µBOT HACROX

CARLOS SORIA MARTINEZ , RICARDO GOMEZ GOMEZALES, JESÚS SOTO CARRION, MACARENA ZOFIO PEREZ

csoria@mfom.es, ricardo.gg@terra.es , jesussoto@wanadoo.es, macarenaz@terra.es

Universidad Pontificia de Salamanca en Madrid

Resumen

Hacrox es un robot pensado inicialmente como robot entrenador sencillo para facilitar el estudio de la micro botica en entornos educativos. Es, o pretende ser, un modelo robusto físicamente, sencillo, barato y fácil de construir.

1. Introducción

Entre las principales característica que merece la pena destacar aparte de las anteriores mencionadas, es la facilidad del intercambio de sensores, que en esta configuración son ópticos (infrarrojos de distancia corta.), por otros táctiles, infrarrojos de distancias largas o combinaciones de estos. Al estar soportados por una estructura delantera separada, ajustable y muy fácil de intercambiar.
Así mismo se usan para la parte electrónica placas de circuitos ya desarrollados y ampliaante usados en estos entornos.

2. Plataforma mecánica usada

La estructura principal, las ruedas y los módulos intercambiables están construidos en PVC-espumoso un material barato y fácil de manipular. Las diferentes piezas han sido unidas con pegamentos para plástico rígido .En la parte delantera del robot, hay una regleta para los sensores CNY70. Esta es regulable en altura y en profundidad. Las ruedas están fabricadas en PVC-espumoso recubierto por una cinta de caucho muy blanda para darle al robot una adherencia elevada.

3. Arquitectura hardware

El sistema completo esta basado en un microcontrolador de Motorola 68HC11. Este funciona con una frecuencia de reloj de 8 Mhz. Consta de 256 bytes de RAM y 512 bytes de Eeprom. Lo cual resulta un microcontrolador muy versátil para aplicaciones de robótica. Los ojos de robot son 6 fotosensosres del tipo CNY70. La distancia optima de funcionamiento es de 2 a 3 mm. Este dispositivo esta compuesto por un fotodiodo emisor y un fototransistor como receptor. Este dispositivo no utiliza modulación de ningún tipo. El circuito de comunicación con el PC esta compuesto por un circuito de interfaz de comunicaciones serie asíncronas implementado en un chip MAX232. Los programas son cargados desde el PC al microcontrolador.

4. Características físicas y eléctricas mas relevantes

Es de destacar su pequeño tamaño, su robustez mecánica , la facilidad de cambio de sensores, y el amplio radio de las ruedas respecto al tamaño del robot. Lo cual le permite recorrer distancias proporcionalmente mayores. En el aspecto eléctrico cabe destacar la posibilidad de por un lado disponer de una única alimentación para el funcionamiento del microcontrolador , placas de interfaz , motores y sensores o el separar eléctricamente la alimentación de los motores , aniñándole una fuente de alimentación diferente para estos del resto de la electrónica..

5. REFERENCIAS

[1] Manual de la placa CT6811 de MICROBOTICA

[2] Manual de la placa ct293+ de MICROBOTICA

[3] Referencia técnica del 68hc11 del Mototorola

1. INTRODUCCION A LA PLACA CONTROLADORA CT6811:

La CT6811 es una tarjeta para el desarrollo de aplicaciones hardware y software basadas en el microcontrolador 68HC11. No sólo sirve para el desarrollo de prototipos, sino que también está pensada para funcionar en **sistemas terminado**s. Se trata sobre todo de una tarjeta muy versátil, que se puede emplear como:

- **Tarjeta entrenadora:** Conectándose a ordenadores PC a través del puerto serie. Es posible enviar programas desde el PC a la CT6811 para que los ejecute. De esta manera, la tarjeta es muy útil para la depuración de programas en fase de pruebas y para aprender a programar el 68HC11 desde un punto de vista práctico: todos los programas creados sobre el 'papel' se pueden ejecutar en un 68HC11 'de verdad'

- **Tarjeta autónoma de contro**l. La CT6811 funciona en modo autónomo, es decir,sin conectarse al PC. Cada vez que se encienda la tarjeta, se ejecutará el programa que previamente se habrá grabado en la memoria EEPROM del 68HC11.Conectando periféricos a través de los puertos de expansión, se consiguen desarrollar sistemas inteligentes de control: control de las luces de una casa, control de la maqueta de un tren, manejo de microbots, alarmas ...

- **Periférico inteligente del PC:** También es posible utilizar la tarjeta para comunicar el PC con el mundo exterior. La CT6811 actuaría como un periférico conectado al PC a través del puerto serie. Este periférico 'inteligente' puede recibir órdenes del PC y actuar en consecuencia. Se pueden realizar aplicaciones del tipo: digitalización de señales y presentación en el PC, diseño de joysticks no convencionales, control de luces de la casa a través del PC, control de maquetas de tren visualizando en el PC la situación de los trenes, semáforos...

2. CARACTERISTICAS TECNICAS DEL MICRO CONTROLADOR 68HC11:

Por ello incorpora todas las características de este micro:

- 512 Bytes de EEPROM en los modelos A1 y A8
- 256 Bytes de memoria RAM interna
- Temporizador de 16 bits
- 3 capturadores de entrada
- 5 comparadores con salida hardware
- Un acumulador de pulsos
- Comunicaciones serie asíncronas
- Comunicaciones serie Síncronas
- 8 canales de conversión analógico digital
- Interrupciones en tiempo real
- 4 puertos de E/S

Bus Expansión: Bus de expansión dividido en 6 puertos de 10 bits cada uno.

Desde estos puertos se tiene acceso a todas las patas del micro.

Switches de configuración: 2 switches de configuración del modo de arranque
de la placa (bootstrap, single chip, expanded, special test) y 2 switches
disponibles para aplicaciones del usuario

Led de pruebas conectado al bit 6 del puerto A. Este led se puede conectar y
desconectar por medio de un jumper

Pulsador de reset/pruebas IRQ: Pulsador para inicializar la tarjeta. Este
pulsador se puede utilizar también, cambiando un jumper, como un pulsador de
propósito general conectado a la entrada de interrupciones IRQ.

Niveles VRH y VRL configurables a VCC y masa respectivamente mediante 2
jumpers

Modos de funcionamiento entrenador/autónomo configurables mediante un
jumper

Alimentación a través de clemas o conector tipo jack

Conexión directa al PC por medio de un cable tipo teléfono

Reset software y hardware de la placa.

El núcleo central de la tarjeta CT6811 está constituido por el microcontrolador 68HC11A1 de Motorola. Los demás bloques surgen como una extensión del 68HC11.La tarjeta se puede dividir en 8 bloques: el corazón de la placa (68HC11), el circuito de reloj, el circuito de inicialización o reset, el circuito de pruebas, el circuito de comunicaciones con el PC, la configuración del sistema, los puertos de expansión y la alimentación del sistema completo. A continuación se explica cada bloque:

Microcontrolador 68HC11A1:
Se trata de un microcontrolador de 8 bits. Además de una CPU que le permite ejecutar instrucciones y realizar operaciones aritmético lógicas, dispone en el propio chip de memorias ROM, RAM y EEPROM, así como una serie de periféricos integrados que hacen de este micro una herramienta muy potente.

Circuito de reloj: Este circuito permite que el micro obtenga la señal de reloj necesaria para funcionar. Está constituido por un cristal de 8MHZ, que es el valor máximo que permite el 68HC11. Además, con este valor, el micro es capaz de comunicarse con el PC a las velocidades de 1200, 2400 ó 9600 baudios.

Circuito de reset: El circuito de reset permite la correcta inicialización del 68HC11. Se ha diseñado de tal forma

que es posible realizar un reset por 'hardware', apretando un pulsador, o bien realizar un reset por software desde el PC, cuando la CT6811 está funcionando en modo entrenador. El reset software es muy útil cuando se están desarrollando microbots, pues cada vez que hay que cargar un programa nuevo en el microbot, no es necesario desplazarse hasta él y apretar el botón de reset, sino que directamente desde el PC se puede llevar a cabo.

Configuración: La parte de configuración de la CT6811 está constituida por 6 jumpers y 4 switches. Dos de los switches permiten configurar el 68HC11 para trabajar en cualquiera de los 4 modos para los que está diseñado: bootstrap, single chip, expanded y special. Los otros 2 switches están a disposición del usuario através de los puertos de expansión para configurar tarjetas periféricas conectadas a la CT6811. Los 6 jumpers permiten configurar la CT6811 para realizar diferentes funciones, que se comentarán más adelante.

Circuito de pruebas: La CT6811 dispone de un led conectado al bit 6 del puerto A del 68HC11. Este led, que se puede conectar y desconectar a través de un jumper, es muy útil para la realización de software de pruebas. El pulsador de la tarjeta, se puede configurar mediante un jumper para actuar bien como pulsador de reset o bien como un pulsador normal conectado a la entrada de interrupción IRQ del 68HC11. De esta forma, es posible realizar pequeños programas de prueba que lean el pulsador y actúen en consecuencia.

Comunicaciones con el PC: Esta es una de las partes más importante. A través del circuito de comunicaciones es posible comunicarse con el PC para intercambiar información. Las comunicaciones se realizan a través del puerto serie del PC, mediante a norma RS-232. La velocidad máxima compatible con el PC es de 9600 baudios, aunque el 68HC11 es capaz de transmitir a mucha más velocidad.

Puertos de expansión: La CT6811 dispone de 6 puertos de expansión donde es posible conectar los diferentes periféricos. Todas las señales del 68HC11, salvo las correspondientes al reloj (EXTAL y XTAL), son accesibles desde los puertos de expansión. Cinco de los puertos coinciden con los 5 puertos del 68HC11: A,B,C,D y E. El sexto puerto está destinado a llevar las señales de control del 68HC11.

Alimentación: La tensión de alimentación de la CT6811 oscila entre 4.5 y 5.5 voltios. La tarjeta dispone de un conector hembra de tipo jack cilíndrico para la conexión de un transformador, y dos bornas ajustables mediante tornillos para la

3. CONFIGURACIÓN DE LOS JUMPERS

Existen 8 jumpers que nos permiten configurar aspectos del 68HC11 y de la CT6811.

Jumpers JP1 y JP2: Configuración de los niveles VRH y VRL del micro. Estos jumpers se utilizan para fijar los niveles de tensión de las entradas VRH y VRL del 68HC11. El jumper JP1 actúa sobre VRH. Si está colocado, la pata VRH está cortocircuitada con VCC. Si no se coloca, esta tensión se puede fijar desde el exterior, a ravés del puerto E. El jumper JP2 actúa sobre VRL. Si está colocado, el pin VRL se cortocircuita con GND. En caso contrario queda accesible desde el exterior a través del puerto E. Estos dos jumpers sólo se utilizan cuando se trabaja con el conversor A/D del 68HC11. Los jumpers normalmente estarán colocados. Sólo para aplicaciones muy específicas habrá que quitarlos y acceder a VRL y VRH desde el puerto E.conexión de cables de alimentación, provenientes por ejemplo de pilas o baterías.

Jumper JP3: Conexión/desconexión del LED de pruebas. El diodo LED está conectado al bit 6 del puerto A del 68HC11 cuando el jumper JP3 está colocado. Si se quita, el LED quedará desconectado y el bit 6 del puerto A podrá ser utilizado para otros propósitos que no sean encender el led.

Jumper JP4: **Configuración del pulsador: RESET/IR**Q. El jumper JP4 se trata de un jumper triple, que puede estar colocado en la zona de la izquierda, cortocircuitando los dos pines situados más a la izquierda, o puede estar colocado en la zona de la derecha, cortocircuitando los dos pines situados más a la derecha. Cuando el jumper está situado en la zona de la derecha (posición marcada como RST en la tarjeta), el pulsador actúa sobre la entrada de reset del 68HC11. Por tanto, en esta posición, cada vez que se apriete el pulsador, **se realizará un reset hardware de la CT681**1. Cuando el jumper está situado en la posición de la izquierda (marcada como IRQ), el pulsador se conecta a la entrada de interrupción no enmascarable IRQ del 68HC11. De esta forma, podemos utilizar el pulsador en nuestros programas para realizar pruebas. Si el jumper está en esta posición, sólo se puede realizar un reset de la CT6811 a través del PC, utilizando el reset software.

Jumper JP5 1 : Configuración modo de funcionamiento de la CT6811:

- **Entrenador/autónomo.** Este es uno de los jumpers más importantes. Nos permiten decidir el funcionamiento de la CT6811: como una tarjeta autónoma o como una tarjeta entrenadora conectada al PC. Cuando el jumper está quitado, la tarjeta está configurada en modo entrenador. Todo el software habrá que cargarlo desde el PC. Sin embargo, cuando este jumper está puesto, al pulsar reset, la CT6811 comenzará a ejecutar el programa que tiene grabado en la EEPROM. No necesita del PC para cargar ningún software.
- **Jumper JP6: Conexión/desconexión del LVI.** Este jumper conecta o desconecta el dispositivo MC34064 a la entrada de reset. Cuando está conectado, el dispositivo MC34064 previene que se pueda borrar el contenido de la memoria EEPROM del 68HC11. Cuando se detectan niveles de tensión por debajo de los niveles normales de funcionamiento del 68HC11, el MC34064 realizará un reset del 68HC11. Esto

ocurre al encender y apagar la CT6811. Sin embargo, puede ocurrir que mientras la CT6811 esté funcionando, aparezca un pico de caída de tensión y el micro sufra un reset. Por ello se da la opción de desconectar el MC34064 quitando el jumper JP6.

Nota importante: El jumper JP5 lo que está haciendo es cortocircuitar o no los pines TX y RX del 68HC11 a través de una resistencia de pull-up. Por ello, si estamos trabajando en modo autónomo, NO FUNCIONARAN LAS COMUNICACIONES CON EL PC.

Jumper JP7: Conexión/Desconexión del Reset Software. Ya se ha comentado la posibilidad de poder realizar un *reset software* (desde el PC) de la CT6811. Esta señal de *reset* se canaliza por el DTR del puerto serie, esto provoca que algunos programas de comunicaciones no puedan comunicarse con la CT6811. Por eso se ha añadido este *jumper* que permite separar el DTR de la señal de *reset*, de forma que se pierde el *reset sotware* pero se gana el poder utilizar otros programas de comunicaciones. Si el *jumper* está colocado abajo (on) estará activado el *reset softwar*e, si está colocado arriba (off) estará desactivado el *reset software*.

Jumper JP8: Liberación de la XIRQ. La tarjeta CT6811 se puede utilizar con otros modelos de la familia del microcontrolador 68hc11A1. Uno de ellos es el modelo 68hc11E9, que dispone de 12Kbytes de EPROM. Para poder programar esta memoria no volátil es necesario introducir 12v por la entrada XIRQ, que como se verá más adelante se encuentra en el bus de control. Para poder introducir esta tensión sin dañar los circuitos de la tarjeta CT6811es necesario quitar el *jumper* JP8. Esto sólo se hará cuando se vaya a programar la EPROM, una vez finalizada la programación se liberará la XIRQ y se volverá a poner el *jumpe*r.

4. CONFIGURACIÓN TÍPICA DE LA CT6811 EN MODO ENTRENADOR

La configuración típica en modo entrenador es : todos los jumpers colocados excepto el jumper JP5, el jumper JP4 situado a la derecha, y el jumper JP7 abajo. Los switches 1 y 2 deben estar ‘hacia abajo’. La CT6811 tiene que estar alimentada y conectada al PC a través del cable de teléfono. La alimentación se hace a través de un transformador conectado a la red o a través de baterías o pilas, pero **nunca ambos simultáneament**e.

5. CONFIGURACIÓN TÍPICA DE LA CT6811 EN MODO AUTÓNOMO

Todos los jumpers colocados . El jumper JP4 debe estar situado en la posición de la derecha. En esta situación, si la CT6811 está alimentada, cada vez que se apriete el pulsador, el 68HC11 comenzará a ejecutar el programa previamente almacenado en la memoria EEPROM.

6. DESCRIPCIÓN DE LOS PUERTOS

- **Puerto** B: Igual que el puerto A pero con el puerto B del 68HC11. El puerto B del 68HC11 sirve también como parte alta del bus de direcciones en caso de ampliación del micro con memoria externa. Cuando funciona en modo no expandido, el bit PB0 se corresponde con A8, PB1 con A9, ..., PB7 con A15.

- **Puerto C:** A este puerto de expansión se lleva el puerto C del 68HC11. Este puerto funciona como un puerto normal de E/S cuando no hay conectada memoria externa. Cuando se conecta memoria externa, este puerto lleva el byte bajo del bus de direcciones multiplexado con el bus de datos. El bit PC0 se corresponde con AD0, PC1 con AD1,..., PC7 con AD7.

- **Puerto D:** Este puerto está constituido por el puerto D del 68HC11 junto con las señales TXPC, RSTPC, RXPC y GND. El puerto D del 68HC11 dispone de 6 bits, que están compartidos con el SCI y el SPI del 68HC11: PD0/RX, PD1/TX, PD2/MISO, PD3/MOSI, PD4/SCK, PD5/SS. La señales TXPC y RXPC se utilizan para conectar a la CT6811 otros dispositivos que utilicen la norma de comunicaciones RS-232. Se denominan TXPC y RXPC porque están cortocircuitados con las señales TX y RX provenientes del PC. Si por ejemplo se quiere conectar una calculadora HP a la CT6811, habrá que hacerlo a través de las señales TXPC y RXPC. Por TXPC se transmite la información **hacia** la CT6811; por RXPC se recibe la información **de** la CT6811. La señal RSTPC está reservada para usos futuros. **No conectar nada en este pin.**

- **Puerto E:** Constituido por el puerto E del 68HC11. Este puerto en el 68HC11 tiene una doble función: puerto de entrada de 8 bits ó 8 canales de conversión A/D. Las señales VRL y VRH se utilizan para introducir las señales de referencia alta y baja del conversor. Si los jumpers JP1 y JP2 están colocados, por estos podemos obtener la alimentación: VRH=VCC, VRL=GND. Si no están colocados estos jumpers, podremos introducir las tensiones de referencia necesarias para nuestra aplicación.

- **Control:** Por este puerto se han ‘recopilado’ una serie de señales del 68HC11 para el control. Por los pines SW3 y SW4 se sacan las señales correspondientes al estado de los switches de usuario 3 y 4.

7. Alimentación

La CT6811 tiene una tensión nominal de alimentación de 5 voltios, aunque se puede alimentar sin ningún problema entre 4.5 y 5.5 voltios. Dispone de dos conectores diferentes de alimentación para alimentar la placa bien por medio de un transformador entre 4.5 y 5.5 voltios conectado a la red o bien directamente a través de pilas o baterías.

La alimentación a través del transformador se emplea cuando se está utilizando la CT6811 como una entrenadora. En este caso, puesto que la entrenadora tiene que estar junto al PC, lo mejor es alimentar mediante el transformador. Sin embargo, cuando está funcionando en modo autónomo, es más cómodo utilizar una batería independiente.
Es importante hacer notar que **la parte exterior del jack macho debe ser GN**D, mientras que **la parte interior debe ser VC**C. En la mayoría de los transformadores comerciales, la polaridad se puede invertir cambiando un switch. **Antes de conectar un transformador asegurarse de que la polaridad es la correcta: Masa por la parte exterior(-), alimentación (+) por la parte interio**r.. La polaridad correcta se encuentra indicada en la propia CT6811. Estas bornas disponen de dos tornillos en la parte superior que permiten fijar los cables de alimentación que se introduzcan. Los terminales de VCC de las bornas y el conector jack se encuentran cortocircuitados, lo mismo que los terminales de GND. Por ello, **nunca se debe alimentar la CT6811 mediante un transformador y baterías a la ve**z. O se utiliza el transformador, o se utilizan las bornas.

8. Modos de funcionamiento del 68HC11

El 68HC11 puede funcionar de 4 modos diferentes: Single chip, expanded, bootstrap y special test. En cada modo se dispone de un mapa de memoria diferente.

Single Chip: En este modo de funcionamiento, el mapa de memoria del 68HC11 está constituido por la memoria RAM, la memoria EEPROM, los registros de control y la memoria ROM. Este modo está pensado para funcionar cuando existe un programa grabado en la ROM, de tal manera que al arrancar se comience a ejecutar el programa indicado por los vectores de interrupción que se encuentran en la ROM.

Expanded: Además del mapa de memoria del modo single chip, es posible acceder al resto de las posiciones de memoria conectando memorias externas. El precio a pagar es que se pierden 2 puertos de E/S el puerto B y el puerto C. En este modo se puede utilizar la memoria ROM interna, pero también es posible deshabilitar esta ROM y acceder a los vectores de interrupción que se encuentren en una memoria externa.

Bootstrap: Este modo difiere del single chip en que los vectores de interrupción no se encuentran en la memoria ROM de 8K sino que se encuentran en otra memoria ROM, llamada la ROM de arranque. Al arrancar en este modo, automáticamente comienza a ejecutarse el programa BOOTSTRAP que se encuentra en esta ROM.

Special Test: Igual que el modo Bootstrap con la salvedad de que se puede acceder a memoria externa. Este modo se utiliza para realizar pruebas de fábrica. En este modo especial se tiene acceso a determinados registros de control que en otros modos están protegidos. Con la tarjeta CT6811, estos modos de funcionamiento del micro se pueden configurar mediante 2 switches. En los modos de trabajo Entrenador y autónomo de la CT6811, se utiliza siempre el modo bootstrap del 68HC11. Los modos special test y expanded se pueden utilizar si se conecta a la CT6811 una tarjeta periférica con memoria, como por ejemplo la CT3216. El modo single chip se utiliza cuando existe algún programa grabado en la memoria ROM del micro, como es el caso de algunos micros del tipo A1 y A0 que llevan grabados en la ROM el programa BUFFALO de Motorola.

9. Modelos De 68hc11 Y Sus Caracteristicas Principales

La tarjeta CT6811 incorpora como microcontrolador principal el MC68HC11A1, pero este puede ser sustituido por otros de la familia, en concreto por los modelos de las Familias A, D y E. En la tabla siguiente se muestran los modelos que los autores han utilizado en alguna aplicación de la CT6811.

Modelo	RAM	ROM	EPROM	EEPROM
MC68HC11A8	256	8K	0	512
MC68HC11A1	256	0	0	512
MC68HC11A0	256	0	0	0
MC68HC811E2	256	0	0	2048
MC68HC711E9	512	0	12K	512

Todos estos modelos tienen los mismos recursos internos, 8 conversores, SPI, SCI, capturadores, comparadores, etc, únicamente se diferencian en el tipo y cantidad de la memoria disponible.
La tarjeta CT6811, como ya se ha dicho, trae de serie el MC68HC11A1. Es un microcontrolador fácilmente obtenible y al tener EEPROM interna se puede utilizar en modo autónomo, aunque para ello sea necesario arrancar en BOOTSTRAP con el jumper JP5 puesto. Es decir se perderán las comunicaciones serie con el PC. Un truco para resolver esto es arrancar el sistema con un pulsador conectado en el lugar del jumper, o mejor aún, con un pequeño sistema que cortocircuite dicho jumper justo después de hacer un reset para acto seguido liberar la conexión.
Un inconveniente no ajeno al programador es la capacidad de memoria disponible en el sistema. Inicialmente la CT6811 tiene 256 bytes de RAM y 512 de EEPROM, aunque existe un módulo de ampliación de memoria que añade 32K de RAM y 32 de EEPROM externas. El inconveniente de la ampliación es la pérdida del puerto B y del puerto C, es decir un gran número de entradas y salidas. Por eso los autores aconsejan sustituir el modelo de microcontrolador, se ahorra espacio, no se pierden puertos y la fiabilidad será mayor, el sistema se reduce en componentes. Además la tarjeta CT6811 ha

sido diseñada para servir de entrenadora primero y luego de producto final. Otras tarjetas, como TRANTOR, también desarrollada por el Grupo J&J incorporan memoria RAM y EEPROM externas. TRANTOR ha sido pensada principalmente como entrenadora de software. Una vez desarrollado el software se le encargaría a Motorola un MC68HC11A8 con el programa grabado en ROM. Esto tiene el inconveniente del precio, por eso una opción bastante buena es utilizar microcontroladores MC68HC711E9 o MC68HC811E2.

El MC68HC811E2 tiene 2K de memoria EEPROM situados desde la posición $F800 hasta la $FFFF, con lo cual los vectores de interrupción están en EEPROM, incluído el de reset. Esta característica hace que el modo autónomo sea mucho mejor. Ahora se tienen dos posibilidades para acceder a dicho modo. La primera es hacer un reset en BOOTSTRAP y utilizar el jumper JP5, la segunda es configurar el vector de interrupción del reset para que apunte al principio del programa en EEPROM. Se configura la CT6811 para arrancar en modo Single Chip, y al hacer un reset el programa grabado en EEPROM empezará a funcionar independientemente de la situación del jumper JP5, es decir se conservan las comunicaciones serie con el PC. El MC68HC711E9 tiene otras características interesantes. La primera es que el tamaño de la RAM y de la EEPROM coincide. Esto es útil a la hora de probar software. Con los modelos anteriores los programas de la EEPROM pueden ser mayores que los de la RAM, por lo tanto no se pueden probar en ella. La segunda característica es que incluye 12K de EPROM situadas en las posiciones $D000-$FFFF. Ahora también se puede inicializar el vector de reset, por lo que todo lo
referente al modo autónomo del MC68HC811E2 sigue siendo válido salvo una pequeña diferencia. En este caso la memoria es EPROM y sólo se puede grabar una vez, a no ser que se disponga del modelo con ventana (OTP) para poder borrarla. Es útil disponer del modelo con ventana para desarrollar software y luego grabar las versiones definitivas en modelos sin ventana. Otra solución es utilizar tarjetas con memoria externa y cuando el programa este terminado traspasarlo a los modelos E9. Las soluciones son variadas y la más adecuada dependerá del tipo, tiempo, coste, etc... de la aplicación o desarrollo a implementar.
Por último mencionar la utilidad del MC68HC11A0. En este caso sólo se puede usar la memoria RAM, en concreto los 256 bytes. ¿ Por qué no tiene EEPROM , la respuesta está en que son modelos del MC68HC11A1 que tienen estropeada dicha memoria. En lugar de retirarlos del mercado se les desactiva la misma y se vende como un MC68HC11A0. Los autores los han utilizado en tarjetas insertadas en el PC, donde el microcontrolador era programado desde los programas de aplicación del PC. Según la aplicación se reprogramaba en el acto la tarjeta, a modo de periférico programable. Al tener la memoria del PC no era imprescindible la EEPROM y debido a que al principio el coste de este modelo era significativamente menor que el del resto, se podía prescindir de un modelo con EEPROM. Actualmente no hay tanta diferencia y lo anterior se puede cuestionar, sin olvidar que la CT6811 con un MC68HC11A0 solamente se podría usar en modo entrenadora, no como producto final.
Para más información sobre la familia del Motorola MC68HC11 se recomienda acudir al manual de referencia técnica de Motorola. ‘MC68HC11 REFERENCE MANUAL’.

10. Set De Instrucciones Del 68hc11:

ABA	Añadir el contenido del acumulador B al acumulador A
ABX	Añadir el contenido del acumualdor B (sin signo) al contenido del registro X
ABY	Añadir el contenido del acumualdor B (sin signo) al contenido del registro Y
ADCA	Añadir al acumulador A un dato y el bit de acarreo.
ADCB	Añadir al acumulador B un dato y el bit de acarreo.
ADDA	Añadir un dato al registro A
ADDB	Añadir un dato al registro B
ADDD	Añadir un dato de 16 bits al registro D
ANDA	Realizar una operación lógica AND entre un dato y el acumulado A.Resultado en A
ANDB	Realizar una operación lógica AND entre un dato y el acumulador B.Resultado en B
ASLA	Desplazar un bit a la izquierda el acumulador A.
ASLB	Desplazar un bit a la izquierda el acumulador B
ASLD	Desplazar un bit a la izquierda el acumulador D
ASRA	Desplazar un bit a la derecha el acumulador A
ASRB	Desplazar un bit a la derecha el acumulador B
BCC	Saltar si no hay acarreo
BCLR	Poner a cero bits de la memoria
BCS	Saltar si hay acarreo
BEQ	Saltar si igual a cero
BGE	Saltar si mayor que o igual a cero
BGT	Saltar si mayor que cero
BHI	Saltar si es mayor
BHS	Saltar si mayor o igual
BITA	Comprobar el bit especificado del acumulador A
BITB	Comprobar el bit especificado del acumulador B
BLE	Saltar si menor que o igual a cero
BLO	Saltar si menor (Mismo que BCS)
BLS	Saltar si menor o igual
BLT	Saltar si menor que cero
BMI	Saltar si negativo
BNE	Saltar si no es igual
BPL	Saltar si es positivo
BRA	Salto incondicional
BRA	Saltar si los bits especificados están a cero
BRA	No saltar nunca. (equivale a un NOP de 2 bytes)
BRA	Saltar si los bits especificados están a uno
BSET	Poner los bits especificados a uno
BSR	Saltar a una subrutina
BVC	Saltar si no ha habido overflow
BVS	Saltar si ha habido overflow
CBA	Comparar el acumulador A con el B
CLC	Poner a cero bit de acarreo
CLI	Permitir las interrupciones
CLR	Poner a cero el contenido de memoria especificado
CLRA	Poner a cero el acumulador A
CLRB	Poner a cero el aumulador B
CLV	Poner a cero el bit de overflow
CMPA	Comparar el acumulador A con un dato
CMPB	Comparar el acumulador B con un dato
COMA	Complementar a uno el acumulador A
COMB	Complementar a uno el acumulador B
COM	Complementar a uno el contenido de memoria especificado
CPD	Comparar el registro D con un dato
CPX	Comparar el registro X con un dato
CPY	Comparar el registro Y con un dato
DAA	Ajuste decimal
DEC	Decrementar una posición de memoria especificada
DECA	Decrementar el acumulador A
DECB	Decrementar el acumulador B
DES	Decrementar el puntero de pila SP
DEX	Decrementar el registro X
DEY	Decrementar el registro Y
EORA	Operación XOR entre un dato y el acumulador A
EORB	Operación XOR entre un dato y el acumulador B
FDIV	División
IDIV	División entera
INC	Incrementar el contenido de una posición de memoria
INCA	Incrementar el acumulador A
INCB	Incrementar el acumulador B
INS	Incrementar el puntero de pila SP
INX	Incrementar el registro X
INY	Incrementar el registro Y
JMP	Salto incondicional
JSR	Salto a una subrutina
LDAA	Cargar un dato en el acumulador A
LDAB	Cargar un dato en el acumulador B
LDD	Cargar un dato de 16 bits en el registro D
LDS	Cargar un dato de 16 bits en el puntero de pila SP
LDX	Cargar un dato de 16 bits en el registro X
LDY	Cargar un dato de 16 bits en el registro Y
LSL	Desplazamiento de un bit hacia la izquierda del contenido de una posición de memoria
LSLA	Desplazar el acumulador A un bit hacia la izquierda
LSLB	Desplazar el acumulador B un bit hacia la izquierda
LSLD	Desplazar el acumulador D un bit hacia la izquierda
LSR	Desplazaar el contenido de una posición de memoria un bit hacia la derecha
LSRA	Desplazar el acumulador A un bit hacia la derecha
LSRB	Desplazar el acumulador B un bit hacia la derecha

LSRD	Desplazar el registro D un bit hacia la derecha
MUL	Multiplicación sin signo
NEG	Complementar a dos el contenido de una posición de memoria
NEGA	Complementar a dos el contenido del acumulador A
NEGB	Complementar a dos el contenido del acumulador B
NOP	No operación
ORAA	Realizar la operación lógica OR entre un dato y el acumulador A
ORAB	Realizar la operación lógica OR entre un dato y el acumulador B
PSHA	Meter el acumulador A en la pila
PSHB	Meter el acumulador B en la pila
PSHX	Meter el registro X en la pila
PSHY	Meter el registro Y en la pila
PULA	Sacar el acumulador A de la pila
PULB	Sacar el acumulador B de la pila
PULX	Sacar el registro X de la pila
PULY	Sacar el registro Y de la pila
ROL	Rotación a la izquierda del contenido de una posición de memoria
ROLA	Rotar a la izquierda el acumulador A
ROLB	Rotar a la izquierda el acumulador B
ROR	Rotar a la derecha el contenido de una posición de memoria
RORA	Rotar a la derecha el contenido del acumulador A
RORB	Rotar a la derecha el contenido del acumulador B
RTI	Retorno de interrupción
RTS	Retorno de subrutina
SBA	Restar un dato al acumulador A
SBCA	Restar un dato y el bit de acarreo al acumulador A
SBCB	Restar un dato y el bit de acarreo al acumulador B
SEC	Poner a uno el bit de acarreo
SEI	Inhibir las interrupciones
SEV	Poner a uno el bit de overflow
STAA	Almacenar el acumulador A en una posición de memoria
STAB	Almacenar el acumulador B en una posición de memoria
STD	Almacenar el registro D
STOP	Para el reloj del sistema
STS	Almacenar el puntero de pila SP
STX	Almacenar el registro X
STY	Almacenar el registro Y
SUBA	Restar un dato al acumulador A
SUBB	Restar un dato al acumulador B
SUBD	Restar un dato al registro D
SWI	Provocar una interrupción software
TAB	Tansferir el acumulador A al acumulador B
TAP	Transferir el acumulador A al registro CCR
TBA	Transferir el acumulador B al acumulador A
TEST	Instrucción de test. Sólo se puede ejecutar en el modo de test
TPA	Transferir el registro CCR al acumulador A
TST	Comprobar si una posición de memoria está a cero
TSTA	Comprobar si el acumulador A está a cero
TSTB	Comprobar si el acumulador B está a cero
TSX	Transferir el puntero de pila al registro X
TSY	Transferir el puntero de pila al registro Y
TXS	Transferir el registro X al puntero de pila
TYS	Transferir el registro Y al puntero de pila
WAI	Esperar a que se produzca una interrupción
XGDX	Intercambiar los valores de los registro X y D
XGDY	Intercambiar los valores de los registros Y y D

11. Introducción A La Tarjeta De Expancion Ct293+:

La CT293+ es una nueva tarjeta que proporciona al sistema Tower la posibilidad de controlar motores y sensores. Esta diseñada para adaptarse perfectamente a la CT6811 y poder controlarla sin ninguna variación tanto en Bootstrap como en Single Chip. La tarjeta también puede controlarse desde otro sistema por ejemplo el puerto paralelo de un PC. Esta tarjeta es la versión mejorada de la CT293, pero se ha respetado la compatibilidad. Los programas de su antecesora valen también para la nueva, lo único que hay que verificar es la situación de los sensores y motores. La CT293+ es el soporte principal de los dos primeros niveles de la torre BOT, con ella se proporciona el movimiento a los sistemas de control y la capacidad de analizar el entorno. Con esta placa y con la CT6811 se puede obtener la plataforma de un microbot.
Los prototipos Quark y Monumental desarrollados en enero de 1997 por el Grupo J&J son ejemplos de microbots. El prototipo Quark es capaz de seguir una línea negra, pero también puede realizar recorridos que previamente grabados. Esto último se consigue gracias al sistema de encoders que lleva instalado. Monumental fue el ganador del primer concurso de microbots de sumo celebrado en la Universidad de Informática en la UAM.
La tarjeta CT6811 es el sistema microcontrolador usado para la programación y la tarjeta CT293+ es la interfaz entre la tarjeta anterior y los recursos externos (motores, sensores, entadas digitales y entradas analógicas).
Las características de la tarjeta CT293+ se pueden resumir en:

1. Posibilidad de control de dos motores de continua o uno paso_paso.
2. Capacidad para leer cuatro sensores de infrarrojos, pudiendo ser estos optoacopladores.
3. Disponibilidad de 8 entradas digitales de propósito general con la posibilidad de usarlas como entradas analógicas.
4. Conexión de los elementos externos con clemas.
5. Alimentación de los motores externa o interna.

En el resto de la memoria se describen los conocimientos que se consideran necesarios para sacar el máximo partido a esta tarjeta. Todo ello con programas de ejemplo debidamente comprobados.

12. Descripción De Los Elementos De La Ct293+:

1. DISPOSICIÓN Y EXPLICACIÓN DE LOS COMPONENTES DE LA CT293+:

La placa, básicamente tiene dos bloques independientes. El primero de ellos (BLOQUE A) se encarga de los motores y de los sensores de infrarrojos, mientras que el segundo (BLOQUE E) controla las entradas digitales/analógicas. Hay un bus de control para cada bloque, llamados PUERTO A y PUERTO E. El usuario que este familiarizado con la CT6811 se dará cuenta del motivo de esta nomenclatura. El bus A (Puerto A) es el que maneja el bloque A, es decir motores e infrarrojos, el bus E (Puerto E) maneja el bloque E, que tiene las entradas analógico/digitales. Situación de los componentes de la El bloque A esta formado por el chip L293B 1 que gestiona la potencia de los motores y el chip 40106 que tiene dos funciones. La primera consiste en realizar una pequeña lógica para facilitar el uso de los motores, la segunda en realizar una onversión de niveles en la lectura de los infrarrojos. A este bloque también pertenecen los arrays de resistencias de 220 ohmios y 47 Kohmios que se usan para polarizar los infrarrojos. El bloque E esta formado por un array de ocho resistencias de valor 4K7. Sirve de pull-up para las entradas digitales. Más adelante se explica esto con más detalle.

2. Descripción de los elementos de la CT293+:

- C1,C2 : Condensadores en paralelo con los motores. Sirven para eliminar ruido. El valor del condensador dependerá del motor utilizado. No poner condensadores electrolíticos
- C3,C4 : Condensadores de eliminación de ruido para las pastillas integradas.
- S1 : Array de cuatro switches. Acodado. Sirve para anular las entradas de los sensores.
- U2 : Pastilla 40106 c-mos inversora. Adapta niveles de entrada de los sensores.
- U1: driver de potencia L293B 1 para control de motores.
- PUERTO A, E: Conectores tipo bus acodado (5+5 líneas, Macho). Conexión al sistema de control.
- R1: Array de resistencias de 4+1, polarización de infrarrojos.
- R2: Array de resistencias de 4+1, polarización de infrarrojos
- R3: Array de resistencias de 8+1 , pull_up de las entradas digitales.
- J2 : Clema doble para la alimentación motores.
- J4-J8 : Clemas dobles para las entradas digitales.
- motor1, motor2 : Clemas dobles donde se conectan los motores.
- JP1: *Jumper* para conexión interna de los motores.
- sensor1 - sensor4: Conexiones para los sensores de infrarrojos CNY70.

3. DESCRIPCIÓN DE LOS BUSES DE CONTROL.:

Como ya se ha dicho la tarjeta tiene dos bloques independientes, cada uno de ellos se conecta a la CT6811 mediante un cable de expansión. Hay veces en las cuales el usuario desea saber la disposición de los pines del bus ya sea por curiosidad, necesidad o simplemente porque quiera conectar la tarjeta a otro sistema de control.
En este apartado se describen dichos buses. En la pagina 103 del manual de Microbotica se indica cómo construir el cable de unión entre la CT293+ y la CT6811 para que la asignación de pines entre los buses sea correcta. Esto es importante ya que si se construye mal se puede derivar la alimentación a otros pines diferentes pudiendo incluso dañar algún elemento de la CT293+. La muesca que aparece en el conector hembra del bus coincide con la que hay en el conector macho.
Al construir el cable de bus hay que situar los conectores hembra con las muescas apuntando en la misma dirección, (viendo el cable verticalmente). Si se ve en horizontal que es el caso de la figura, se observa que las muescas se sitúan inversamente, si una mira hacia arriba la otra lo hace hacia abajo.

PUERTO A (BUS A)	DESCRIPCION
PA0	entrada digital/analógica 1
PA1	entrada digital/analógica
PA2	entrada digital/analógica 3
PA3	entrada digital/analógica 4
PA4	entrada digital/analógica 5
PA5	entrada digital/analógica 6
PA6	sentido de giro Motor 2
PA7	estado sensor 3
GND	Masa de la CT293+
VCC	Alimentación TTL (5v)
PE6	entrada digital/analógica 7
PE7	entrada digital/analógica 8
VRL	nivel bajo del conversor.
VRH	nivel alto del conversor.

A la izquierda de la tarjeta CT293+ hay unos contactos metálicos agrupados de tres en tres y con un marco que dice sensor X. Estos contactos no forman parte del bus de control o expansión, su misión es la de servir de conector para enganchar los sensores de infrarrojos. En el capítulo 3 se explica como realizar esta unión, ahora se indica su significado.

13. INSTALACIÓN DE LOS SENSORES Y MOTORES

La CT293+ puede incorporar bastantes tipos de sensores, también puede manejar motores paso a paso y motores de continua. En este capítulo se pretenden dar algunos consejos y guías prácticas para evitar conexiones erróneas.

1. CONEXIÓN DE LOS MOTORES DE CONTINUA A LA CT293+:

   La tarjeta esta preparada para poder controlar dos motores de continua simultáneamente. Se realiza por medio del circuito integrado L293B que contiene dos circuitos internos con el ‘puente en H’ necesario para controlar motores. Es un integrado que se adapta perfectamente ofreciendo una etapa de control y de potencia en un mínimo espacio. La conexión de los motores a la tarjeta se realiza a través de las clemas dobles llamadas ‘motor_1’ y ‘motor 2’. Estas clemas están situadas en el centro de la parte inferior de la tarjeta,. Los motores se pueden situar lejos de la placa, aunque es aconsejable que el cable de unión no supere los 20 cm, por cuestiones de ruido eléctrico, y pérdida de potencia. Generalmente los motores tienen un sentido de giro preferente. Al girar en ese sentido las escobillas resbalan sobre el colector produciendo un desgaste pequeño. Los fabricantes de motores suelen indicar este sentido colocando un signo positivo en alguno de los conectores del motor. Si se introduce la tensión de acuerdo con ese criterio el giro obtenido será el correcto. Esto no quiere decir que el motor se vaya a romper si se introduce mal , lo único que refleja es que si la aplicación va a mover el motor en un solo sentido conviene que este coincida con el preferente para aumentar la vida útil del mismo. Cuando el motor se va a utilizar en aplicaciones que requieran ambos giros, lo anterior no afectará mucho, el desgaste de las escobillas se producirá sobre todo por los cambios de sentido y no por el rozamiento inverso que será despreciable. Se va a utilizar esta característica para definir una forma normalizada de conectar los motores, de forma que cualquier microbot construido siguiendo este patrón será compatible a nivel de software. Siguiendo la forma de conexión anterior y mirando los motores de frente se puede sacar esta tabla. Cuando el lector construya su microbot puede optar por colocar los motores al azar y luego construirse una tabla como esta o intentar situar los motores siguiendo lo anterior para obtener una tabla igual. Cualquiera de las dos opciones sirve aunque se aconseja la segunda para garantizar una compatibilidad.

BIT DE DIRECCION	BIT DE DIRECCION	MOTOR 2	MOTOR 1

Bit 6 ON (1) DERECHA OFF (0) - IZQUIERDA	Bit 5 ON (1) - DERECHA OFF (0) – IZQUIERDA	Bit 4 ON (1) - motor ON OFF (0) - motor OFF	Bit 3 ON (1) - motor ON OFF (0) - motor OFF

Tabla de control de los motores

Se explica ahora la función del *jumper* JP1. Este es el encargado de seleccionar entre la alimentación interna o externa de los motores. Cuando se conecta el *jumpe*r, la tensión TTL (+5v) se conecta con la entrada de alimentación de los motores. La ventaja es que con una sola fuente de tensión se puede dar energía a todo el sistema. El inconveniente esta en el ruido que los motores introducen en el sistema. Sobretodo aparecen caídas bruscas de tensión que pueden desprogramar la EEPROM interna del 68HC11. Para evitar esto esta el *jumper* JP6 en la CT6811, este *jumper* protege a la eeprom haciendo un *reset* de la placa siempre que la tensión de alimentación este por debajo de 4.5v. Esto supone un compromiso. Si se protege la eeprom el sistema puede desconectarse automáticamente en las caídas bruscas de tensión, si se quita el *jumper* JP6 de la CT6811 no se produce esa desconexión (salvo caídas muy grandes <2.5) pero se corre el riesgo de que se desprogrme la eeprom. (En los microbots del Grupo J&J el LVI (JP6) suele estar desconectado y aún así rara vez se ha desprogramado la eeprom).
Cuando el *jumper* JP1 de la CT293+ esta quitado significa que se ha seleccionado la alimentación de motores externa. En este caso para que los motores se muevan se tiene que introducir la alimentación por la clema situada junto a las clemas de los motores.. La desventaja es la necesidad de utilizar dos fuentes de alimentación, mientras que la ventaja es el poder poner más tensión a los motores. Por ejemplo se puede utilizar 12v, 9v, 1v, ... sin interferir con la alimentación TTL. Otra ventaja es que no se produce tanto ruido en la línea y el LVI puede ser compatible con los motores. El L293B permite alimentaciones para los motores de hasta +36v y una corriente de salida de 1Amperio. Cuando se use en condiciones extremas probablemente sea necesario ponerle un disipador.

2. CONEXIÓN DE LOS SENSORES DE INFRARROJOS A LA CT293+:

Otro de los elementos que maneja la CT293+ son los infrarrojos CNY70. En concreto esta tarjeta tiene capacidad para controlar directamente cuatro. Si se recurre a las entradas digitales extras se puede llegar a controlar hasta un número de doce. Cada CNY70 incorpora un emisor y un receptor en el mismo encapsulado.
Estos infrarrojos son de corta distancia, son capaces de distinguir el negro de otro color mientras la distancia del infrarrojo a la superficie no supere unos pocos centímetros. Lo más normal es situarlos a medio centímetro del suelo para evitar interferencias con otras fuentes de luz, ya sea solar o artificial.

A la izquierda de la tarjeta CT293+ están situados cuatro conectores acodados de tres pines donde se deben conectar los sensores de infrarrojos. Para realizar dicha conexión, tan solo se tiene que lanzar un cable desde la pata X del sensor hasta el pin X del conector, teniendo en cuenta que las patas C y A se deberán cortocircuitar.

14. PROGRAMACIÓN DE LA TARJETA CT293+:

La tarjeta esta pensada para conectarse a la tarjeta CT6811. Aunque se puede conectar a otros tipos de controladores. El rendimiento máximo de la tarjeta se saca cuando se conecta al puerto A y al puerto E de la CT6811. El control de la CT293+ se realiza a través de dos bytes. Un byte controla los motores y ensores de infrarrojos y el otro las entradas digitales/analógicas.
Se divide este capítulo en dos partes, en la primera se explica como controlar el bloque A de la CT293+ (Motores e infrarrojos) y en la segunda se explica como usar el bloque E (entradas digitales/analógicas).

1. PROGRAMACIÓN DEL BLOQUE A. MOTORES Y SENSORES DE INFRARROJOS: Lo primero es unir el Bloque A de la CT293+ con la CT6811 utilizando el bus A., para ello conectar un cable de bus entre el Puerto A de la CT293+ (conector J1) y el Puerto A de la CT6811 (conector J1). aquellas personas que no poseen la CT6811 tienen que conectar el Puerto A de la CT293+ con el puerto correspondiente en su sistema de control.
Automáticamente al colocar dicho cable se proporcionará la alimentación TTL a la CT293+. Ahora hay que elegir una de las dos opciones siguientes. La primera consiste en utilizar esa alimentación TTL para alimentar los motores, para ello hay que colocar en su sitio el *jumper* JP1 de la CT293+. La segunda consiste en alimentar los motores con otra fuente de alimentación, para ello hay que desconectar el *jumper* JP1 e introducir la alimentación externa por la clema J2. Tener cuidado con la polaridad.
Una vez realizado lo anterior el control de la C DirecciónT293+ es muy sencillo. Tan solo se utiliza un byte de control, que se corresponde con el Puerto A de la CT6811, posición de memoria $1000 del *microcontrolador* 68hc11. El

significado de los bits de dicho byte se detalla a continuación.

2.Los Sensores De Infrarrojos:

Los bits en blanco significan que son de entrada, y proporcionan el estado de los infrarrojos. Para que la lectura de estos bits sea correcta es necesario activar las **Al utilizar la alimentación externa tener mucho cuidado con no olvidar desconectar el jumper JP1 y con la polaridad de la tensión de alimentación de los motores. Si se utiliza alimentación única antes de colocar el jumper JP1 desconectar la alimentación externa de los motores (clema J2).**
salidas de los sensores de infrarrojo correspondientes. Eso se hace con los interruptores que hay en la placa. La correspondencia entre la localización física del infrarrojo y su interruptor. Correspondencia entre los sensores y el byte de control
Los interruptores (switches) que se han puesto en la placa tienen la utilidad de anular y dejar libres los bits que emplean los infrarrojos. Si el sistema sólo va a manejar motores se pueden utilizar los bits restantes como entradas sin entrar en conflicto con los sensores. Esto es importante cuando se trabaja con el Puerto A de la CT6811 ya que esos bits son muy útiles y no conviene malgastarlos. Como norma básica se aconseja que los interruptores estén a ON cuando se vayan a manejar los sensores, en caso ontrario ponerlos a OFF. En la placa existen dos tipos de microinterruptores, el tipo A se caracteriza porque para cerrar el circuito (poner ON) hay que situar la palanca hacia abajo. El tipo B se caracteriza por lo contrario, para poner ON hay que situar la palanca hacia arriba. Se propone un ejercicio. Realizar un programa que lea el sensor 4, si esta detectando negro que no encienda el LED de la CT6811 y si esta detectando blanco que lo encienda. Para realizar este programa se activará el switch S1, ya que corresponde al sensor4. Luego se procederá a crear el software. Hay que leer el byte $1000 del 68HC11, es decir el Puerto A. De este byte se ignorarán todos los bits menos el bit 2 (PA2), correspondiente al sensor 4. Según el valor leído se actuará en consecuencia con lo propuesto.
El programa no esta optimizado, se propone al lector que lo simplifique. Como orientación se indica que puede pasar de ocupar 22 bytes (el de arriba) a ocupar 16 bytes. Se recomienda mirar el juego de instrucciones y reordenar el código. De todas formas en el apartado 4.1.2. se indica la solución. El código del programa se muestra en la página siguiente.

Sensor 1 >> switch 4 >> BIT 0 (PA0)
Sensor 2 >> switch 3 >> BIT 1 (PA1)
Sensor 3 >> switch 2 >> BIT 7 (PA7)
Sensor 4 >> switch 1 >> BIT 2 (PA2)

La **interpretación** del bit de un sensor es la siguiente:

Bit a **1** significa detectado **Negro**

Bit a **0** significa detectado **Blanco (No negro)**

3. Los Motores De Continua.
Al bloque A pertenece la gestión de los motores. Los bits sombreados (del byte de control) son salidas y se encargan del funcionamiento de los motores. Los bits marcados con Dirección seleccionan el sentido e giro. Los bits marcados con ON/OFF indican si se conecta o no el motor.
El control de los motores se realiza con un circuito que tiene como parte principal el chip L293B que es capaz de suministrar hasta un Amperio a cada uno de los motores. Conectando los motores como se dijo en el capítulo tres y mirando cada motor de frente se obtiene la tabla de control de abajo. Se recomienda que cada uno conecte los motores como quiera y que luego se construya una tabla como la siguiente. Otra opción es probar diferentes conexiones hasta encontrar una similar a la propuesta aquí. Esto no es complicado, pensar que todo consiste en invertir los cables que unen el motor con la placa.

Figura 1.

Switch 2 Sensor 3	Motor 2 Dirección	Motor 1 Dirección	Dirección Motor 2 ON/OFF	Dirección Motor 1 ON/OFF	Switch 1 Sensor 4	Switch 3 Sensor 2	Switch 4 Sensor 4

Bit 7 in Bit 6 out Bit 5 out Bit 4 out Bit3 out Bit 2 out Bit 1 out Bit 0 in

RESOLUCIÓN DEL LABERINTO.

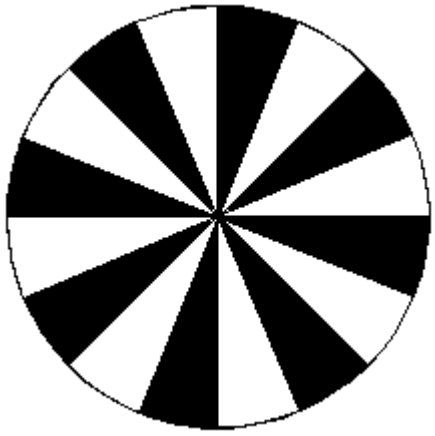
La solución propuesta trata de explorar una ruta almacenada previamente en memoria, el seguimiento de esta ruta esta determinado por un ENCORDER situado en la rueda derecha y seleccionada por unos microswitch previamente, activando la ruta adecuada dependiendo del punto de partida.

Objetivos: Recorrer el laberinto en el menor tiempo posible de forma optima.

Material utilizado:

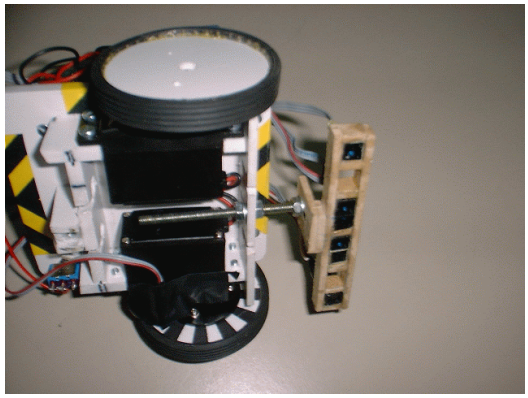
- Un disco de 8 franjas negro (16 veces seccionado por estas 8 franjas) diseñado con una herramienta CAD e impreso por una impresora laser.
- Un sensor infrarrojo CNY70.
- Microcontrolador MC68HC11
- Tarjeta de desarrollo CT6811
- Tarjeta CT293+
- 2 Motores
- Estructura Base Modular.

Nota: Este material utilizado, es usado en otra prueba del concurso cambiando la disposición de ciertas conexiones.



Disco del Encoder

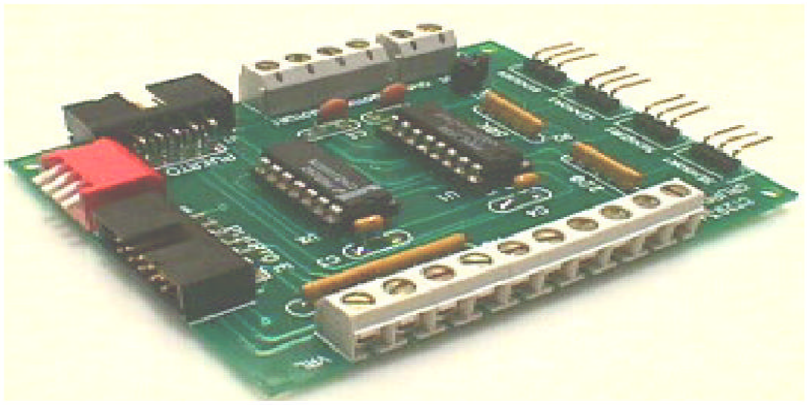
Disposición del Encoder



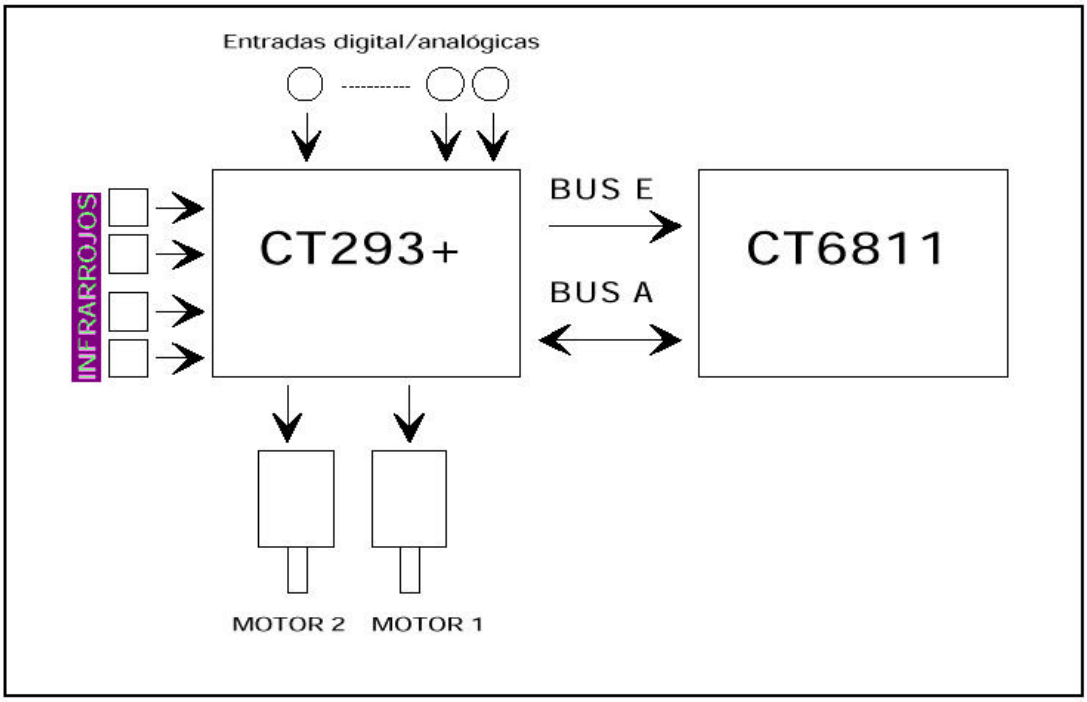
Descripción y disposición física.

La CT293+ es una nueva tarjeta que proporciona al sistema Tower la posibilidad de controlar motores y sensores. Esta diseñada para adaptarse perfectamente a la CT6811 y poder controlarla sin ninguna variación tanto en Bootstrap como en Single
Chip. La tarjeta también puede controlarse desde otro sistema por ejemplo el puerto paralelo de un PC.

La CT293+ es el soporte principal de los dos primeros niveles de la torre BOT, con ella se proporciona el movimiento a los sistemas de control y la capacidad de analizar el entorno. Con esta placa y con la CT6811 se puede obtener la plataforma de un microbot.
Esta tarjeta se acopla perfectamente con la CT6811 proporcionando un medio de E/S con dispositivos analógicos y dígitales.



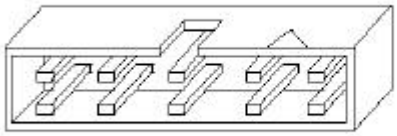
CT-293 1



CT293 acoplada CT6811 1

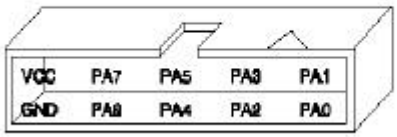
No nos pararemos a analizar en detalle toda la estructura de la placa CT293, solo describiremos las estructuras Hardware de comunicación con la CT6811, por la relación

intrínseca con el PUERTO A y la programación de este.

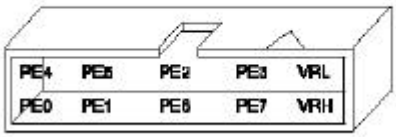


Conector macho del bus de conexión

Posición correcta de los conectores hembras en los buses de conexión



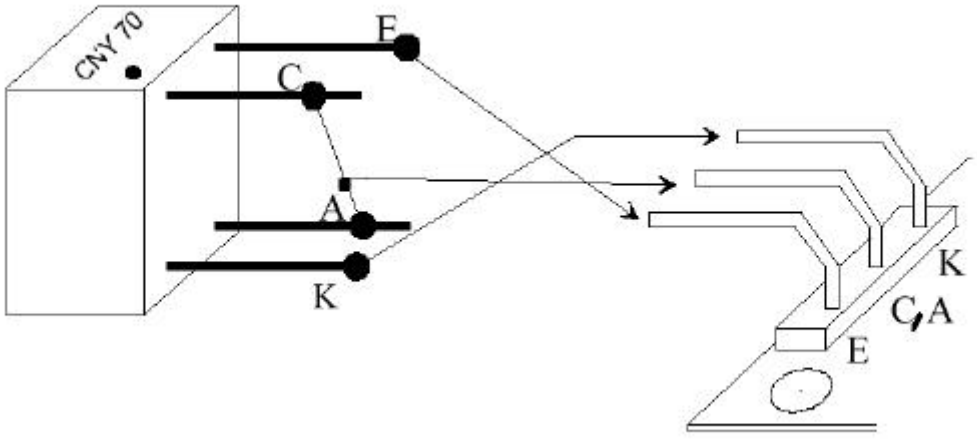
Configuración PUERTO A



Configuración PUERTO E

El sensor de infrarrojos esta colocado a menos de 2mm del disco y la soldadura física es la siguiente:

Esta tarjeta tiene capacidad para controlar directamente cuatro. Si se recurre a las entradas digitales extras se puede llegar a controlar hasta un número de doce. Cada CNY70 incorpora un emisor y un receptor en el mismo encapsulado. Estos infrarrojos son de corta distancia, son capaces de distinguir el negro de otro color mientras la distancia del infrarrojo a la superficie no supere unos pocos centímetros. Lo más normal es situarlos a medio centímetro del suelo para evitar interferencias con otras fuentes de luz, ya sea solar o artificial.

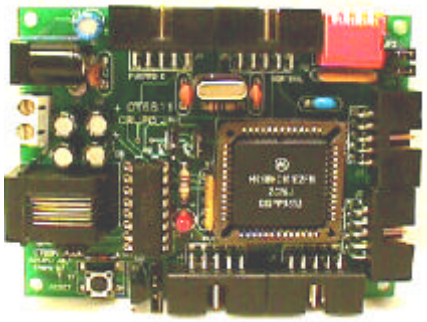


Conexión física del sensor CNY70

A la izquierda de la tarjeta CT293+ están situados cuatro conectores acodados de tres pines donde se deben conectar los sensores de infrarrojos. Para realizar dicha conexión ver la figura, tan solo se tiene que lanzar un cable desde la pata X del sensor hasta el pin X del conector, teniendo en cuenta que las patas C y A se deberán cortocircuitar

Disposición del sensor frente al Disco Encoder.

LA CT6811, DESCRIPCIÓN GENERAL DE USO



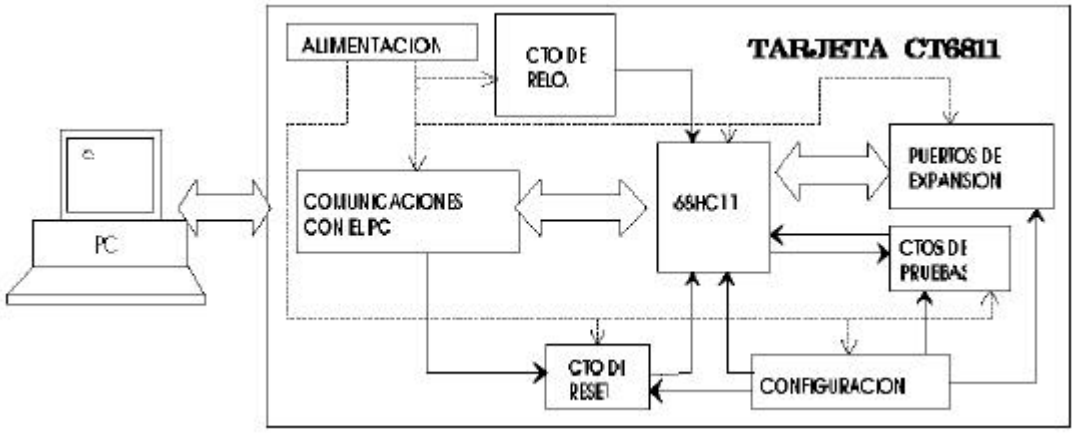
Características de la CT6811.

Microcontrolador 68HC11:
Por ello incorpora todas las características de este micro:

512 Bytes de EEPROM en los modelos A1 y A8
256 Bytes de memoria RAM interna
Temporizador de 16 bits
3 capturadores de entrada
5 comparadores con salida hardware
Un acumulador de pulsos
Comunicaciones serie asíncronas
Comunicaciones serie Síncronas
8 canales de conversión analógico digital
Interrupciones en tiempo real
4 puertos de E/S

Bus Expansión: Bus de expansión dividido en 6 puertos de 10 bits cada uno.
Desde estos puertos se tiene acceso a todas las patas del micro.
Switches de configuración: 2 switches de configuración del modo de arranque de la placa (bootstrap, single chip, expanded, special test) y 2 switches disponibles para aplicaciones del usuario
Led de pruebas conectado al bit 6 del puerto A. Este led se puede conectar y desconectar por medio de un jumper
Pulsador de reset/pruebas IRQ: Pulsador para inicializar la tarjeta. Este pulsador se puede utilizar también, cambiando un jumper, como un pulsador de propósito general conectado a la entrada de interrupciones IRQ.
Niveles VRH y VRL configurables a VCC y masa respectivamente mediante 2 jumpers
Modos de funcionamiento entrenador / autónomo configurables mediante un jumper
Alimentación a través de clemas o conector tipo jack
Conexión directa al PC por medio de un cable tipo teléfono
Reset software y hardware de la placa

DIAGRAMA DE BLOQUES DE LA TARJETA CT6811

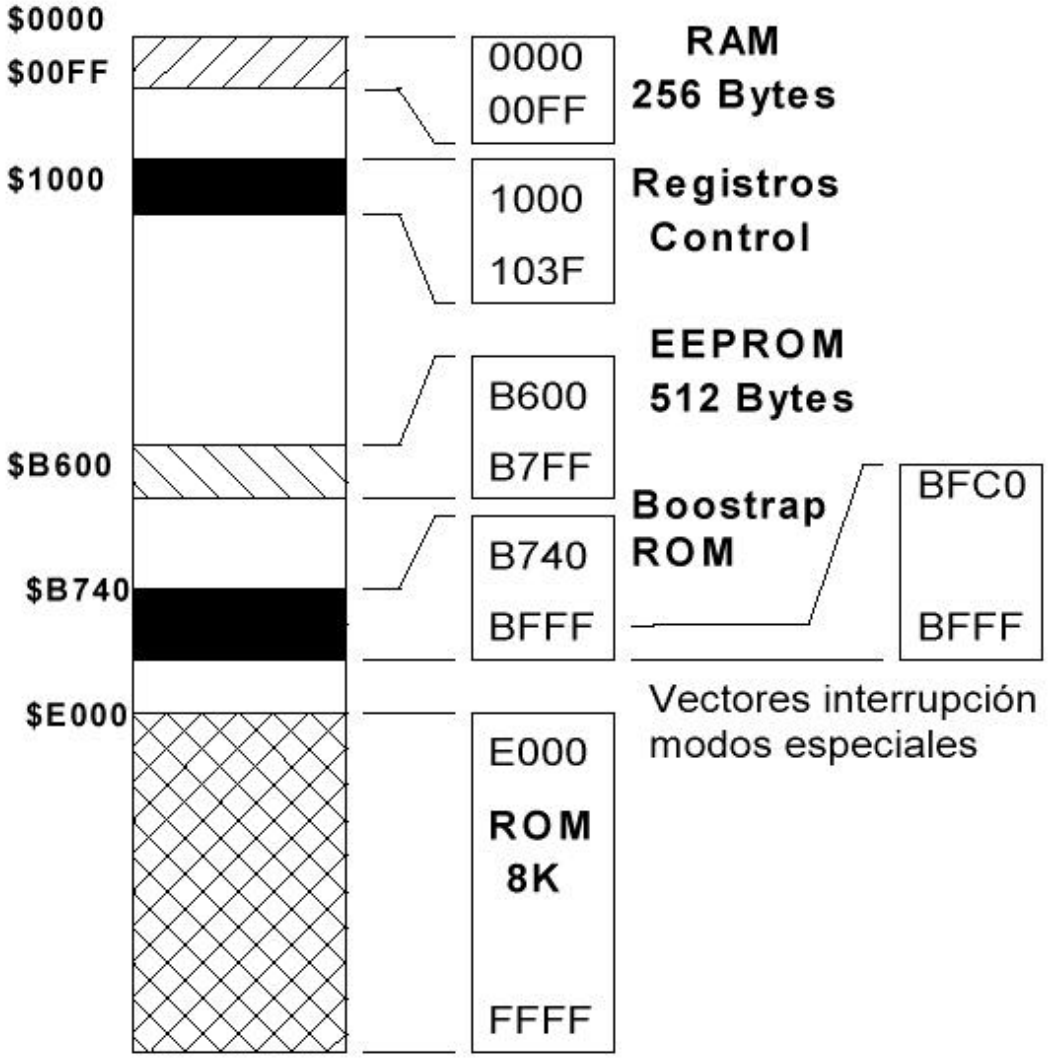


Nuestro programa esta albergado en un entorno de memoria reducida. Solo hay 512 bytes para poder realizar todas las funciones.

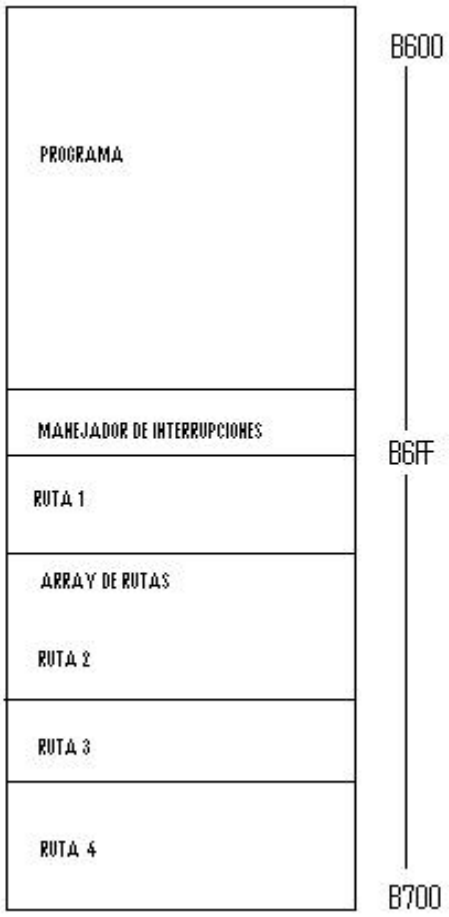
MAPA DE MEMORIA

El MCU direcciona 64Kbytes de memoria, parte de esta memoria se encuentra dentro del MCU y el resto se puede implementar mediante chips de memoria externos al MCU. Según el modelo de microcontrolador empleado (A8, A0, A2, E9...) se dispondrá de más o menos recursos de memoria. La memoria RAM se sitúa por defecto a partir de la dirección $0000 hasta la $00FF (256 bytes de RAM).
De la dirección $1000 hasta la $103F se encuentran situados los registros de control del MCU. Estos registros son células de memoria RAM o EEPROM (no volátiles). Las células EEPROM sólo se pueden escribir bajo unas circunstancias especiales, y siempre que el MCU esté en modo especial.
Entre las direcciones $B600-$B7FF se sitúan 512 bytes de memoria EEPROM que no están disponibles en el modelo A0. En los modos especiales se activa una memoria ROM en las direcciones $BF40-$BFFF que contiene el programa BOOTSTRAP usado para cargar programas externos y una tabla con los vectores de interrupción.
Entre las direcciones $E000-$FFFF se encuentra la memoria ROM, que salvo ocasiones muy especiales siempre está desconectada. En esta ROM el fabricante puede grabar cualquier aplicación que el usuario quiera (y que pueda pagar!!).
Desde la dirección $FFC0 hasta el final se encuentra la tabla de vectores de interrupción del modo normal.

En la figura se muestra el mapa de memoria por defecto. Configurando algunos registros de control adecuadamente se puede "remapear" el sistema para ajustarlo a las necesidades de la aplicación. Por ejemplo, la memoria RAM se puede situar en cualquier otra parte (con alguna restricción) dentro de los 64Kbytes del mapa de memoria. Lo mismo ocurre con los registros de control. El registro que permite cambiar las direcciones de la memoria RAM y de los registros de control se denomina INIT y se encuentra en la dirección $103D.
Las direcciones de memoria están formadas por 2 bytes, es decir, direcciones de 16 bits. Los bits del 7 al 4 del registro INIT permiten establecer los 4 bits de mayor peso de la dirección en la que se va a mapear la memoria RAM. Los 12 bits restantes no se pueden fijar. Los bits del 3 al 0 hacen lo mismo pero con la dirección de comienzo de los registros de control. El mapa de memoria queda dividido en 16 páginas de 4KBytes cada página (16*4KB = 64Kb). Tanto la RAM interna como los registros de control se pueden situar en cualquiera de estas 16 páginas:
$0000 (Situación de la RAM por defecto), $1000 (Situación de los registros de control por defecto), $2000, $3000, $4000, $5000 ... ... $D000, $E000 y $F000.
Debido a la posibilidad de remapear, pueden surgir conflictos al encontrarse 2 recursos internos o externos en las mismas posiciones de memoria. El MCU resuelve esto adjudicando unas prioridades. Si se mapean la RAM, los registros de control y un dispositivo externo en la zona de la ROM (A partir de la dirección $E000 en adelante) el MCU asigna la máxima prioridad a los registros de control, después a la RAM, después la ROM y finalmente el recurso exterior. En este caso, si se intentase acceder a las direcciones $E000-$E03F se seleccionarían los registros de control. A partir de la $E03F ya no existirían los registros de control y se activaría la memoria RAM. A partir de la dirección en la que la RAM se acabase (sólo son 256 bytes de RAM) se accedería a la ROM. Al dispositivo exterior nunca se podría acceder a no ser que se mapease fuera de la zona destinada a la ROM o que se desactivase la ROM.


$0000
$00FF
$1000
$B600
$B740
$E000
0000
00FF
RAM
256 Bytes
1000
103F
Registros
Control
B600
B7FF
EEPROM
512 Bytes
B740
BFFF
Boostrap
ROM
BFC0
BFFF
Vectores interrupción
modos especiales
E000
ROM
8K
FFFF

DISPOSICIÓN DE NUESTRO PROGRAMA EN LA MEMORIA EEPROM



1

[1] El Encoder usa la ACTIVACIÓN por flanco cuando el sensor detecta la franja negra. Entonces se genera una INTERRUPCIÓN que es tratada por las rutinas que manejan esa Interrupción.

PUERTOS QUE INFLUYEN EN EL PROGRAMA

En general usamos el puerto A para la comunicación con el sensor/es del ENCODER y los motores.

EL PUERTO A

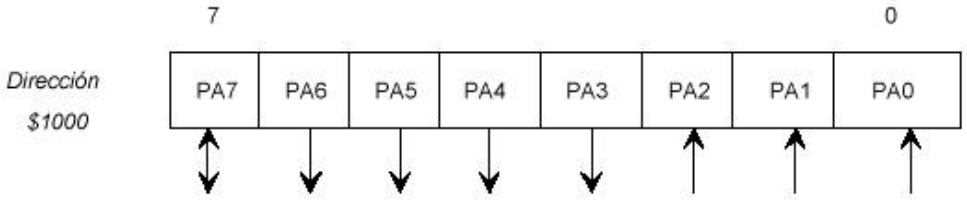


Figura 15: Bits del puerto A

El puerto A dispone de 3 pines de entrada, 4 pines de salida y uno configurable como entrada o salida. Se encuentra mapeado en memoria en la dirección $1000. Los pines del puerto A están compartidos por otros recursos: comparadores, acumulador de pulsos y capturadores. En la tabla se muestran todas las funciones que están asignadas a cada pin del puerto A.

	7							0
Direccion $1000	PA7 PAI OC1	PA6 OC2 OC1	PA5 OC3 OC1	PA4 OC4 OC1	PA3 OC5 OC1	PA2 IC1	PA1 IC2	PA0 IC3

PA0-PA2	Bits de entrada
PA3-PA6	Bits de salida
PA7	Bidireccional
OCx	Salidas de los comparadores
ICx	Capturadores de entrada
PAI	Acumulador de oulsos

Esquema de configuración en la conexión entre el PUERTOA y la CT293

Switch 2 Sensor 3	Motor 2 Dirección	Motor 1 Dirección	Motor 2 ON/OFF	Motor 1 ON / OFF	Switch 1 sensor 4	Switch 3 sensor 2	Switch 4 sensor 1	PuertoA $1000
Bit 7-in	Bit 6-out	Bit 5-out	Bit 4-out	Bit 3-out	Bit 2-in	Bit 1-in	Bit 0- in	

Usamos el BIT0 y los 4 bits marcados de los motores para el control de dirección. La activación del sensor es tratada por una rutina almacenada en el código de segmento del programa. Cuando se activa una interrupción por FLANCO de SUBIDA se llama a la rutina de tratamiento correspondiente.

(Posteriormente para una mayor precisión usaremos 2 encoders si nos da tiempo a implementar las rutinas, de momento oficialmente presentamos 1 encoder).

CONTROL DE MOTORES POR EL PUERTO A

	MOTOR 1	MOTOR 2
DIRECCIÓN	BIT 5 ON (1) - DERECHA OFF (0) - IZQUIERDA	BIT 6 ON (1) - IZQUIERDA OFF (0) - DERECHA
ESTADO	BIT 3 ON (1) - motor ON OFF (0) - motor OFF	BIT 4 ON (1) - motor ON OFF (0) - motor OFF

Control de pulsos por interrupción del sensor asociado al encoder.

Para la conexión, lo único que debemos tener en cuenta es que los bits 0, 1 y 2 del puerto A del microcontrolador MC68HC11 están asociados con tres capturadores de entrada, que nos podrán ser de gran utilidad a la hora de capturar las señales que llegan de los sensores. El bit 7 del puerto A también está asociado con otro recurso interno del microcontrolador: el acumulador de pulsos.

Este recurso es muy interesante para aplicaciones como la que estamos considerando. Como un acumulador de pulsos tiene asociado un contador que se va incrementando automáticamente con la llegada de nuevos pulsos, la cuenta de los pasos en el giro de las ruedas se haría de forma casi automática. Desafortunadamente, el MC68HC11 tiene un solo acumulador de pulsos por lo que en cualquier caso deberemos hacer uso de al menos un capturador.

Programación

Como se ha mencionado anteriormente, utilizaremos los capturadores de entrada para recibir los datos de los sensores. Concretamente, se utilizarán los capturadores de entrada IC3 e IC1, asociados con los bits 0 y 2 del puerto A, respectivamente. Como el código necesario es similar para uno y otro capturador, sólo comentaremos el primer caso: el capturador IC3 (bit 0 del puerto A) que se corresponderá con el encoder de la rueda derecha según el montaje propuesto. Un capturador de entrada genera una interrupción cada vez que detecta un flanco en la señal que tiene asociada. El flanco, que es configurable como ahora veremos, puede ser de subida, de bajada o ambos. En nuestro caso, el flanco se producirá siempre que el sensor detecte un cambio de blanco (nivel bajo) a negro (nivel alto) .

La programación puede dividirse, conceptualmente hablando, en tres partes: Configuración del capturador de entrada, manejo de la interrupción y uso. Lo iremos viendo detenidamente.

REGISTROS DE CONTROL, CAPTURADOR.

7	6	5	4	3	2	1	0
OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
X	X	X	X	X	X	X	1

Registro TMSK1 ($1022)

Esto en ensamblador del MC68HC11 se corresponde con la instrucción:

```
LDX #$1000                ; Apunta a los registros de ctrl
BSET TMSK1,X $01          ; Permite interrupciones del IC3
```

El segundo registro a tener en cuenta es el TCTL2 ($1021). Con los dos bits menos significativos de este registro indicamos qué tipo de flanco queremos utilizar. Los otros bits están asociados con los otros dos capturadores de entrada, quedando otros dos sin uso.

7	6	5	4	3	2	1	0
0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
X	X	X	X	X	X	0	1

Registro TCTL2 ($1021)

Nuevamente en ensamblador del MC68HC11 se corresponde con las instrucciones:

```
LDX #$1000                ; Apunta a los registros de ctrl
BCLR TCTL2,X $02          ; Flanco de subida del IC3
BSET TCTL2,X $01          ; Flanco de subida del IC3
```

Los bits, tomados dos a dos, de este registro pueden tener los valores recogidos en la siguiente tabla:

Flanco	EDGxB	EDGxA
Inhibidos	0	0
Subida	0	1
Bajada	1	0
Ambos	1	1

Entonces, según estas tablas, estamos configurando el capturador de entrada IC3 en flanco de subida, por lo que, resumiendo, cada vez que el microcontrolador detecte un flanco de subida en su pin PA0 proveniente de un cambio provocado al pasar de un segmento blanco a otro negro y detectado por el sensor, se generará la interrupción asociada con el capturador de entarda IC3.

A continuación veremos cómo capturar esta interrupción y qué hacer cuando se produce.

MANEJO DE LA INTERRUPCIÓN

Una de las cosas que tenemos que hacer al estar trabajando con interrupciones es indicar la micro que éstas están permitidas. Esto se hace mediante la instrucción CLI en la parte de inicialización de nuestro programa.

Lo que nos queda, ya para el caso que estamos tratando, es reservar espacio para una variable en la que llevaremos la cuenta de los pasos que ha dado el motor, escribir la rutina de tratamiento de la interrupción y asociar dicha rutina con la interrupción.

Definimos la variable pasos_d de 16 bits con lo que podemos contar hasta 65535 pasos, si son necesarios más, puede manejarse el desbordamiento y utilizar una segunda varible para ampliar el contador.

La rutina de tratamiento de la interrupción en este caso es bastante sencilla. Cada vez que se produzca una interrupción incrementaremos la variable pasos_d, borraremos el flag que indica que no se ha tratado la interrupción del IC3 e finalizaremos con una instrucción de retorno de interrupción.

El flag de la interrupción del IC3 está se encuentra en el registro TFLG1 junto con los flags de los otros capturadores. El del IC3 ocupa el bit menos significativo. Estos bits se activan a nivel bajo (se pone a '0') cada vez que se produce una interrupción. Una vez tratada, debemos borrar el flag (escribir un '1') desde la rutina de tratamiento para indicar que la interrupción ha sido tratada, en caso contrario se anidarían las interrupciones.

```
h_enc_d  PSHY
         LDY #$B677              ;Inicializamos Indexador
         LDAA pasos_d,Y          ; Carga el número de pasos
         INCA                    ;incrementamos contador
         STAA pasos_d,Y          ; Almacena el número de pasos
         PULY                    ; Desapilamos Y
         LDAA #$01               ; Borra el flag de interrupción IC3
         STAA TFLG1,X
         RTI                     ; Retorno de interrupción
```

Por último, nos queda indicar dónde se encuentra el manejador de la interrupción. El vector de dirección es dependiente del modo en el que esté operando el microcontrolador. Para los modos en los que tengamos los vectores de interrupción en la RAM interna (sin ampliaciones de memoria), esto puede indicarse de la siguiente forma:

```
ORG $B6E2
; Instala el manejador

 JMP h_enc_d
```

EL PROGRAMA

```
PORTA EQU $0                        ; Puerto A
TCTL2 EQU $21                       ; Activaciones de flancos ICn
TMSK1 EQU $22                       ; Interrupciones de capturadores
TFLG1 EQU $23                       ; Flags de capturadores
pasos_d EQU $00                      ; En la B6FF

ORG $B600                           ; Programa en ROM

      BRA inicio

;Rubrutina de inicialiación
init  BCLR TCTL2,X $02              ; Flanco de subida del IC3
      BSET TCTL2,X $01              ; Flanco de subida del IC3
      BSET TMSK1,X $01              ; Permite interrupciones del IC3
      CLI                           ; Permite interrupciones
      RTS


inicio
        LDX #$1000                  ; Apunta a los registros de ctrl
        LDY #$B700                  ; Apunta al MAPA_ARRAY
                                    ; Mapa del laberinto EJEMPLO
        LDAA #$20                   ; 20 PASOS DELANTE
        STAA $00,Y

        LDAA #$FF                   ; GIRA DERECHA 90 º
        STAA $01,Y

        LDAA #10                    ; 10 PASOS DELANTE
        STAA $02,Y

        LDAA #$FE                   ; GIRA IZQ 90 º
        STAA $03,Y

        LDAA #$FE                   ; GIRA IZQ 90 º
        STAA $04,Y

        LDAA #10                    ; 10 PASOS DELANTE
        STAA $05,Y

        LDAA #$00                   ; PARAR
        STAA $06,Y


        BSR init                    ; Inicializa interrupciones
        LDY #$B6FF
tratar  PSHY
        LDY #$B6FF
        LDAA #$00
        STAA pasos_d,Y              ;Inicializamos la var pasos_d
        PULY
        INY                         ;Incrementamos a la sig. posición
        LDAA $00,Y
        CMPA #$00
        BEQ stop                    ; Si es 0, paramos
        CMPA #$FF
        BEQ girader                 ; Si es FF 90º DERECHA
        CMPA #$FE
```

```
        BEQ giraizq                 ; Si es FE 90º IZQUIERDA
        BLS avanza




Avanza  LDAA #$18                   ; Avanza
        STAA PORTA,X

bucle   PSHY
        LDY #$B6FF
        LDAA pasos_d,Y              ;Carga los pasos de la rueda
derecha
        PULY
        CMPA $00,Y                  ; Lo compara con 8
        BNE bucle                   ; Si no se ha lledado, continua
        CLRA                        ; Detiene el robot
        STAA PORTA,X
        BRA tratar

girader LDAA #$58                   ; Avanza
        STAA PORTA,X

bucle1  PSHY                        ; Apilamos indexador (Ptro)
        LDY #$B6FF                  ;Inicializamos Indexador
        LDAA pasos_d,Y              ; Carga los pasos de la rueda D
        CMPA #6                     ; Lo compara con 6 (precision)
        PULY                        ; Desapilamos
        BNE bucle1                  ; Si no se ha llegado, continua
        CLRA                        ; Detiene el robot
        STAA PORTA,X
        BRA tratar

giraizq LDAA #$38                   ; Gira a la izq
        STAA PORTA,X

bucle2  PSHY                        ; Apilamos (Ptro a Array)
        LDY #$B6FF                  ; Inicializamos Indexador
        LDAA pasos_d,Y              ; Carga los pasos de la rueda I
        CMPA #6                     ; Lo compara con 6
        PULY                        ; Desapilamos indexador
        BNE bucle2                  ; Si no se ha llegado continua.
       CLRA                         ; Detiene el robot
        STAA PORTA,X
        BRA tratar


stop    CLRA
        STAA PORTA,X
        BRA stop                    ; Parar




h_enc_d PSHY
        LDY #$B677                  ; Inicializamos Indexador
        LDAA pasos_d,Y              ; Carga el número de pasos
        INCA                        ; incrementamos contador
        STAA pasos_d,Y              ; Almacena el número de pasos
```

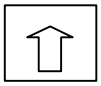
```
PULY                        ; Desapilamos Y
LDAA #$01                   ; Borra el flag de interrupciónIC3
STAA TFLG1,X
RTI                         ; Retorno de interrupción


ORG $B6E2                   ; Instala el manejador
JMP h_enc_d


END
```

DEFINICION DE TERMINOS

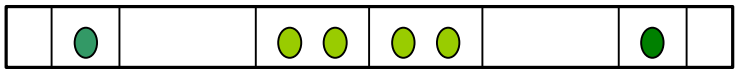
Para evitar confusiones con términos relativos (derecha, izquierda, etc.), se considera que el sentido de marcha o avance del robot es:

IZQUIERDA  **DERECHA**

Los sensores están dispuestos en la zona frontal del robot.

DISPOSICION DE LOS SENSORES

El robot dispone en su parte frontal de un total de seis sensores. Los cuatro sensores centrales se utilizan para seguir el camino (dos para seguir el borde izquierdo del camino y otros dos para seguir el borde derecho). Además dispone de dos sensores adicionales situados a izquierda y derecha que permiten distinguir las señalizaciones. A lo largo de este documento los sensores centrales se denominaran “sensores de seguimiento” y los laterales “sensores de indicación”



Sensores de indicación

Sensores de seguimiento

FUNCIONAMIENTO GENERAL

El robot empieza a seguir el camino con los cuatro sensores, si detecta la existencia de una señalización, pasa a seguir la línea con los dos sensores del lado en el que se ha detectado la señalización.

Una vez pasada la bifurcación continua utilizando los cuatro sensores para seguir el camino.

Las señales de los sensores de seguimiento están mapeadas sobre el puerto A, del microcontrolador, de la siguiente manera (desde una vista superior):

Desde el sensor mas a la izquierda hasta el sensor mas a la derecha: bit 2,1,7,3

Las señales de los sensores de indicación están mapeadas sobre el puerto C, del microcontrolador, de la siguiente manera (desde una vista superior):

Sensor izquierdo bit 7

Sensor derecho bit 6

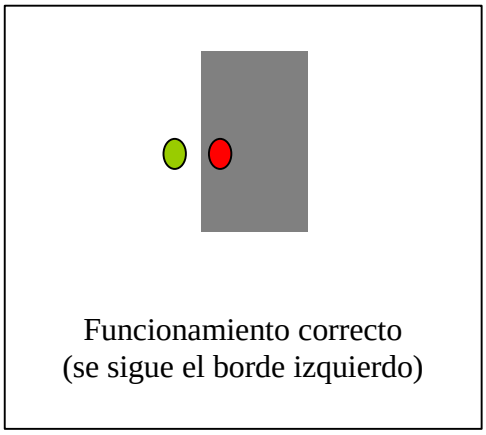
Seguimiento de la línea con dos sensores (modo de seguimiento con dos sensores)

Los sensores se sitúan de forma que uno quede fuera de la línea y otro dentro (uno detectara “claridad” y el otro “oscuridad”, mandando 0 y 1 respectivamente).

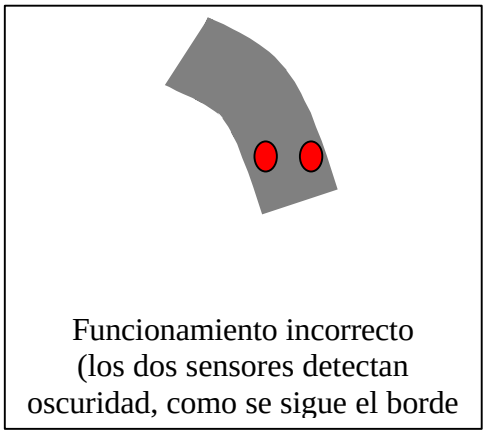
Si el robot deja de seguir el borde los sensores detectaran “claridad” “claridad” o “oscuridad” “oscuridad”.

En el primer caso los dos sensores se han salido del camino, es decir si seguían el borde izquierdo, el robot debe girar a la derecha. Si seguían el borde derecho deben girar a la izquierda.

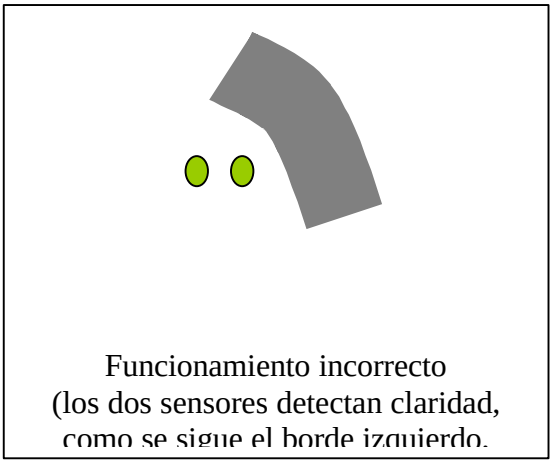
En el segundo caso, los dos sensores están dentro del camino, por tanto si seguían el borde izquierdo, el robot debe girar a la izquierda y a la inversa.



Funcionamiento correcto
(se sigue el borde izquierdo)



Funcionamiento incorrecto
(los dos sensores detectan oscuridad, como se sigue el borde



Funcionamiento incorrecto
(los dos sensores detectan claridad, como se sigue el borde izquierdo.

Este modo de seguimiento es muy simple pero es inútil si existen bifurcaciones, ya que siempre se seguirá la bifurcación derecha si los sensores están situados a ese lado o a la inversa.

Seguimiento de la línea con cuatro sensores (modo de seguimiento con cuatro sensores)

Se utilizan los cuatro sensores para seguir el camino, dos para seguir el borde derecho y otros dos para seguir el borde izquierdo.

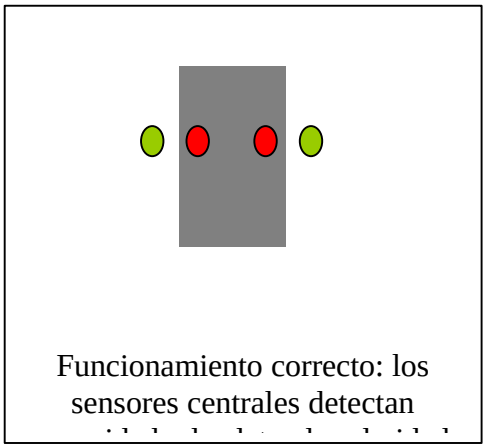
Este modo de funcionamiento da muy buenos resultados, ya que el robot corrige su trayectoria con menos frecuencia y en consecuencia avanza mas rápido.

Si se recibe "claridad" "oscuridad" "oscuridad" "claridad", se esta siguiendo el camino correctamente.

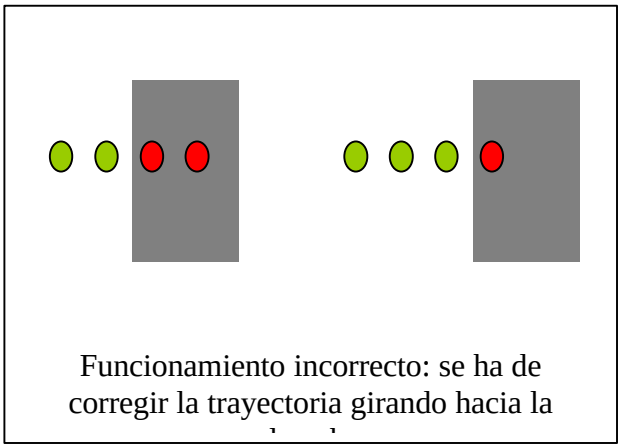
Si se recibe "claridad" "claridad" "oscuridad" "oscuridad" se ha desviado hacia la izquierda, y debe girar hacia la derecha para corregir su trayectoria.

Si se recibe "oscuridad" "oscuridad" "claridad" "claridad" se ha desviado hacia la derecha y debe girar hacia la izquierda.

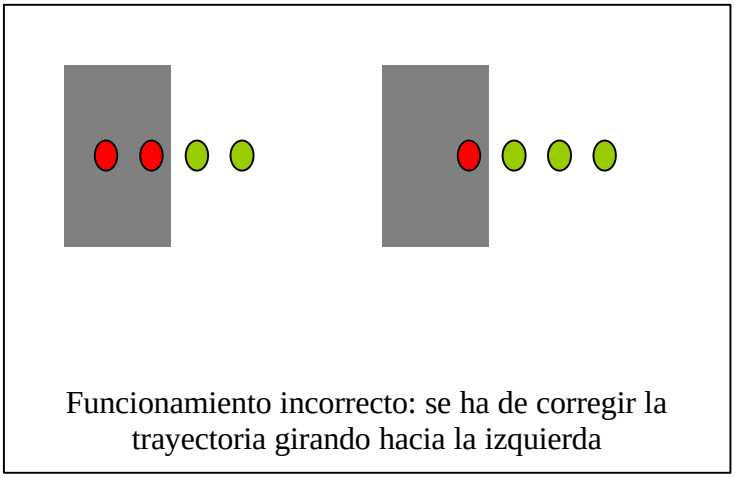
(Hemos tenido en cuenta los casos "claridad" "oscuridad" "oscuridad" "oscuridad" y "oscuridad" "oscuridad" "oscuridad" "claridad", aunque no debería ser necesario por que se sondea el puerto con suficiente frecuencia como para corregir la trayectoria antes de que se llegue a desviar tanto).



Funcionamiento correcto: los sensores centrales detectan



Funcionamiento incorrecto: se ha de corregir la trayectoria girando hacia la



Funcionamiento incorrecto: se ha de corregir la trayectoria girando hacia la izquierda

Detección de bifurcaciones y elección del “camino optimo”

Para detectar las bifurcaciones se dispone de los sensores de indicación mapeados sobre el puerto C.

Cuando el sensor de indicación izquierdo detecta “oscuridad”, el robot pasa al modo de seguimiento con dos sensores, utilizando los sensores situados sobre el borde izquierdo, por lo tanto seguirá la bifurcación de la izquierda.

Cuando el sensor de indicación derecho detecta “oscuridad”, el robot pasa al modo de seguimiento con dos sensores, utilizando los sensores situados sobre el borde derecho, y por tanto siguiendo la bifurcaron de la derecha.

Para volver al modo de seguimiento de cuatro sensores, se tiene en cuenta que el sensor de indicación opuesto al que ha detectado la indicación ha de pasar por encima del camino que no se sigue, es decir detectara oscuridad y en ese momento se pasa a cuatro sensores.

```
        ORG $0000
PORTC   EQU $03
DDRC    EQU $07
PORTA   EQU $00
VAR     EQU $FF
ESTADO  EQU $FE
SW      EQU $FD

      LDX  #$1000
        LDY  #$0000
      LDAA #$07
      STAA DDRC,X
        LDAA #$00
        STAA ESTADO,Y
        STAA SW,Y


        CLRB

inicio
        STAA  PORTA,X
        LDAA  PORTA,X
        STAA  VAR,Y
        CMPB #$01
        BNE der
        BRCLR VAR,Y $87 izquierda
der     CMPB #$02
        BNE sigui
        BRCLR VAR,Y $87 derecha
sigui
        BRCLR VAR,Y $87 derecha

        BRSET PORTC,X $80 ini2auxd
        BRSET PORTC,X $40 ini2auxi

otr
        CMPB #$01
        BNE der1
        BRCLR VAR,Y $87 izquierda
        BRSET PORTC,X $40 izquierda
der1    CMPB #$02
        BNE sigui1
        BRSET VAR,Y $87 derecha
        BRSET PORTC,X $80 derecha

sigui1

        BRSET VAR,Y $85 derecha
        BRSET VAR,Y $84 derecha
        BRSET VAR,Y $81 avanza
        BRSET VAR,Y $03 izquierda
        BRCLR VAR,Y $85 izquierda

        BRA inicio
derecha
      LDAA #$58
      BRA  inicio

izquierda
```

```
      LDAA #$38
      BRA  inicio
avanza
            LDAA #$18
inicioaux   BRA  inicio


finaux BRA fin
iniante2        BRA otr

ini2auxd
        LDAB $#02
        BRA ini2sender

ini2auxi
        LDAB $#01

ini2senizq
         BRSET PORTC,X $40 continuaizq
         BRA continua2i


continuaizq     BRCLR PORTC,X $C0 iniante2
                BRCLR PORTC,X $40 continuaizq

continua2i      BRSET PORTA,X $87 izq2senizq
                BRSET PORTA,X $03 izq2senizq
                BRSET PORTA,X $01 avan2senizq
                BRSET PORTA,X $00 der2senizq



der2senizq
      LDAA #$58
        STAA PORTA,X
        BRA  ini2senizq

izq2senizq

      LDAA #$38
        STAA PORTA,X
        BRA  ini2senizq



avan2senizq
      LDAA #$18
        STAA PORTA,X
        BRA  ini2senizq

;=================================================

ini2sender
         BRSET PORTC,X $80 continuader
         BRA continua2

continuader     BRCLR PORTC,X $C0 iniante2
                BRCLR PORTC,X $80 continuader

continua2       BRSET PORTA,X $87 der2sender
                BRSET PORTA,X $84 der2sender
                BRSET PORTA,X $80 avan2sender
                BRSET PORTA,X $00 izq2sender
```

```
der2sender
      LDAA #$58
        STAA PORTA,X
        BRA  ini2sender

izq2sender

      LDAA #$38
        STAA PORTA,X
        BRA  ini2sender



avan2sender
      LDAA #$18
        STAA PORTA,X
        BRA  ini2sender



fin     LDAA #$40
        STAA PORTA,X
        END
```

Diagrama a “mano alzada”: