

Guía para Proyecto 5: Robot MiniSumo

Club de Robótica - FI - UNLP

1. Objetivos de la guía

Esta guía se plantea a los interesados en encarar el proyecto del robot MiniSumo para facilitarles la tarea de comenzar a programar con Arduino en base al proyecto y para que puedan avanzar más rápidamente en el desarrollo de otros proyectos más sofisticados. Este proyecto aun no se ha materializado, con lo cuál, esta guía solo hace un lineamiento del proyecto para avanzar con la programación, cosa que a la hora de implementarlo sólo quede construir el robot.

2. Planteo del problema

Algunas consideraciones para pensar la programación y fabricación del robot:

1. Buscar en internet guías, papers, videos, trabajos que expliquen el reglamento general de estos robots y sus principales características.
2. Asegurarse de poder programar la placa Arduino UNO desde una PC y conocer las funciones básicas de programación. Comprender cómo se configuran los pines como entradas y salidas, qué es una salida PWM, qué es el ADC del microcontrolador y cómo usarlo, etc.
3. Pensar una o varias maneras de adquirir lecturas sobre la ubicación del contrincante. ¿Qué sensores utilizar?
4. Utilizar sensores reflexivos en la parte inferior para detectar los bordes del tatami y no salirse del mismo.
5. Una rutina de programa debe comandar los motores para cumplir el item anterior.
6. Pensar alguna estrategia de ataque o de defensa. ¿Qué conviene hacer? Hay algunos que dan un salto para evitar ataques.
7. Ahora, los motores que tenga el robot deben reaccionar de tal manera que el autito “ataque” con toda la potencia disponible. Por lo tanto, la alimentación es un detalle importante, al igual que los motores.
8. ¿Qué forma conviene que tenga el robot? Algunos tienen forma *circular* para que el oponente “resbale”.
9. Conviene que utilice ruedas de goma, para no deslizarse.

3. Ejercicios para practicar con Arduino + Autito

Ejercicio 1: Volviendo a programar con Arduino

Conecte el Arduino a una PC con el software ya instalado. Verifique que los drivers se encuentran instalados mirando en *Herramientas, Puerto Serial*. Debería poder visualizar un COM distinto de COM1. Una vez corroborado lo anterior, puede arrancar:

- (a) El shield de potencia del autito posee dos **LEDs** conectados a los pines **10** y **13** del Arduino. Enciéndalos alternadamente con una frecuencia de 1 segundo.
- (b) El shield también posee un pulsador conectado al pin **14**(digital). Deje encendido un LED y al presionar el pulsador del shield, apáguelo y encienda el segundo LED.
- (c) Como ya está cansado de retos sencillos, haga que ambos LEDs parpadeen (con una frecuencia de 1 vez por segundo) un número de veces aleatorio entre 1 y 10. Para iniciar de vuelta el parpadeo, haga que sea necesario presionar el pulsador de la placa; esto es, para que quede más prolijo el código. *Plus:* Puede verificar el número aleatorio obtenido enviándolo por el puerto serie y observándolo en el monitor serial del IDE de Arduino.

- (d) ★ Envíe por el puerto serie una orden para seleccionar uno de los dos LEDs y luego envíe el número de veces que debe parpadear dicho LED (suponiendo un parpadeo de 1 vez por segundo). El código debe reiniciarse al presionar el pulsador para esperar nuevas órdenes

Ayuda: Para el inciso a) puede ayudarle la función `millis()`, con la que puede calcular intervalos de tiempo. Para el inciso b) puede serle útil la función `digitalRead(#pin)`. En c) puede incorporar la función `random(min, max)` Nota: Recuerde que en realidad el número es *pseudoaleatorio*. Por ejemplo, puede inicializar la función con un valor de tensión leído en un pin cualquiera. Para d) utilice las funciones del puerto serie que se listan en **Serial**, en la referencia de Arduino.

Ejercicio 2: Del tatami no se sale...

Los sensores reflexivos se encuentran conectados a los pines analógicos del Arduino (**A[0..5]**). Están polarizados de tal manera, que el pin correspondiente a un sensor indicará un valor pequeño (aprox 0) si el mismo se encuentra ante superficies reflexivas. Por el contrario, ante superficies de baja reflexión, indicará un valor alto (aprox. 1024). En este proyecto usaremos al menos un sensor para detectar una línea blanca, que podría ser el borde de un tatami de MiniSumo.

- (a) Determine a qué pin se encuentra conectado cada sensor. Puede serle útil una vez realizadas las mediciones, enviarlas por el puerto Serie y visualizarlas en el monitor serie de Arduino.
- (b) Los valores medidos por el ADC van de 0 a 1024. Puede que los sensores, ya sea por la superficie, por la iluminación ambiente o por la polarización, nunca midan menos de un valor dado, por ejemplo 24. Lo mismo para un valor máximo, por ejemplo, 980. Sería bueno determinar, para una superficie dada, cuál es el valor máximo y cuál es el valor mínimo medido sobre dicha superficie. Lea distintos valores sobre la pista (sobre blanco y negro) a lo largo de 10 segundos y muestre el resultado (el valor máximo y el mínimo medidos) en pantalla a través del monitor serie.

Nota: No almacene muestras a lo loco, porque se va a quedar sin memoria. Puede realizar el algoritmo de búsqueda de máximo y mínimo casi sin usar memoria.

- (c) A veces conviene trasladar las mediciones en un cierto rango, a otro, para facilidad en cálculos. Por ejemplo, si una persona desea medir temperatura entre 0 y 100 grados, y las mediciones en realidad son entre 400 y 800, por poner un ejemplo, conviene que *mapee* el rango [400 800] a [0 100]. Utilice la función `map()` para trasladar todo el rango de las lecturas de los sensores (desde el mínimo hasta el máximo detectado) al rango [0 100].

Nota: Si en algún momento llegan mediciones fuera del rango que usted midió con anterioridad, al hacerle la función `map()` irán a un valor quizás no deseado. Por lo tanto, debe imponer restricciones en los valores. Esto también se denomina *saturar*. La función `constrain()` puede ayudarle.

- (d) Ahora que pudo medir el blanco de la línea y ubicó la medición en un rango querido, recordemos un poco de los motores del autito:
- (I) Los motores utilizados para la tracción de las ruedas funcionan con tensión continua, y para invertir la dirección de giro, basta con invertir la polaridad de la tensión aplicada. Recordar que el motor izquierdo está controlado por los pines **8** y **5** del Arduino, mientras que el derecho se encuentra controlado por los pines **13** y **7**. Haga un programa que mueva hacia adelante y hacia atrás al autito.
- Nota:** Recuerde que para variar la tensión aplicada se utilizaba PWM mediante la función `analogWrite(#pin,D)`.
- (II) ¿Cómo deberían reaccionar los motores para que el robot no se escape del tatami?
- (III) ¿Conviene que se mueva recto hacia el centro de la pista? Quizás al invertir el movimiento conviene que altere su dirección, por si su oponente viene atrás para atacarlo.
- (IV) Si el oponente lo está empujando fuera de la pista y usted detecta el borde, quizás tenga que utilizar toda la potencia disponible para volver al tatami. Es decir, debería poder detectar también que su oponente lo está *tocando*. ¿De qué manera puede hacerlo? ¿Con qué sensores?

Ejercicio 3: Midiendo al enemigo

Como ya vimos, el robot debe detectar la presencia de su contrincante para intentar atacarlo o esquivarlo. Hay muchas maneras de realizar la medición utilizando distintos sensores. A continuación se listan dos métodos.

1 - Usando el sensor ultrasónico HC-SR04

Este sensor es un sensor de ultrasonido. El diagrama de tiempos de la Figura 1 indica que es necesario enviar un pulso de 10 μseg mediante el Arduino antes de comenzar el sensado. Luego el módulo HC-SR04 enviara 8 pulsos de una señal de 40 kHz esperando recibir el eco. Midiendo el tiempo que tarda en llegar el eco Δt en microsegundos, puede determinarse la distancia a la que se encuentra el objeto dónde rebotó la señal de la siguiente manera:

$$d = \frac{\Delta t}{2} \times 0,03435 \text{ cm}/\mu\text{s}$$

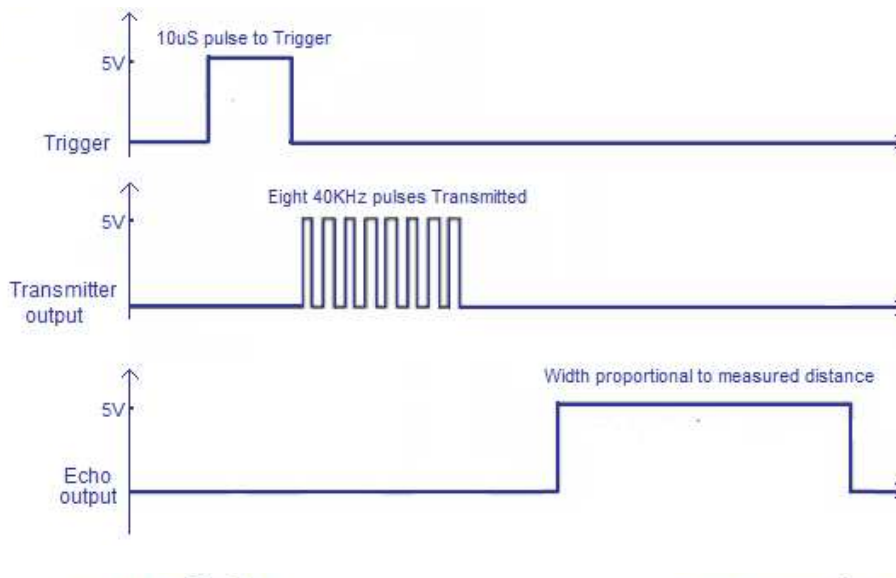


Figura 1: Diagrama temporal de señales útiles para el sensor HC-SR04.

- Busque en Internet información sobre el sensor, como cuántos pines tiene, cómo se usan y cómo configurar los respectivos pines en el Arduino.
- Realice un programa en Arduino para medir la distancia que hay entre el sensor y un objeto. Puede enviar el dato a la PC a través del puerto serie y visualizarlo en el monitor serie de Arduino.
- Este sensor tiene un cierto rango donde funciona correctamente. Determinélo, realizando mediciones.
- El precio de este sensor ronda los 4 USD. ¿Le parece que es el sensor más conveniente para esta aplicación?

Nota: Para utilizar de manera sencilla el sensor, puede resultarle piola la librería **Ultrasonic.h** que se puede encontrar en la Web.

2 - Usando emisor y receptor infrarrojo

Puede utilizarse un esquema similar al de los sensores CNY70 para detectar proximidad. Es decir, midiendo reflexión de un rayo irradiado por un emisor infrarrojo con un fototransistor. La ventaja de este tipo de sensores es que pueden conseguirse en cualquier local de electrónica, no así con el HC-SR04. Además, tanto el emisor IR como el fototransistor son económicos.

- Busque en Internet información y esquemas que utilicen estos componentes al utilizarlos para medir proximidad.
- Para regular a qué distancia el receptor detecta al emisor, se debe variar crecientemente la corriente con que se alimenta al emisor. Piense cómo puede armar un ensayo para determinar la relación entre corriente del emisor vs distancia de detección.
- El problema que aparece, es que el fototransistor que utilice para recibir el haz reflejado también recibe luz ambiente. Entonces puede que detecte reflexión (en realidad es luz ambiente) sin que el haz infrarrojo haya rebotado en ningún objeto. Lo que se puede hacer para solucionar este aspecto, es alimentar el diodo infrarrojo con una corriente pulsante, de una dada frecuencia. Luego, medir con un pin del Arduino la frecuencia de la señal medida en el fototransistor. Si esta se corresponde con la enviada en el emisor, entonces el haz infrarrojo está rebotando en un objeto.

- (I) Busque en la Web ejemplos de aplicación de sensores IR estimulados de dicha manera.
- (II) El método de enviar información “montada” en un valor de frecuencia se denomina *modulación*. Investigue sobre *demodulación*, *frecuencia portadora* y *señal moduladora*. Busque proyectos con microcontroladores donde se utilicen estos términos.
- (III) Investigue sobre el integrado *TSOP1138*. Otro similar es el *IRM2638*. Ambos son demoduladores de frecuencias cercanas a los 38 kHz . Los controles remotos convencionales los utilizan.