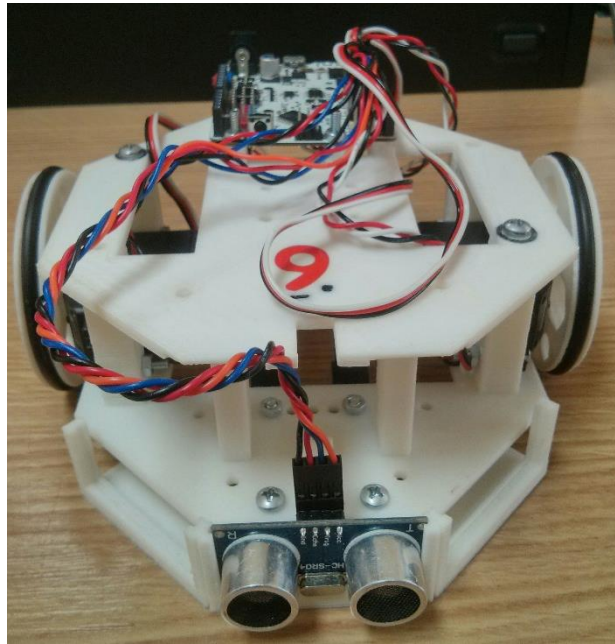


Robot Móviles – Sigue Líneas



Práctica 7 – Teoría (v1.6 octubre 2022)

Software para robots

Cristian González García

gonzalezcristian@uniovi.es

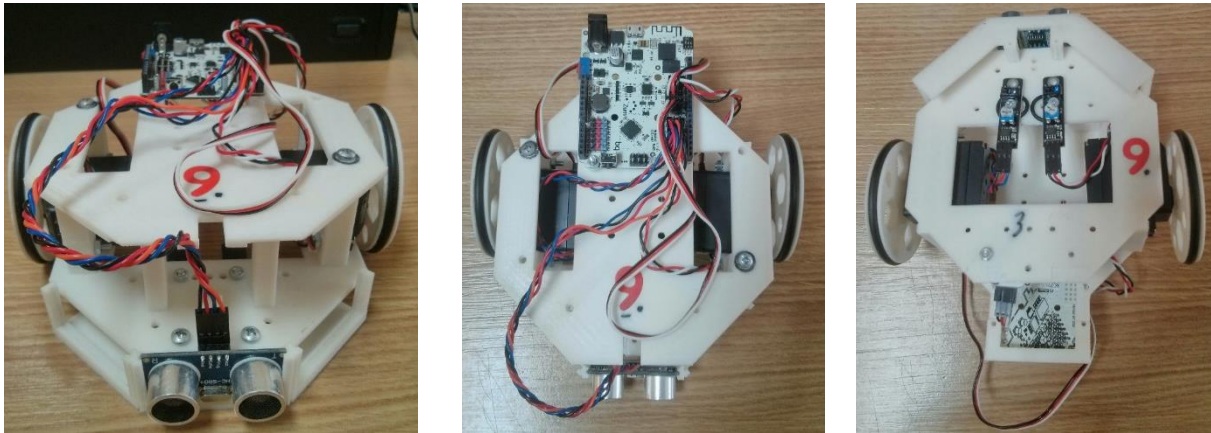
Basado en el material original de Jordán Pascual Espada

Índice

1.	Introducción	2
2.	Conceptos.....	2
3.	Avanzar y retroceder	5
4.	Cambios de dirección	6
	Cambio de dirección con movimiento	6
	Cambio de dirección sobre el propio eje	7
5.	Detección de línea.....	7

1. Introducción

El objetivo de esta práctica es trabajar con robots móviles. Para ello, se utilizará el siguiente robot de tipo móvil:



Podéis probarlo fuera del aula de prácticas con el simulador de la asignatura que se encuentre en el campus virtual junto a su manual.

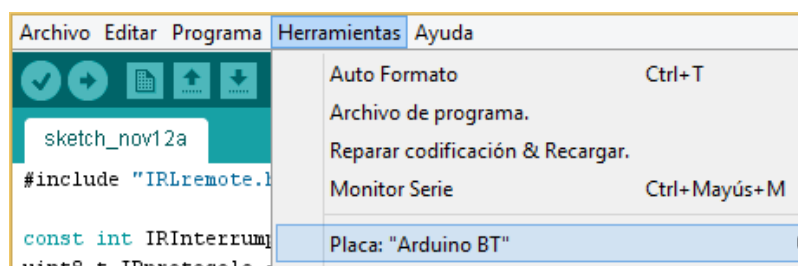
2. Conceptos

El robot se compone de:

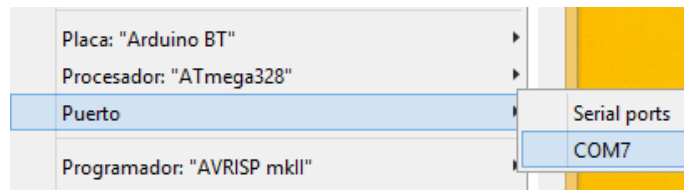
- 2 servomotores de rotación continua (ruedas)
- 1 sensor de ultrasonidos para la distancia
- 2 sensores infrarrojos de detección de línea
- 1 placa ZUM BT-328 (basada y compatible con Arduino BT)

La placa:

- **Tiene una palanca para encender/pagar el robot en un lateral (Roja y Azul)**
 - Encendido: hacia el enchufe. En este modo el robot funciona con la lógica que tenga almacenada.
 - Apagado: hacia los pines. En este modo el robot no se mueve, pero puede recibir nueva lógica. Asegurarse de que se encuentra en este modo para que no empiece a andar sobre la mesa y se pueda subir el código a la palca sin problemas.
- La **conexión** a la placa se realiza por **micro USB-B** en vez del por USB tipo B como en el Arduino.
- Debemos configurar el tipo de placa en Arduino BT, sino no subirá los programas.



- No suele indicar el nombre de la placa en el puerto, no obstante, sí que puede estar conectada igualmente.



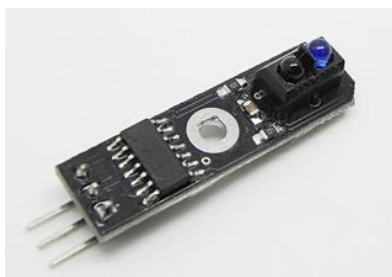
- Suele tardar unos minutos en subir los programas, más que la placa Arduino Uno.
- Tiene Bluetooth integrado.

Esquema de los componentes:

Componente	Conexión (Pin digital)
Sensor de línea central-izquierdo	2
Sensor de lineal central-derecho	3
Servo izquierdo	8
Servo derecho	9
Ultrasonidos Trig	4
Ultrasonidos Echo	5

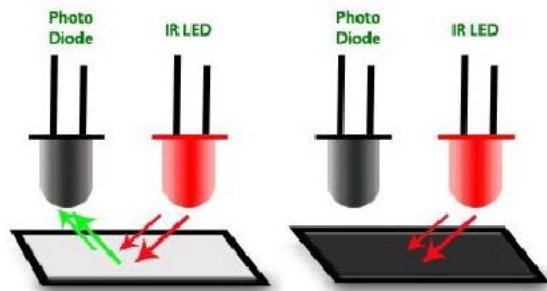
Importante: vamos a respetar siempre las posiciones de los pines que se muestran en el siguiente esquema. El objetivo es que todos los robots estén conectados de igual manera y el código pueda ser probado en cualquiera de ellos. De esta forma, si cambiáis de robot no tendréis que preocuparos por los pines.

Componentes que pueden requerir calibración:



Sensores infrarrojos de detección de línea: el sensor emite luz infrarroja de forma constante, utilizando el LED IR transparente o azul, y por medio del fotodiodo infrarrojo recibe dicha onda, que hace que se active (LOW o 0) si la luz infrarroja «rebota» contra algo y vuelve al sensor, que suele pasar cuando es de color claro o blanco, o mismamente el dedo. Devuelve «HIGH» o 1 en el caso de que no rebote, por ejemplo, una superficie oscura o negra impedirá dicho rebote, o bien, si lo dejáis contra el aire boca arriba. Para calibrarlo, se puede usar algo blanco para simular la ausencia línea, como un folio. Su rango de funcionamiento es entre 5 y 20 mm.

El negro es el receptor, que es un fotodiodo que recibe la luz reflejada, y el azul es un LED de luz infrarroja y hace de emisor.



(equivalente al camino/negro).

Se deben regular utilizando la mano en el regulador que poseen. No hace falta destornillador ya que se gira fácilmente. Para ello, probad que se enciende el LED rojo del sensor y envía un 0 (LOW) cuando lo tapáis con el dedo (equivalente al suelo/blanco), y que la luz del LED esté apagada y envía un 1 (HIGH) el sensor cuando lo dejáis boca arriba destapado

Probad también que **ambas luces de ambos sensores se encienden casi a la vez cuando los reguláis**. Esto es muy importante para que el robot esté equilibrado y se mueva correctamente y no salga más o tarde más en responder de un lado que del otro cuando se sale del circuito.

¡Cuidado! Según el modelo, el sensor podría funcionar a la inversa y devolver el 0 y 1 intercambiados a como se han explicado.

Los colores claros hacen rebotar la onda y los oscuros la absorben. Si la onda no rebota contra nada, el receptor no recibirá ninguna señal, que será lo mismo que contra un color oscuro.



Servomotores de rotación continua: si al enviar la señal de «detenido» un servomotor continúa moviéndose significa que está mal calibrado.

Aquí hay que tener mucho cuidado, pues **tienen que estar los dos perfectamente calibrados**. Es decir, que se envíe un 90 a ambos y que ninguno se mueva o haga ruido. En caso contrario, el robot no avanzará en línea recta, sino torcido.

Batería: esta batería va colocada dentro del robot, de forma que quede encajada en su parte interior, entre un motor y la carcasa superior e inferior. **Hay que enchufar el USB a la salida de 1A.**



Se puede utilizar casi cualquiera que sirva para cargar el móvil siempre y cuando concuerde con los voltajes del Arduino (7-12V y 0,5-1A recomendado). En caso de quedarse sin batería, se podría utilizar también un cable USB lo suficientemente largo para alimentarlo. No obstante, siempre teniendo cuidado de que este cable llegara a todos lados o siguiendo al robot con el portátil (no recomendado).

3. Avanzar y retroceder

Para avanzar y retroceder debemos mover los dos servomotores que controlan las ruedas en el mismo sentido. Para eso, uno tiene que girar de forma horaria y el otro de forma antihoraria, ya que los vemos desde arriba y por la forma en cómo están montados los servomotores.

Es **muy importante calibrar los dos servomotores** muy bien para que el robot avance en línea recta.

Dependiendo de la señal que se les envíe, los servomotores se pueden moverse a mayor o menor velocidad.

```
#include <Servo.h>

Servo servoLeft;
Servo servoRight;

int pinServoLeft = 8;
int pinServoRight = 9;

void setup(){
  servoLeft.attach(pinServoLeft);
  servoRight.attach(pinServoRight);
}
```

```

void loop(){
  // Atrás
  servoLeft.write(180); // Velocidad Máxima
  servoRight.write(0);
  delay(3000);
  servoLeft.write(100); // Próximo a 90, despacio
  servoRight.write(80);
  delay(3000);

  servoLeft.write(90); // Detener
  servoRight.write(90);
  delay(3000);

  // Adelante
  servoLeft.write(80); // Próximo a 90, despacio
  servoRight.write(100);
  delay(3000);
  servoLeft.write(0); // Velocidad Máxima
  servoRight.write(180);
  delay(3000);
}

```

4. Cambios de dirección

En nuestro caso, el robot funciona con **locomoción diferencial**: no tiene ruedas directrices como un coche, solamente tiene ruedas motrices. Dependiendo del sentido, y de la velocidad de giro y sentido de las ruedas, se puede cambiar la dirección del robot. Luego, hay varias opciones para hacer girar al robot.

Cambio de dirección con movimiento

Una rueda permanece parada, mientras la otra gira.

En lugar de estar parada, la primera rueda podría girar simplemente más despacio que la segunda. En el momento en que las dos ruedas no se muevan a la misma velocidad se producirá un cambio en la dirección.

```

#include <Servo.h>

Servo servoLeft;
Servo servoRight;

int pinServoLeft = 8;
int pinServoRight = 9;

void setup(){
  servoLeft.attach(pinServoLeft);
  servoRight.attach(pinServoRight);
}

```

```
void loop(){
  servoLeft.write(90);
  servoRight.write(180);
  delay(3000);
  servoLeft.write(90);
  servoRight.write(0);
  delay(3000);
}
```

Cambio de dirección sobre el propio eje

Las dos ruedas giran a la misma velocidad en sentidos contrarios.

Aproximadamente debemos girar las ruedas durante 1.000 ms. para hacer un giro de 180 grados. Este tiempo puede variar en función del robot, su calibración, el estado de los motores y el tipo de los motores (marca, modelo, etc.).

```
#include <Servo.h>

Servo servoLeft;
Servo servoRight;

int pinServoLeft = 8;
int pinServoRight = 9;

void setup(){
  servoLeft.attach(pinServoLeft);
  servoRight.attach(pinServoRight);
}

void loop(){
  servoLeft.write(180);
  servoRight.write(180);
  delay(1000);
  servoLeft.write(90);
  servoRight.write(90);
  delay(3000);
  servoLeft.write(0);
  servoRight.write(0);
  delay(1000);
  servoLeft.write(90);
  servoRight.write(90);
  delay(3000);
}
```

5. Detección de línea

Los sensores que utilizamos para detectar la línea son digitales. Retornan un **1** si consiguen leer la línea. De lo contrario, retornarán un **0** al no leer la línea. ¡Cuidado! Según el modelo, el sensor podría funcionar a la inversa.

El siguiente programa hace que el robot avance hasta que detecta una línea en el suelo. Al detectarla gira 180 grados y continúa caminando de nuevo.

```
#include <Servo.h>

int NO_LINE = LOW; // También podría ponerse 0
int LINE = HIGH; // También podría ponerse 1

Servo servoLeft;
Servo servoRight;

int pinIrDer = 3;
int pinIrIzq = 2;
int pinServoRight = 9;
int pinServoLeft = 8;

void setup(){
  Serial.begin(9600); // Descomentar si queréis debuguear por consola
  pinMode(pinIrDer, INPUT);
  pinMode(pinIrIzq, INPUT);

  servoLeft.attach(pinServoLeft);
  servoRight.attach(pinServoRight);
}

void loop(){
  if(digitalRead(pinIrIzq) == NO_LINE || digitalRead(pinIrDer) == NO_LINE){
    servoLeft.write(0);
    servoRight.write(180);
  }else{
    servoLeft.write(180);
    servoRight.write(180);
    delay(1500); // Media vuelta
  }
}
```