

Taller 2: Comportamientos Reactivos

?

Viernes 23/09

A continuación se presentan los ejercicios a resolver. Se recomienda fuertemente que ante cualquier duda repasen los slides de la clase.

1 Introducción

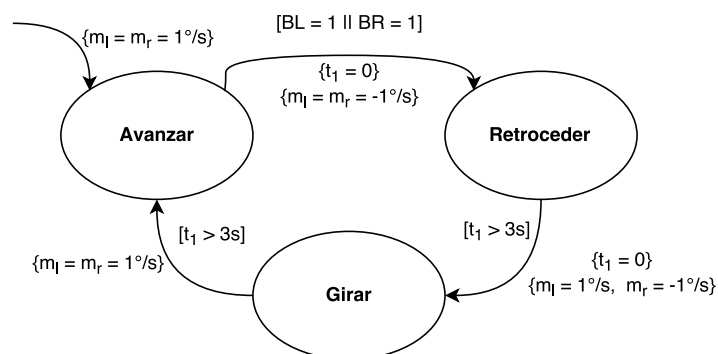
Para resolver este TP se utilizará el simulador `123circuit`. Una vez que se haya resuelto correctamente un comportamiento en el simulador, se podrá evaluar el mismo ejecutando en el robot ExaDuino.

Para resolver los ejercicios que se presentan a continuación, se utilizará un circuito existente y que contiene un código esqueleto. Para trabajar sobre el mismo, acceder al link presentado en cada ejercicio y duplicar el circuito para poder editarlo.

Ejercicio 1 (resuelto)

Nota: a efectos de este Taller, el presente ejercicio se encuentra resuelto y servirá a modo de guía para la resolución de los siguientes ejercicios. Se recomienda observar la resolución (código cargado en la simulación) en el circuito correspondiente: <https://circuits.io/circuits/2779457>.

El primer comportamiento reactivo que queremos programar en un robot se trata de la evasión de obstáculos. Para eso, en primera instancia vamos a pensar que contamos con un robot que posee solamente dos sensores de contacto (*bumpers*). De esta forma sólo nos podemos enterar de la presencia de un objeto cuando el robot choca contra éste. Utilizando el circuito base (esqueleto) mencionado anteriormente se pide programar el comportamiento que se muestra en la maquina de estados a continuación:

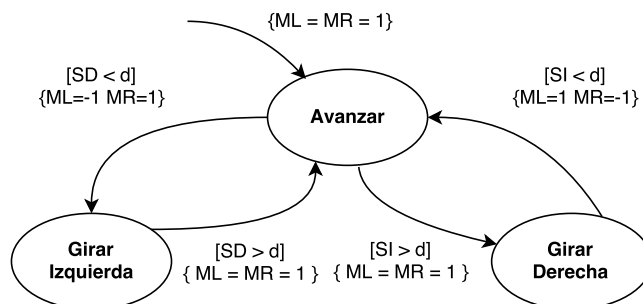


Nota: para controlar los comportamiento que dependen del paso del tiempo, utilice la función `millis()` (sugerencia, utilice la variable `t` presente en el esqueleto).

Ejercicio 2

Para mejorar el comportamiento de evasión de obstáculos y evitar que el robot tenga que chocar con los objetos para detectarlos, en una segunda instancia queremos usar los sensores infrarrojos existentes (simulados en el circuito como sensores de luz). Para eso se quiere lograr que cuando el robot vea un objeto con el sensor de la izquierda el robot gire hacia la derecha, y por el contrario, si detecta algo con el sensor de la derecha queremos que gire hacia la izquierda.

En otras palabras, el comportamiento que se busca se puede definir mediante la máquina de estados a continuación:

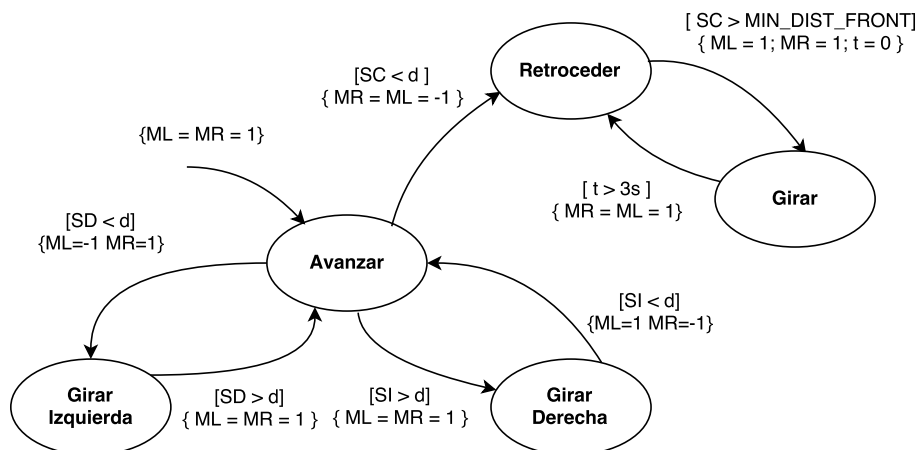


Link: <https://circuits.io/circuits/2779469>.

Ejercicio 3

El comportamiento que definimos en el ejercicio anterior tiene el problema que cuando el robot detecta un obstáculo sólo con el sensor del medio no puede evitarlo. Por lo tanto quisieramos hacer que cuando el robot además ve un obstáculo con el sensor central, retroceda como en el ejercicio 1 pero hasta una distancia de MIN_DIST_FRONT, luego gire 3s y continúe avanzando.

La máquina de estados correspondiente es:



Nota: utilice el circuito del ejercicio anterior como base para el ejercicio actual.