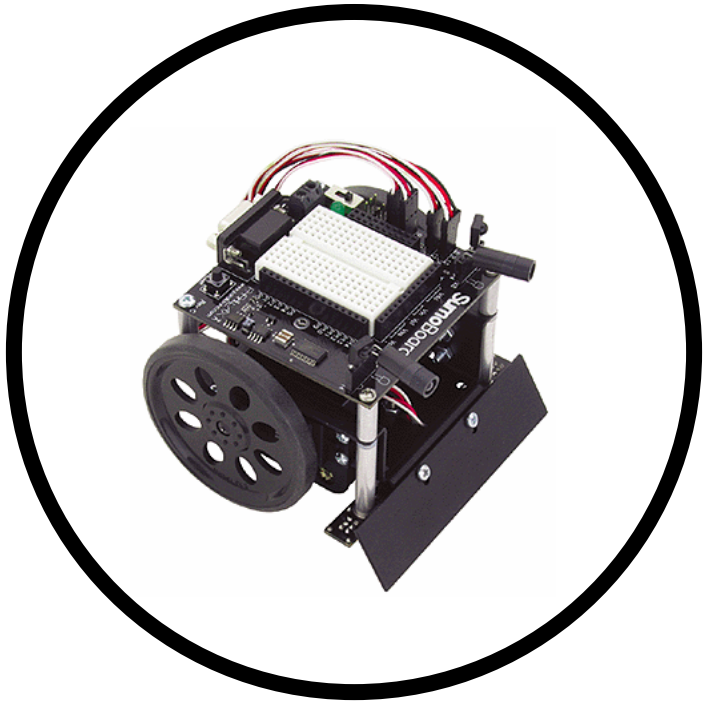


Estrategias de solución para la prueba del Minisumo, Madrid-Bot 2009

Salustiano Nieva
Juan Antonio Breña Moral

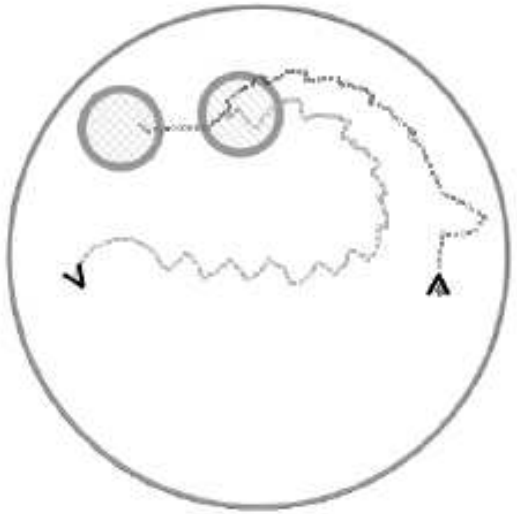
CONSEJERÍA DE EDUCACIÓN
Comunidad de Madrid

EDUCAMADRID

MADRIDBOT

Índice

1. Introducción
2. Arquitectura del robot
3. Madridbot 2009
4. Maquinas de estado finito
5. Arquitectura Subsumption
6. Una estrategia de solución
7. Pseudo código
8. Técnicas avanzadas
9. Recursos



“Divide et vinces”
Julio Cesar

Introducción

El objetivo de esta prueba es que dos robots que compiten dentro de un Tatami o área de combate, luchen por conseguir la mayor cantidad de puntos efectivos o Yuhkoh. El combate lo gana el Robot que consigue más puntos. Los robots tienen cumplir la condición de no pesar más de 500 g ni exceder de las dimensiones de 10 cm de ancho por 10cm de largo.

Mas información sobre la prueba:
http://www.madridbot.org/Documentos/Documentos%202008/Normativa%20Minisumo%202008.pdf

Arquitectura del robot

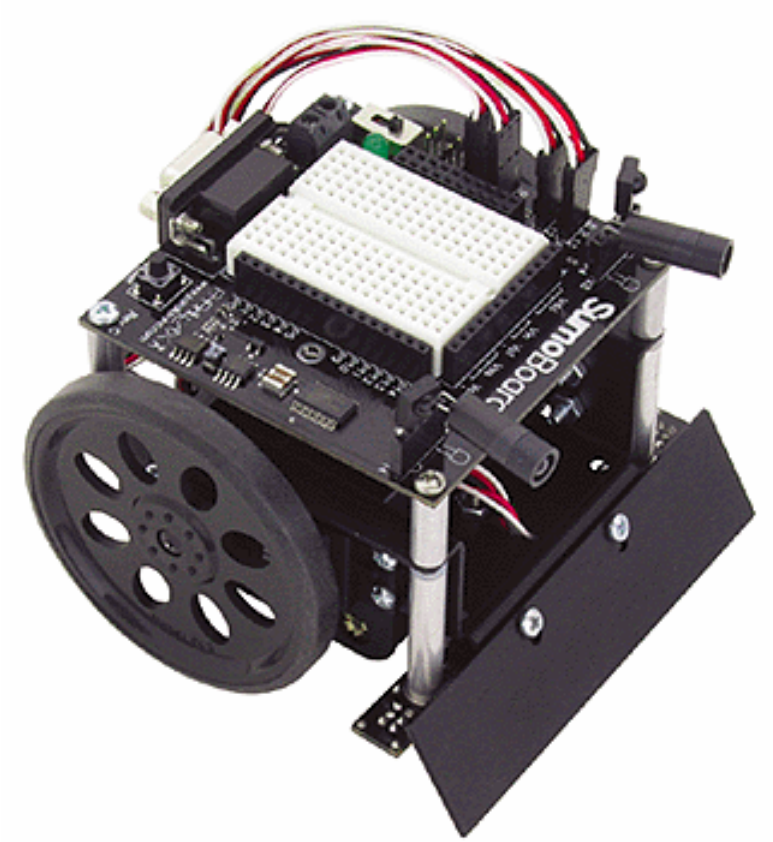
El robot a desarrollar que permitirá superar la prueba del Laberinto constara de la siguiente arquitectura:

Sistema locomotor

El sistema locomotor del robot constara de 4 motores, los cuales emplean protocolo **PWM** para su control.

Sistema sensitivo

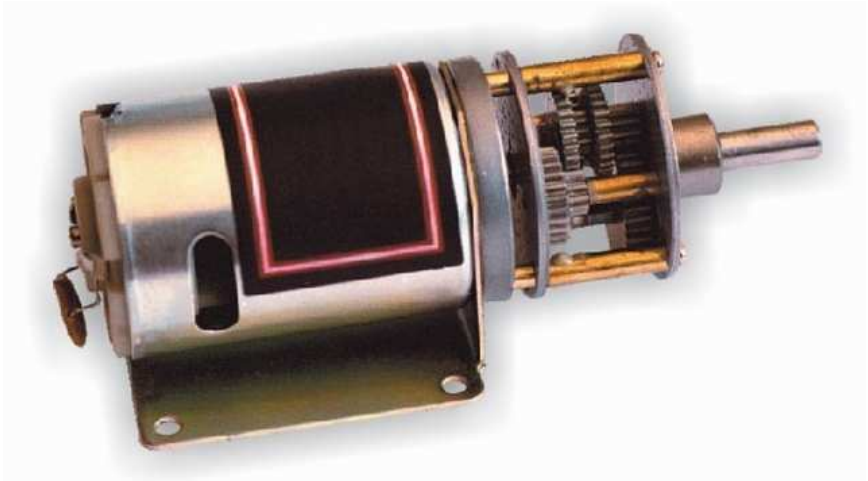
El sistema sensitivo del robot esta compuesto por 4 sensores que miden el color y 2 sensores de medición de distancia

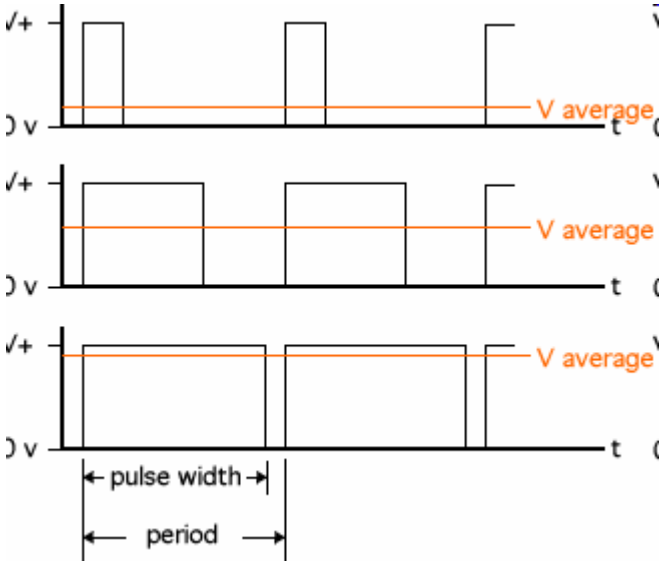


Arquitectura del robot

Sistema locomotor

El sistema locomotor estará compuesto por 4 motores de gran potencia. En esta prueba es fundamental que los motores permitan mover la propia estructura y la del oponente.





Es conveniente que las ruedas tengan gran adherencia.

Arquitectura del robot

Sistema locomotor: Objetivos

Los objetivos del sistema locomotor son los siguientes:

1. Desarrollo de API para realizar las siguientes operaciones:
    1. Marcha hacia delante
    2. Marcha hacia atrás
    3. Giro a la izquierda de 90º
    4. Giro a la izquierda de X Grados
    5. Giro a la derecha de 90º
    6. Giro a la derecha de X Grados
    7. Stop
    8. Set/Get de velocidad

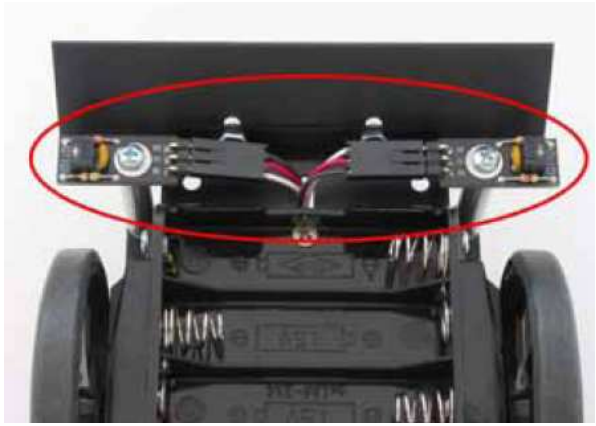
Arquitectura del robot

Sistema sensitivo

El sistema sensitivo del robot esta compuesto por 4 sensores de color y 2 sensores de medición de distancias.

Los primeros sensores servirán para controlar la posición del robot y no permitir que se salga del recinto de competición.
Los segundos sensores permiten medir la distancia.

Sensor infrarrojo



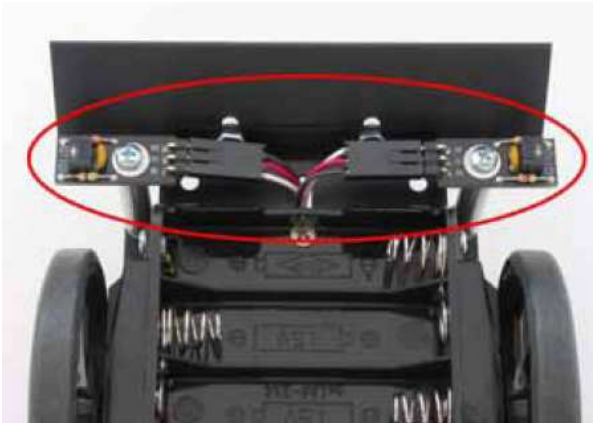
Sensor de detección de limites

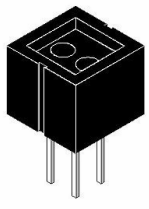
Arquitectura del robot

Sistema sensitivo : Sensores de detección de color

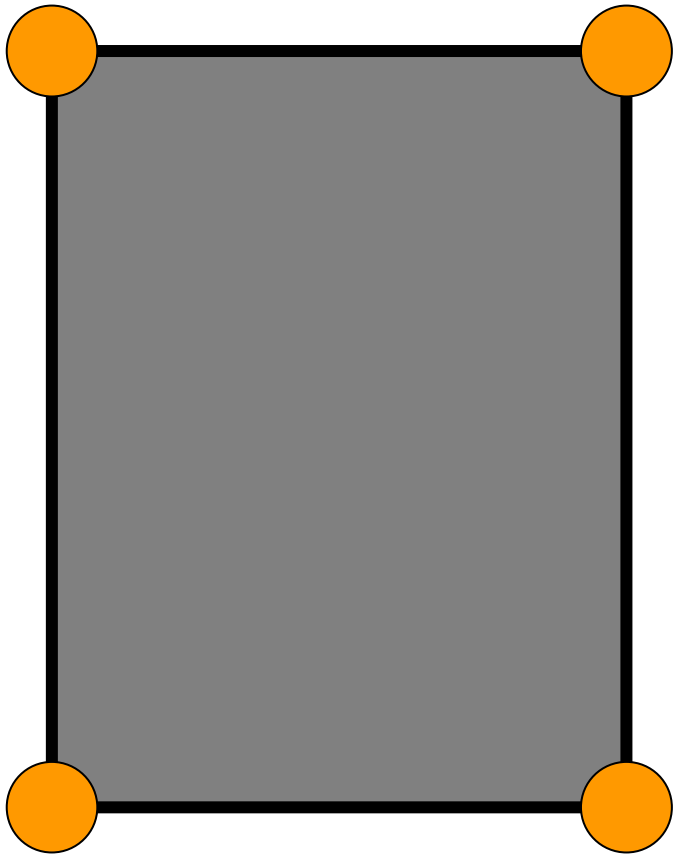
Los sensores de detección de color, detectaran que el robot esta situado dentro del tatami.

<u>La primera prioridad es permanecer en el tatami</u>





Sensor CNY70

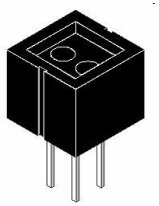


Arquitectura del robot

Sensor CNY70

El CNY70 es un sensor de infrarrojos de corto alcance basado en un emisor de luz y un receptor, ambos apuntando en la misma dirección, y cuyo funcionamiento se basa en la capacidad de reflexión del objeto, y la detección del rayo reflectado por el receptor.

Requerimiento del sensor:
El sensor es la necesidad de tener que situarlo muy próximo al objeto para detectar correctamente la reflexión.

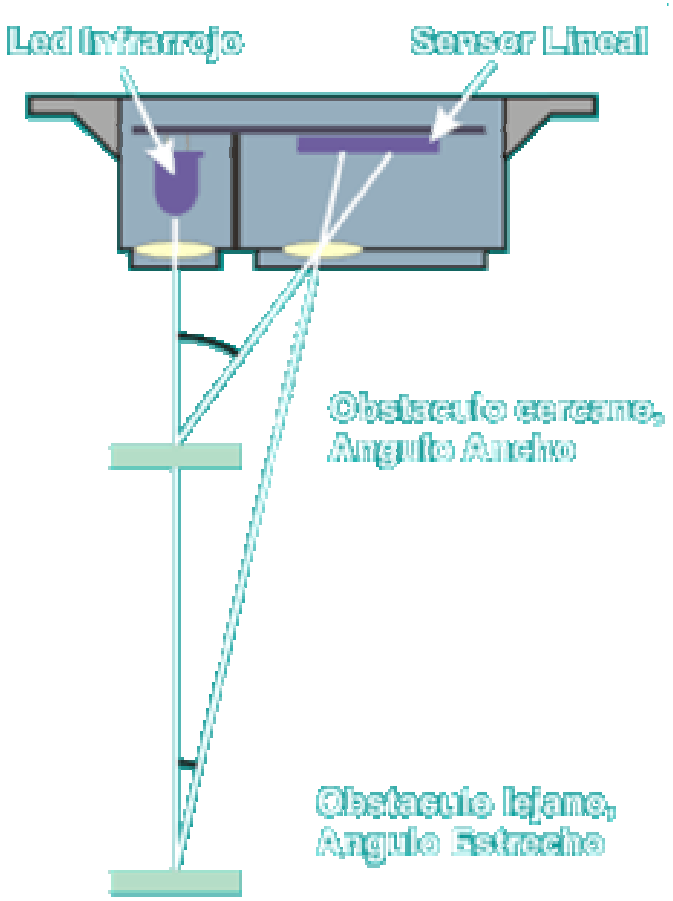


Sensor
CNY70

Arquitectura del robot

Sistema sensitivo : Sensores de distancia

La familia de sensores GP2DXX de Sharp son dispositivos de reflexión por infrarrojos con medidor de distancia proporcional al ángulo de recepción del haz de luz que incide en un sensor lineal integrado.



Arquitectura del robot

Sensores de distancia de la familia GP2DXX

La familia de sensores GP2DXX de Sharp son dispositivos de reflexión por infrarrojos con medidor de distancia. Puede haber de dos tipos: Analógicos y Boléanos. En este caso usaremos uno analógico

Arquitectura del robot

Sensores de distancia de la familia GP2DXX

La familia de sensores es el siguiente:

Modelo	Características	**Rangos absolutos Máximos**		Característica Electro-Opticas				
		VCC (V)	Topr (°C)	Rango de distancia de medida	V_{OH} (V) MIN VCC	V_{OL} (V) MAX	Operativo (mA) MAX	Srandby (µA) MAX
GP2D02	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , 8 bits.	-0.3 a +10	-10 a +60	100 a 800	V_{CC}-0.3	0.3	17	8
GP2D021	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , 8 bits.	-0.3 a +10	-10 a +60	40 a 300	V_{CC}–0.3	0.3	35	8
GP2D05	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , 1 bits.	-0.3 a +10	-10 a +60	100 a 800	V_{CC}–0.3	0.3	22	8
GPD12	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , valor analógico entre 0-3V dependiendo de la distancia	-0.3 a 7	-10 a +60	100 a 800	V_0(TYP) = 0.4V a 80 cm		MAX.50	
GP2D120	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , valor analógico entre 0-3V dependiendo de la distancia	-0.3 a 7	-10 a +60	40 a 300	V_0(TYP) = 0.4V a 30 cm		MAX.50	

Arquitectura del robot

Sensores de distancia de la familia GP2DXX

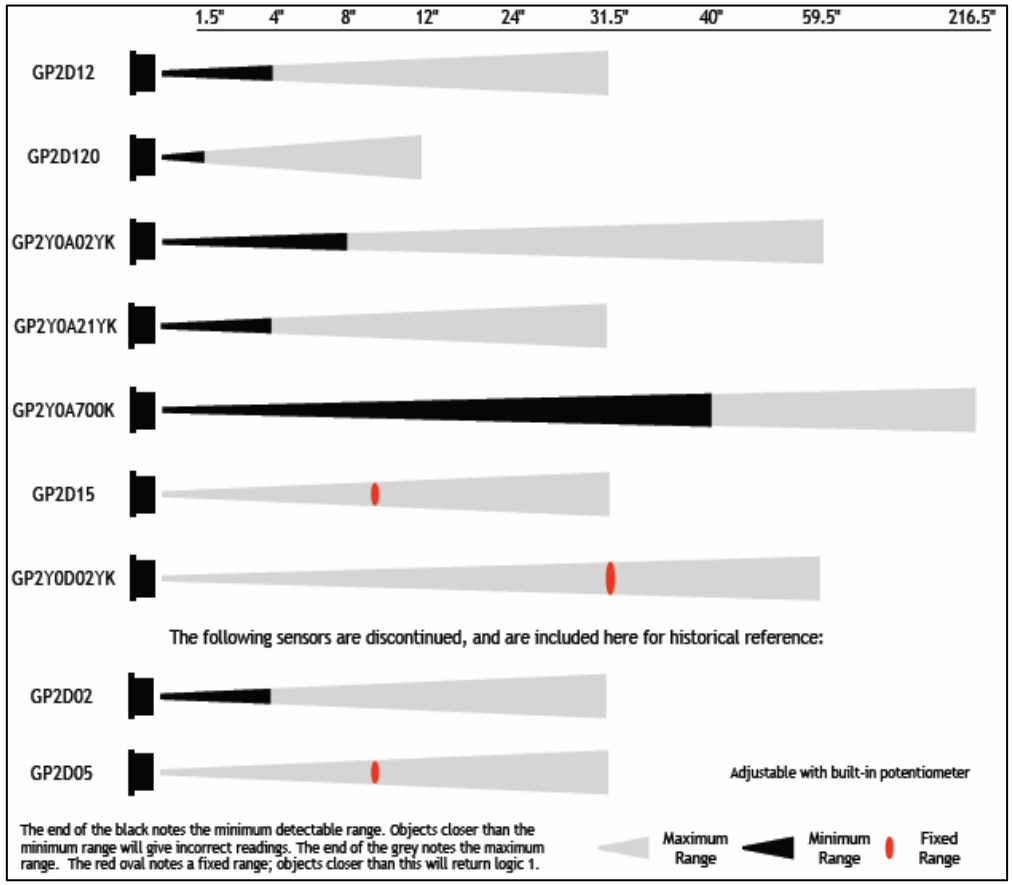
La familia de sensores es el siguiente:

Modelo	Características	**Rangos absolutos Máximos**		Característica Electro-Opticas				
		VCC (V)	Topr (°C)	Rango de distancia de medida	V_{OH} (V) MIN VCC	V_{OL} (V) MAX	Operativo (mA) MAX	Srandby (µA) MAX
GP2D150A	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , valor digital (0 o 1)	-0.3 a 7	-10 a +60	30 a 300 (Detección distancia typ. 15cm)	V_{CC}-0.3	0.6	MAX.50	
GP2D150T	Sensor que mide distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , valor digital (0 o 1)	-0.3 a 7	-10 a +60	30 a 300 (Detección distancia typ. 22cm)	V_{CC}-0.3	0.6	MAX.50	
GP2Y0D02YK	Sensor que mide largas distancias con PSD (Detector Sensible a la Posición), LED infrarrojo y circuito de procesado de señal , valor digital (a 80 cm)	-0.3 a 7	-10 a +60	200 a 1500	V_{CC}-0.3	0.6	MAX.50	

Arquitectura del robot

Sensores de distancia de la familia GP2DXX

La familia de sensores es el siguiente:

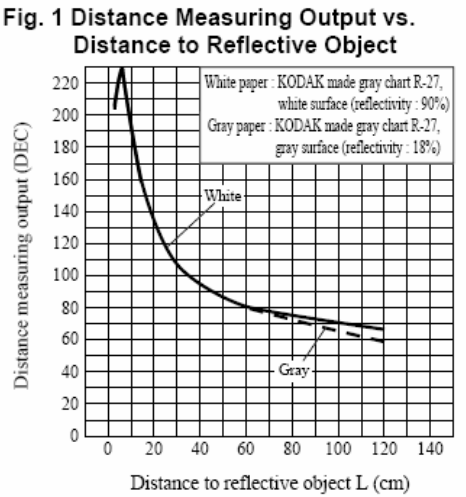


Arquitectura del robot

Sensores de distancia de la familia GP2DXX

El sensor Sharp GP2D02, es un medidor de distancias por infrarrojos que indica mediante una salida digital de 8 bits la distancia a la que se encuentra el objeto sobre el que se refleja el haz luminoso.

Su rango de utilización es desde los 10 a los 80 cm.



Arquitectura del robot

Sistema sensitivo : Objetivos

Los objetivos del sistema sensitivo son los siguientes:

1. Desarrollo de API para realizar las siguientes operaciones:
    1. Establecer rangos de confianza y valores comparativos
    2. Medir distancias en los 2 sensores de distancia
    3. Detectar si en alguna de las partes del robot esta en riesgo

Arquitectura del robot

Plataforma de control

La plataforma de control del robot se basara en 2 elementos: Placa controladora con microcontrolador (PIC16F876) y una plataforma de desarrollo

Placa controladora

La placa electrónica será desarrollada por el IES Antonio Machado

Plataforma de desarrollo

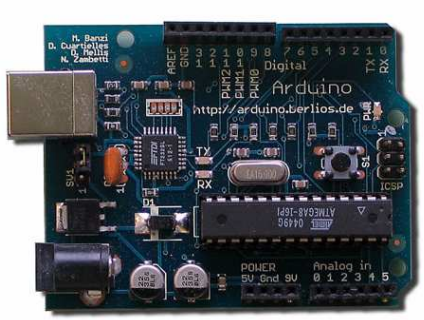
Para la modelización del robot y su posterior codificación en código para el microcontrolador PIC16F876, se empleara **Proteus**.

Arquitectura del robot

Alternativas

La construcción del desarrollo también podría ser posible a través de las siguientes opciones:

1. Lego Mindstorms NXT
2. Arduino
3. Sun spot
4. VEX



Madrid-Bot

Madrid-Bot es un concurso de micro robótica organizado por los Centro que imparten las enseñanzas del Ciclo Formativo de Grado Superior de Desarrollo de Productos Electrónicos en la Comunidad Autónoma de Madrid.

Madrid-Bot intenta ser un concurso de micro-robótica y más aun un lugar de encuentro donde los alumnos matriculados en Centros de Educación Secundaria y especialmente los alumnos y alumnas de Bachillerato, Ciclos Formativos de la familia profesional de electrónica, informática, etc.. puedan no sólo competir con sus prototipos sino también compartir sus conocimientos.

Madridbot

Pruebas del concurso

Las pruebas que se realizan en Madridbot, son las siguientes:

1. Rastreadores
2. Velocistas
3. Laberinto
4. Mini sumo
5. Prueba Libre

Maquinas de estados finitos

Las Máquinas de Estados Finitos (FSM), también conocidas como Autómatas de Estados Finitos (FSA), explicado de forma simple, son modelos de comportamiento de un sistema o un objeto complejo, con un número limitado de modos o condiciones predefinidos, donde existen transiciones de modo. Las FSMs están compuestas por 4 elementos principales:

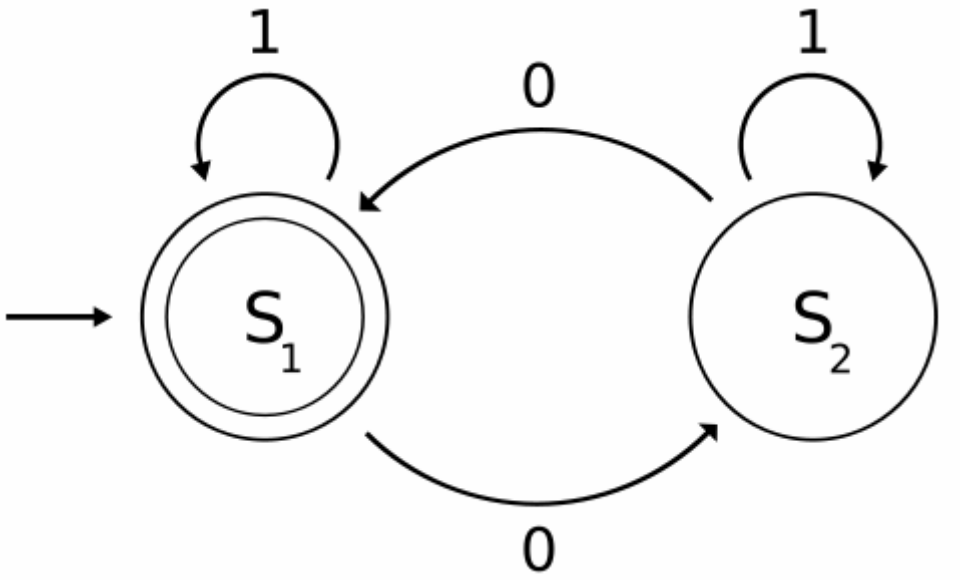
1. Estados que definen el comportamiento
2. Transiciones de estado
3. Reglas o condiciones que deben cumplirse para permitir un cambio de estado
4. Eventos de entrada que son externos o generados internamente

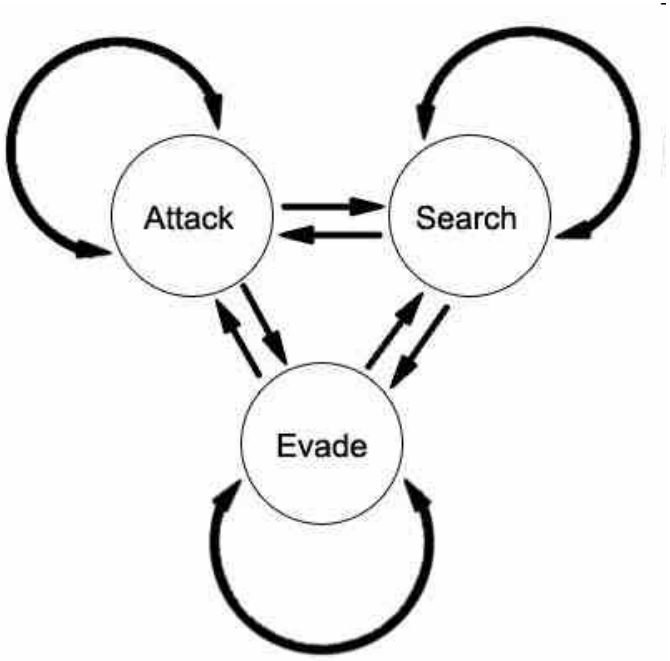
Maquinas de estados finitos

Una máquina de estados finitos debe tener un estado inicial que actúa de punto de comienzo, y un estado actual que recuerda el producto de la anterior transición de estado. Los eventos recibidos como entrada actúan como disparadores, que causan una evaluación de las reglas que gobiernan las transiciones del estado actual a otro estado. La mejor manera de visualizar una FSM es pensar en ella como un diagrama de flujo o un grafo dirigido de estado, aunque como se verá existen técnicas de abstracción más precisas que pueden ser usadas.

Maquinas de estados finitos

La resolución de ciertos problemas, puede ser modelizado a través de teoría de maquinas de estado finitos. En el caso de la prueba del laberinto, este enfoque puede ser aceptado.



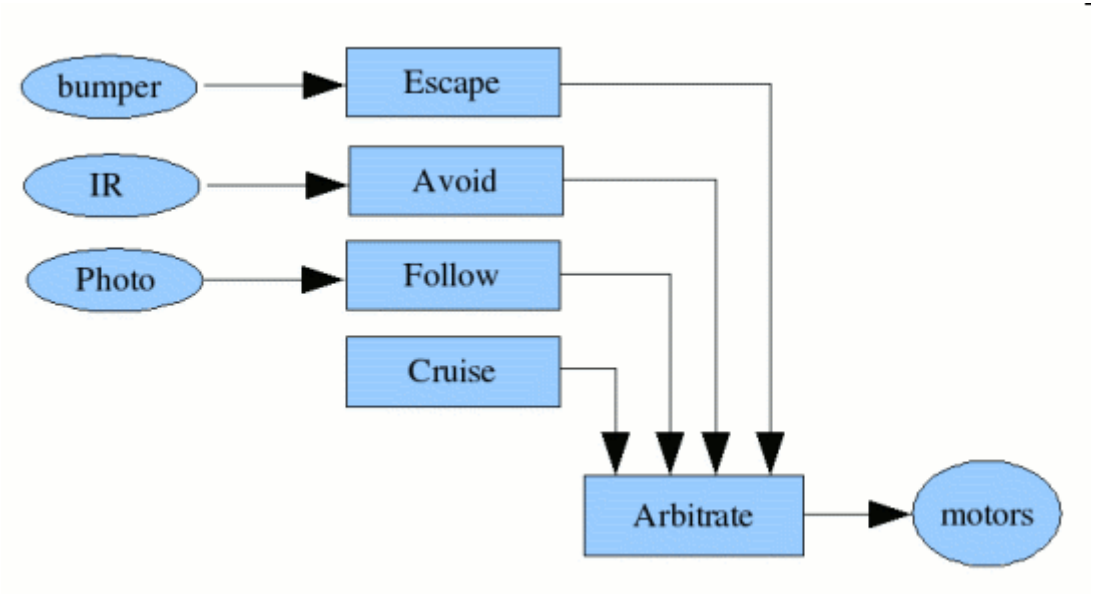


Arquitectura Subsumption

La arquitectura subsumption es una arquitectura de tipo reactiva, desarrollada por Rodney brooks y se basa en la construcción de capas de comportamiento.

Este robot se podría modelizar a través de 4 comportamientos:

1. Atacar
2. Esquivar
3. Buscar
4. Sobrevivir



Una estrategia de solución

La solución se basaría en un sistema que resolviera las siguientes necesidades:

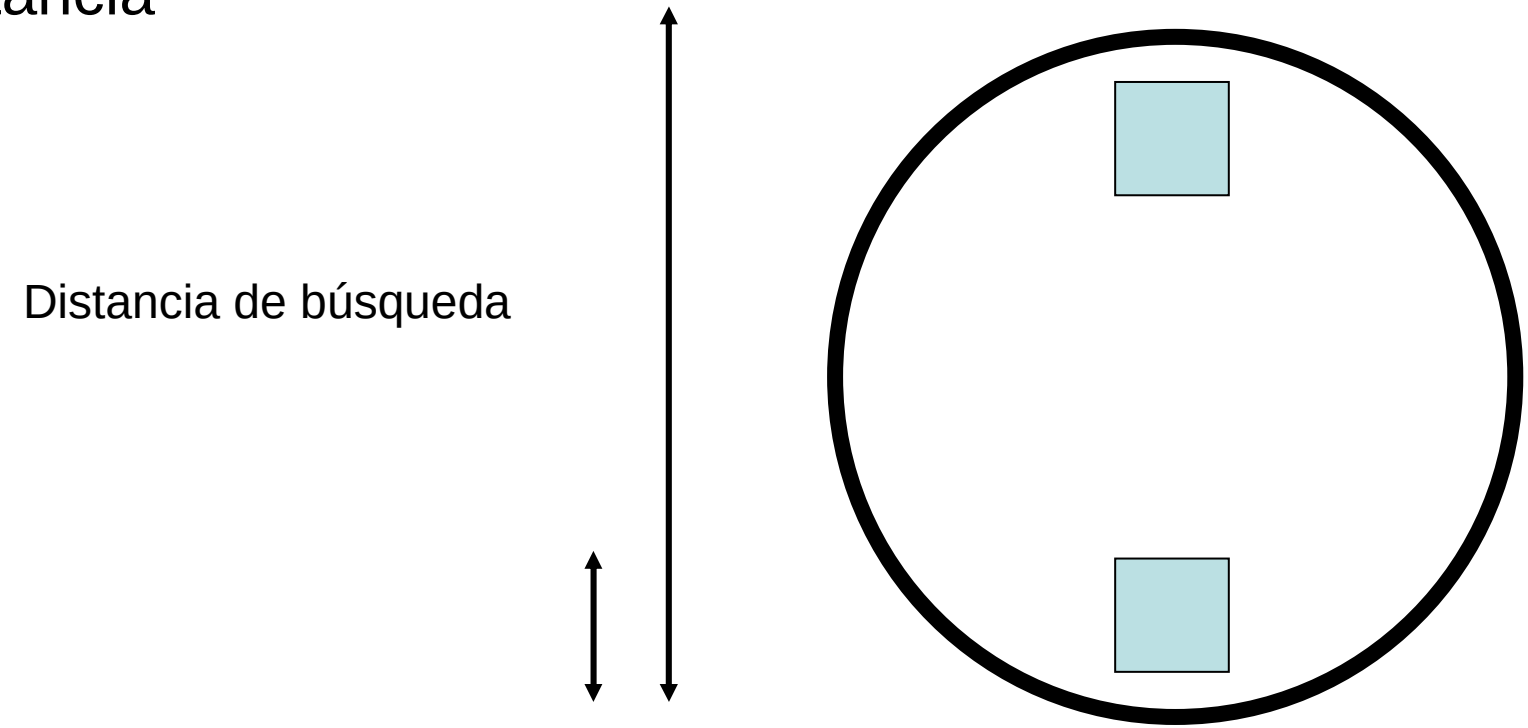
1. ¿Estoy en el tablero?
2. ¿Dónde esta mi oponente?
3. Sacar al contrario del tatami

Según las instrucciones de la prueba, el robot debe esperar 5 segundos desde que se inicia el programa antes de iniciar sus rutinas.

El tablero donde opera tiene 75 cm. De diámetro.

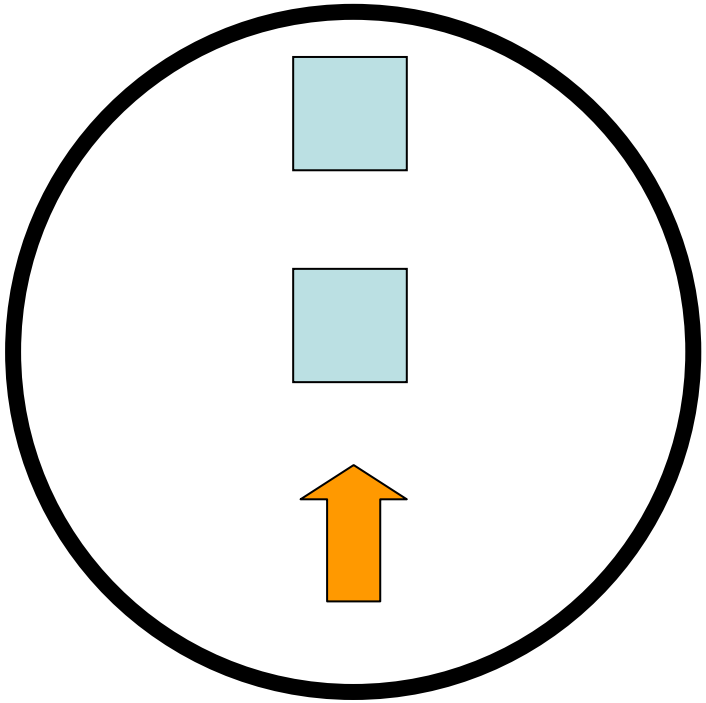
Una estrategia de solución

El robot para buscar al oponente usaría el sensor de medición de distancia y detectaría objetos en la siguiente distancia



Una estrategia de solución

Al detectar un objeto en el campo de visión, aceleraría y avanzaría para ocupar el espacio central del tatami e intentar desplazar a su oponente
fuera del tatami.



Pseudo código

Se propondrán una serie de versiones del sistema de control del robot de Sumo las cuales irán incrementando la dificultad del diseño.

1. sumoBotV1
2. sumoBotV2

Pseudo código

Seudocodigo sumoBotV1:

```
sumoBotV1(){
      esperar5Segundos();
      mientras(true){
            goToEdge();
            backwardSlightly();
            spinRight();
      }
}
```

Pseudo código

Seudocodigo sumoBotV1:

```
goToEdge(){
     mientras(!detectEdge()){
          forward(-1)
     }
}

backwardSlightly(){
     backward(200);
}
```

Pseudo código

Seudocodigo sumoBotV1:

```
spinRight(){

}
```

Pseudo código

El anterior sistema puede mejorarse de las siguientes maneras:

1. Añadir aleatoriedad a los giros
2. Patrones de avance alternativos

Pseudo código

El seudo código que se presenta a continuación se basara en gestionar los comportamientos del robot.

1. ¿Estoy en el tablero?
2. ¿Dónde esta mi oponente?
3. Sacar al contrario del tatami

Pseudo código

Seudocodigo Programa Principal:

```
sumoBotV2(){
	esperar5Segundos();
	mientras(true){
		statusRobot = getPositionStatus();
		if(statusRobot){
			foundOponent = searchOpponent();
			if(foundOponent){
				attackOponent();
			}
		}else{
			surviveOnTatami()
		}
	}
}
```

Pseudo código

Seudocodigo Programa Principal:

```
getPositionStatus(){
      boolean status = true;
      readCNYSensors();
      if(leftFrontal == 0){
            status = false;
      }else if(rightFrontal == 0){
            status = false;
      }else if(leftBack == 0){
            status = false;
      }else if(rightBack == 0){
            status = false;
      }
      return status;
}
```

Pseudo código

Seudocodigo Programa Principal:

```
surviveOnTatami(){
      readCNYSensors();
      knowWhereIsTheDanger();
      survive();
}
```

Pseudo código

Seudocodigo Programa Principal:

```
searchOpponent(){
      boolean status = false;
      status = findOpponent(135);
      if(status){
            status = findOpponent(90);
            if(status){
                  status = findOpponent(45);
            }
      }
      return status;
}
```

Pseudo código

Para mejorar este código se podría mejorar con las siguientes mejoras:

1. Mejorar la precisión de la detección con un uso combinado de los sensores
2. Aleatoriedad de movimientos de ataque
3. Mejora de la “Inteligencia” en la supervivencia

Modelos avanzados

Los modelos avanzados que gestionan el robot Sumobot, emplean las siguientes técnicas:

1. Localización a través del Algoritmo de Monte Carlo y otras técnicas probabilísticas
2. Algoritmos genéticos / evolutivos

Técnicas avanzadas

Algoritmo de Monte Carlo

El Algoritmo de Monte Carlo, permite estimar la localización de un robot en un ambiente conocido, dado sus movimientos y mediciones de los sensores en el tiempo.

Permite solucionar el problema el cual un robot no conoce su posición inicial pero necesita saber en dónde se encuentra. Se basa en la colección de muestras (las cuales son conocidas también como partículas).

Cada muestra consiste de una posible posición que el robot puede ocupar en el tiempo actual, junto con un valor que representa la probabilidad de que el robot esté en esa posición.

Técnicas avanzadas

Algoritmos genéticos

Los Algoritmos genéticos son llamados así porque se inspiran en la evolución biológica y su base genético-molecular.

Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

Recursos

Documentación adicional

Eventos en España:
http://www.madridbot.org/index.htm
http://complubot.educa.madrid.org/inicio.php?seccion=principal
http://www.eis.uva.es/amuva/robolid/

Sumobot:
*www.cs.bgu.ac.il/~sipper/courses/atea061/**Sumobot**.ppt*
http://www.robotika.sk/elosys/2007/SumoCode.html