

# Zxbot - TEKTRONIX

## Prueba Libre

Antonio José Villena Godoy

Andrew Mark Campbell Boross

Rodrigo Maestro Páramo

Ignacio Yélamos Dumont

Estudiantes de Ingeniería Superior en Telecomunicaciones de la Universidad de Málaga

### Resumen

ZXbot es un proyecto de inteligencia artificial que trata de conseguir el máximo de eficiencia usando el mínimo hardware posible. Es por ello que hemos decidido utilizar un robot comercial de gama baja, el Tritt, para demostrar que la inteligencia artificial no se consigue con grandes arquitecturas hardware ni numerosos sensores.

Ha sido adiestrado para competir en dos pruebas donde la inteligencia es el aspecto fundamental: la prueba libre y el laberinto. En la prueba libre mostramos como un microbot es capaz de jugar contra un ser humano a un juego de habilidad mental, el juego de los palillos, que será explicado posteriormente.

Por otro lado participará en la prueba del laberinto, donde con la ayuda de un solo bumper central deberá ser capaz de reconocer en qué zona del laberinto se encuentra y elegir la ruta más óptima que le lleve a la salida más cercana.

### 1. Introducción

El juego que hemos desarrollado es idéntico al que se popularizó en el programa de Rafaela Carrá hace algunos años. Si no sabes de que juego estamos hablando aquí están las reglas del mismo:

Existen 3 filas de palillos colocadas de la siguiente forma sobre un tablero:

- 7 palillos en la fila superior.
- 5 palillos en la fila central.
- 3 palillos en la fila inferior.

Es decir:

```
IIIIIII  
  IIII  
   III
```

El objetivo del juego es el de dejar al oponente con un solo palillo. Para ello se deberán quitar, por turnos, tantos palillos como se deseen de una sola fila.

El juego lo empiezas tú, ya que si empezara el microbot ganaría siempre y no merecería la pena jugar.

Una vez hayas realizado la jugada, se pulsa el botón del microbot. A continuación este leerá los palillos que quedan en el tablero y realizará su jugada sacando del tablero los palillos que necesite.

Una vez finalizada su jugada, nos avisará de que nos toca tirar con una melodía. Esta se irá tocando cada vez más rápido para meternos prisa. Al final el que gana es el que le deja un palillo a su oponente, que en el caso de ser el robot, sonará una melodía distinta.

Aparte se ha diseñado el programa para que detecte las jugadas ilegales, como no coger palillos o poner palillos en lugar de quitar; en este caso sonará la melodía de perder y el robot empezará a dar vueltas como loco.

El diseño físico del juego se realiza mediante un circuito rectangular con cinta aislante, que será la guía del recorrido del robot y para ello tendrá el sensor central.

El sensor izquierdo se utiliza para saber donde están los palillos, los cuales serán contados o movidos según convenga.

Los palillos han sido implementados con cilindros (manguitos PVC de 40) y en su base llevan adherida una cinta para poder ser detectado por el robot.

El aspecto externo del microbot es el siguiente:



## 2. Plataforma mecánica usada

Para poder implementar el juego de los palillos, además del tablero de juego y palillos de la figura, han sido necesarios otros cambios en el microbot.

Hemos introducido los siguientes elementos nuevos:

- Un altavoz, que conectado al puerto B nos permitirá oír una canción cada vez que el robot termine una jugada.
- Un "parachoques" (abrazadera cortada y alargada), sujeta con dos escuadras para los muebles dobladas.
- Un par de "cuernos" (rosca métrica del N°4 y sujeta por 2 tuercas) que nos ayudarán a sacar los palillos en la dirección que queramos.

También hemos variado la disposición de los sensores: hemos colocado uno en un extremo del parachoques y otro en el centro. Esto nos permitirá ir contando los palillos a la vez que seguimos un camino marcado con cinta aislante negra.

## 3. Arquitectura hardware

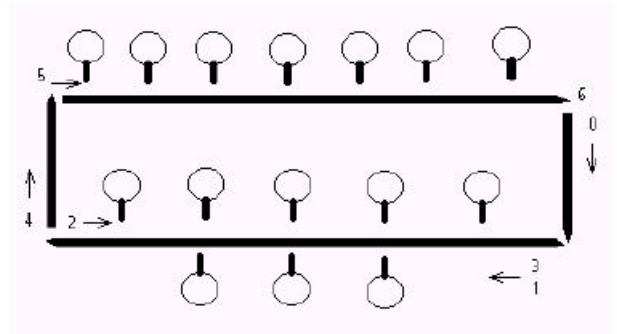
Como ya hemos indicado en el apartado anterior, partimos de un microbot comercial con fines educativos, el Tritt, que consta de lo siguiente:

- Placa controladora CT68 con microcontrolador 68HC11A1, que dispone de 512 bytes de EEPROM para el código del programa y 256 bytes de RAM para las variables.
- Placa CT293 que controla los motores y los sensores.
- Dos motores que controlan dos ruedas traseras independientes.
- Dos sensores de infrarrojos.

## 4. Software y estrategias de control

El programa aunque es bastante corto está muy optimizado en cuanto a tamaño por lo que es bastante críptico a la hora de leer. Para ello es fundamental entender las variables utilizadas que son las siguientes:

\*POSICI: El tablero tiene seis posiciones [0..6] siendo la 0 y la 6 la misma. Esta variable nos dice en que posición estamos. Se inicializa a 0.



\*KIFILA: Indica la fila de la que deberán retirarse los palillos. En caso de estar contando palillos, se pone a FF para que no quite.

\*FILCON: Indica las filas que quedan por contar. Se inicializa a 3. En caso de llegar a 0, indicará que el adversario no ha movido y por tanto dará lugar a un error.

\*PFILA[1..3]: Almacenan el número de palitos restantes en cada fila. Se inicializan a 3,5 y 7.

\*ULTCOL: Indica el último color leído por el sensor izquierdo. Se usa para controlar el conteo de los palitos y evitar posibles ruidos: para contar un palito, debe leer negro 256 veces, y puesto que hay dos transiciones para cada palito (blanco->negro y negro->blanco) se debe dividir el número de palitos contados por 2.

\*KONPAL: Es el contador parcial de los palitos de cada fila.

\*KIPALI: Indica el número de palitos a retirar.

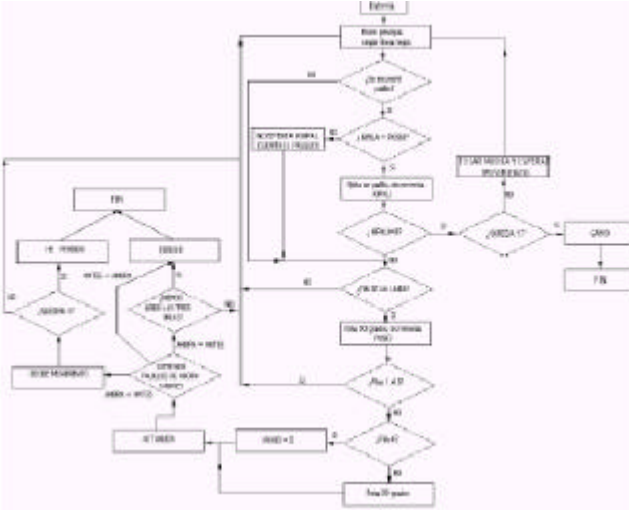
\*CONSEG: Contador de ULTCOL, cuenta hasta 256.

\*PUNTER: Puntero que apunta a la nota que esta tocando el robot.

\*CANCIO: Puntero que apunta a la dirección donde empieza la canción que queremos tocar dependiendo si quiere indicarnos si este en espera, si ha habido una trampa, si ha ganado o ha perdido.

\*VARIA: indica la el tiempo que dura cada nota se inicializa a \$3FF y como la música se puede repetir muchas veces en espera hay tres velocidades que son dividiendo la velocidad inicial por dos y en la cuarta repetición empieza otra vez.

El flujograma del programa es el siguiente:



**\*Batería**

En ésta sección se realiza la comprobación del nivel de la batería, para poder realizar los giros de 90 grados con exactitud. Para ello el robot comienza girando a la derecha hasta que el sensor izquierdo detecte la línea negra, y mide el tiempo que ha necesitado para realizar el giro, almacenándolo en la variable VELOCI para finalmente calcular el que necesitará para uno de 90 grados.

**\*Bucle principal**

Ésta sección se encarga de dirigir el robot por la línea negra sin que se salga ayudándose para ello del sensor central. Mientras sea detectado color negro, el robot continuará caminando en línea recta, y en el momento que deje de detectar negro realizará un giro pequeño hacia la derecha; si tras éste detecta negro la trayectoria ha sido corregida, en caso contrario realizará un giro en sentido opuesto de tamaño doble; si encuentra el negro sigue adelante y si no, significa que ha llegado al final de la línea. Además en este bucle se cuentan los palillos por el sensor izquierdo y en caso de tener que quitarlos se procede.

**\*Música**

Mientras el robot espera nuestro movimiento tocará por el altavoz incorporado la canción “No cambié” de Tamara, incrementando la velocidad progresiva y cíclicamente. Las notas se almacenan a partir de la dirección \$B790. Para tocar las notas a la frecuencia adecuada, se vale de multiplicar el periodo de cada nota por 0.96 (0.F1) para pasar a la siguiente. En cada byte se almacenan 2 notas, una en el nibble superior y otra en el inferior. La salida de la onda cuadrada correspondiente sale por el puerto B, el cual se incrementará. Esto quiere decir que a medida que avanzamos por los bits del puerto, la frecuencia de tal onda se verá duplicada en periodo, y por tanto pertenecerán a la escala anterior. Asimismo cuando el robot gane tocará otra melodía distinta.

**\*Decisión de movimientos**

A la hora de decidir el movimiento que debe realizar, el robot acude a una tabla de movimientos que se almacena en la parte final de la ROM, a partir de la dirección \$B7A0. En cada byte se almacenan dos movimientos, uno en la parte superior y otro en la parte inferior.

El robot acude siempre que puede a las jugadas ganadoras, que son las siguientes:  
 1-1-1 2-5-7 1-2-3 3-4-7 1-4-5 3-5-6 2-4-6 x-x-0

Éste será el número de palillos que dejará el robot al adversario.

En caso de no poder realizar éstas jugadas (es decir que se le ha dejado al robot una de éstas combinaciones), el robot quita un solo palito, el cual hemos programado de manera que despiste al adversario cuanto pueda. Anteriormente creamos una función que generaba un número pseudoaleatorio entre 1 y 3 que escogía una fila al azar, de donde se retiraba un palito, pero finalmente decidimos quitarla para optimizar el código y así poder introducir las dos melodías.

**\*Detección de errores**

Para evitar que se le haga trampa, al robot se le ha programado un algoritmo de detección de errores. En caso de que detecte un error, el robot girará indefinidamente y tocará una música consistente en el propio código del programa convertido a notas musicales. Detectará error en los siguientes casos:

- a) Si cuenta más palitos de los que había anteriormente en una cierta fila.
  - b) Si detecta que no se ha producido ningún movimiento
- Nota: El error que se generaría al coger palillos de dos filas lo hemos quitado por cuestiones de espacio.

**5. Características físicas y eléctricas más relevantes**

Las características físicas y eléctricas no son para nada relevantes en cuanto al objetivo del proyecto, además la velocidad es bastante variable dependiendo del nivel de las pilas. Para más información consultar la página web de la empresa proveedora del Tritt, [www.microbotica.es](http://www.microbotica.es)

**6. Conclusiones**

La experiencia obtenida ha sido bastante gratificante, pese a la innumerable cantidad de bugs y de problemas de depuración que han ido apareciendo a lo largo del desarrollo.

Por otra parte estamos muy orgullosos de los resultados obtenidos que incluso han sido mejorados en relación con las expectativas iniciales.