

Apéndice B

Manual de usuario de la CT6811

La finalidad de este manual es dar una idea global de la CT6811, sin entrar en detalles demasiado técnicos, para que el usuario sea capaz de manejarla y sacar el máximo provecho de ella. Este manual no es un curso de aprendizaje del 68hc11, sino una guía para poder usar la CT6811.

B.1 Primeros pasos

A continuación se enumeran los pasos para ponerla en funcionamiento en modo entrenador sin necesidad de conocer nada más.

1. Asegúrese de que los jumpers JP1, JP2, JP3 y JP8 estén colocados.
2. El jumper JP5 deberá estar quitado. Si la tensión de alimentación es inferior a 5v el jumper JP6 deberá estar quitado también. El estado predeterminado es quitado.
3. El jumper triple JP4 estará situado en la posición RST (a la derecha).
4. El jumper triple JP7 estará situado en la posición ON (abajo).
5. Conecte la tarjeta al PC por medio del adaptador y del cable.
6. Alimente la tarjeta con una tensión comprendida entre 4.5 y 6 voltios (tensión nominal 5v). (Se aconseja leer el apartado 7 de este manual).
7. Teclee `downmcu ctserver -com1 (-com2)`.
8. Teclee `ctdialog -com1 (-com2)`.

El software para el control de la CT6811 se ha desarrollado para las plataformas MSDOS y LINUX, todos los programas tienen una interfaz muy parecida aunque no idéntica, por eso según cual sea la plataforma escogida por el lector puede haber alguna que otra diferencia. Los ejemplos de este manual se refieren a la plataforma LINUX por ser esta la escogida para la realización de este proyecto.

El manual de usuario que se incluye a continuación ha sido escrito por Cristina Doblado Alcázar, Juan González Gómez, Juan José San Martín y el propio autor de este proyecto, por eso y contando con la autorización de mis compañeros incluyo el manual para completar la información del capítulo 4 dedicado a la electrónica de control.

B.2 Presentación de la tarjeta CT6811

La CT6811 (figura 4.3, página 73) 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 terminados. Se trata sobre todo de una tarjeta muy versátil, que se puede emplear como:

1. **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'.
2. **Tarjeta autónoma de control:** 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...
3. **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...

B.2.1 Características de la CT6811

1. Microcontrolador **68HC11**: Por ello incorpora todas las características de este micro:
 - (a) 512 Bytes de EEPROM en los modelos A1 y A8. En el modelo E2 hay 2048 bytes de EEPROM
 - (b) 256 Bytes de memoria RAM interna
 - (c) Temporizador de 16 bits
 - (d) 3 capturadores de entrada
 - (e) 5 comparadores con salida hardware
 - (f) Un acumulador de pulsos
 - (g) Comunicaciones serie asíncronas
 - (h) Comunicaciones serie Síncronas
 - (i) 8 canales de conversión analógico digital
 - (j) Interrupciones en tiempo real
 - (k) 4 puertos de E/S

2. **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.
3. **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
4. **Led de pruebas** conectado al bit 6 del puerto A. Este led se puede conectar y desconectar por medio de un jumper
5. **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.
6. **Niveles VRH y VRL** configurables a VCC y masa respectivamente mediante 2 jumpers
7. Modos de funcionamiento **entrenador/autónomo** configurables mediante un jumper
8. **Alimentación** a través de clemas o conector tipo jack
9. **Conexión directa al PC** por medio de un cable tipo teléfono
10. **Reset software y hardware** de la placa. El jumper JP7 permite anular el reset software.

B.2.2 Diagrama de bloques de la CT6811

El núcleo central de la tarjeta CT6811 está constituido por el microcontrolador 68HC11A1 de Motorola. Los demás bloques surgen como una extensión del 68HC11, configurándolo para funcionar adecuadamente y adaptándolo para las necesidades del usuario. La tarjeta se puede dividir en 8 bloques (ver figura B.1): 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:

1. **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.
2. **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.
3. **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

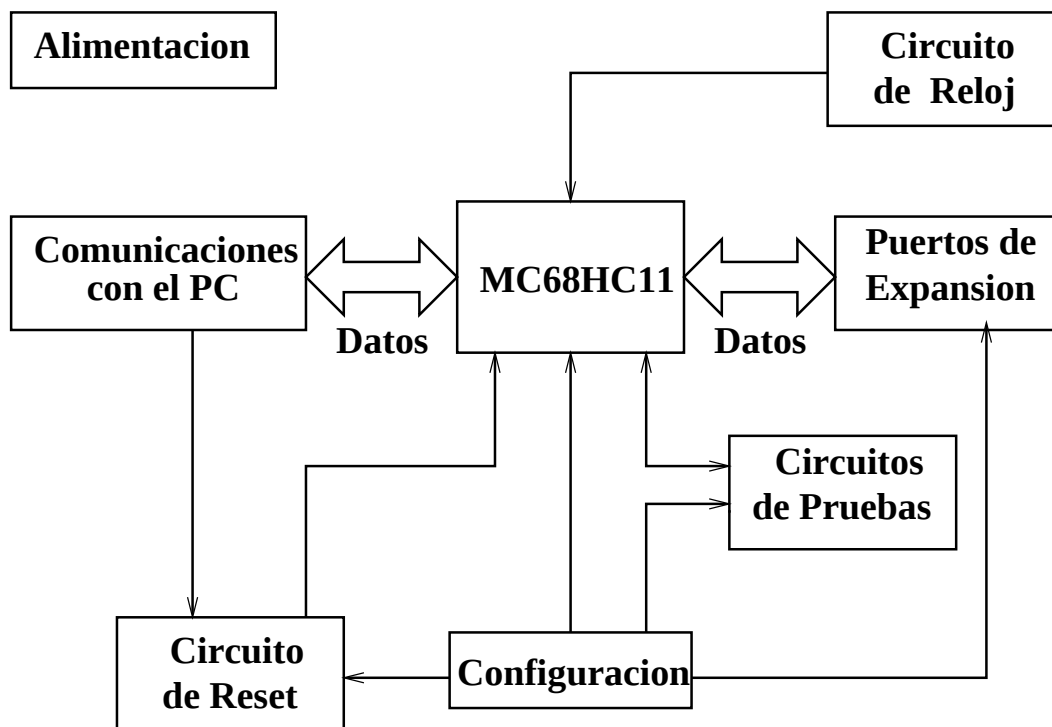


Figura B.1: Diagrama de bloques de la CT6811.

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.

4. **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 a través 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.
5. **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.
6. **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

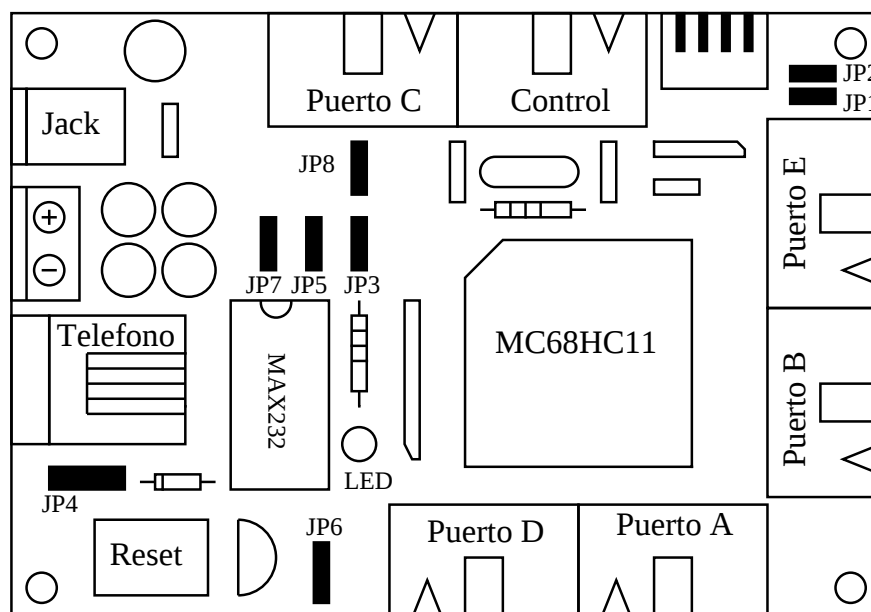


Figura B.2: Componentes de la CT6811.

la 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.

7. **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.
8. **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 conexión de cables de alimentación, provenientes por ejemplo de pilas o baterías.

B.2.3 Aspecto físico y situación de los componentes

En el diseño de la CT6811 se han tenido en cuenta las siguientes condiciones: Los componentes no pueden ser SMD, tienen que ser normales para permitir una sustitución fácil. Hay que facilitar la ampliación del sistema, para ello hay que dividir el bus en grupos funcionales y permitir un acceso sencillo a ellos. Por último hay que buscar el menor tamaño posible..

En la figura B.2 se aprecia la disposición de los componentes de la CT6811, el tamaño final es de 8.2 x 6.4 cm. Estas dimensiones hacen que sea una herramienta muy buena para el diseño de microbots y sistemas empotrados. Además el sistema se ha pensado para ampliarse verticalmente, de manera que las futuras expansiones no aumenten la superficie sino el volumen.

De todos los componentes de la CT6811 los que más pueden interesar al usuario son los siguientes:

- Los jumpers: JP1, JP2, JP3, JP4, JP5, JP6, JP7, JP8
- Puertos de expansión: Puerto A, Puerto B, Puerto C, Puerto D, Puerto E y Control.
- Conectores de alimentación: Jack y clema de alimentación.
- Switches de configuración.

B.3 Modos de funcionamiento

La tarjeta CT6811 se ha diseñado para trabajar en dos modos diferentes: modo autónomo y modo entrenador. Estos modos se configuran mediante los switches de control y uno de los jumpers (JP5).

Funcionamiento en modo entrenador

En esta configuración, la CT6811 se conecta a un PC a través del puerto serie. Los programas se escriben y se compilan en el PC y mediante un programa de comunicaciones serie se envían a la entrenadora para ser ejecutados. También es posible en este modo utilizar el PC como si fuese un terminal tonto de la tarjeta, empleando el teclado del PC para enviar datos a la CT6811 y utilizando el monitor del PC para visualizar datos generados en la CT6811.

El modo entrenador también se emplea para realizar programas de autotest de la tarjeta y presentar los resultados en el monitor del PC.

La configuración típica en modo entrenador tiene todos los jumpers colocados excepto el JP5, el JP4 situado a la derecha (RST) y el JP7 abajo (ON). Los switches 1 y 2 deben estar hacia abajo. Además la CT6811 tiene que estar conectada al PC a través del cable de teléfono y mediante un transformador o baterías alimentada a una tensión entre 4.5 y 6 voltios.

Funcionamiento en modo autónomo

En este modo de funcionamiento la CT6811 al ser inicializada (pulsar botón de reset o alimentar la tarjeta) comienza a ejecutar el programa que se encuentra en la memoria EEPROM interna del micro. Este programa ha sido grabado previamente en modo entrenador utilizando un software especial.

La CT6811 se diseñó pensando en los microbots, pequeños robots móviles y buscando un sistema pequeño y autónomo de control. Por eso este modo adquiere un gran protagonismo en las aplicaciones con la tarjeta. Para configurarla hay dos alternativas dependiendo del modelo de microcontrolador que posea la tarjeta:

1. Lo típico es disponer del 68HC11A1 y entonces todos los jumpers deberán estar colocados, el JP4 situado a la derecha y los switches abajo. Estando la CT6811 alimentada al apretar el pulsador se ejecutará el programa residente en la eeprom: Lo malo es

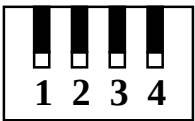
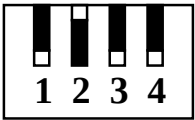
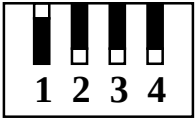
SWITCH 1 y 2	MODO	DESCRIPCION
	BOOTSTRAP	Ejecucion programa BootStrap Modo especial Sin acceso a memoria externa
	SINGLE CHIP	Vector interrupcion en ROM Sin acceso a memoria externa
	SPECIAL TEST	Acceso a memoria externa Modo especial Vector interrupcion en ROM
	EXPANDED	Vector interrupcion en RAM o ROM Acceso a memoria externa

Figura B.3: Configuración de los modos del 68hc11 mediante los switches de la CT6811.

que al estar puesto el jumper JP5 se están perdiendo las comunicaciones serie, salvo que después del arranque se libere dicho jumper.

2. Cada vez es más común disponer del 68HC11E2, microcontrolador que tiene 2Kbytes de memoria EEPROM situados en las posiciones \$F800 - \$FFFF. Para arrancar el modo autónomo se puede hacer lo mismo que antes, o también arrancar la placa con el switch 2 arriba y con el *vector de reset* apuntando al principio del programa (esto último se hace en el programa de la aplicación). La ventaja es que el jumper JP5 puede estar quitado y con ello no se pierden las comunicaciones serie.

B.4 Configuración de la tarjeta

La tarjeta CT6811 dispone de switches y jumpers para que el usuario la configure según sus necesidades. Existen 4 switches y 8 jumpers. De los 4 primeros, dos se utilizan para configurar los modos de funcionamiento del 68HC11 y los otros dos quedan disponibles para las aplicaciones del usuario.

B.4.1 Configuración de los modos del micro

El 68HC11 puede funcionar en 4 modos diferentes. Utilizando los mismos nombres que figuran el manual de 68HC11, los modos son: *single chip*, *special bootstrap*, *expanded* y *special test*. Los dos primeros son los más utilizados y en caso de disponer de un microcontrolador 68HC11A1 se puede reducir el número al modo *Bootstrap*.

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.

Special_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.

Estos modos de funcionamiento del micro se pueden configurar mediante 2 switches en la CT6811. 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 se ha cambiado el microcontrolador, utilizando en lugar del modelo A1, el modelo E2, que permite tener grabado el vector de reset en la EEPROM interna.

Para configurar los modos del micro, sólo se utilizan los switches 1 y 2. Estos switches actúan directamente sobre los pines MODA y MODB del 68HC11, poniéndolos a VCC o a GND. En la figura B.3 se encuentra resumida la información sobre la configuración de los modos.

B.4.2 Configuración de los jumpers

Existen 8 jumpers que 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 través del puerto E. El jumper JP2 actúa sobre VRL. Si está colocado, la pata 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.

- **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/IRQ. 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 CT6811. Cuando el jumper está situado en la posición de la izquierda (marcada como IRQ), el pulsador se conecta a la entrada de interrupción 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:** Configuración modo de funcionamiento de la CT6811: Entrenador o autónomo. Este es uno de los jumpers más importantes. Nos permiten decidir el funcionamiento de la CT6811: modo autónomo o modo 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. El inconveniente de tenerlo puesto es que se pierden las comunicaciones serie, por eso hay veces que es mejor adaptar un pulsador a JP5 y arrancar el programa pulsando primero reset y luego el pulsador nuevo. Mediante este truco se recuperan las comunicaciones serie. Si se quiere evitar hacer todo esto hay que cambiar el modelo de microcontrolador, en lugar de utilizar el A1 hay que poner el E2. Para arrancar en modo autónomo configuraremos la tarjeta en modo *single_chip* sin necesidad de utilizar el jumper JP5.

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 FUNCIONARÁN LAS COMUNICACIONES CON EL PC. Hay veces que es mejor adaptar un pulsador a JP5 y arrancar el programa pulsando primero reset y luego el pulsador nuevo. Mediante este truco se recuperan las comunicaciones serie. Si se quiere evitar hacer todo esto hay que cambiar el modelo de microcontrolador, en lugar de utilizar el A1 hay que poner el E2. Ahora, para arrancar en modo autónomo se configura la tarjeta en modo *single_chip* mediante los switches y por lo tanto no se tiene que utilizar el jumper JP5.

- **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, es decir por debajo de 4.5 voltios, el MC34064 realizará

JUMPER	FUNCIÓN	CONECTADO	QUITADO
JP1	Actuar sobre VRH	VRH = Vcc	VRH accesible desde el bus E
JP2	Actuar sobre VRL	VRL = GND	VRL accesible desde el bus E
JP3	Actuar sobre el LED	LED conectado a PA6	LED desconectado
JP4	Usos del pulsador	Izquierda: IRQ	Derecha: Reset
JP5	Modo funcionamiento	Modo Autónomo	Modo Entrenador
JP6	Actuar sobre LVI	LVI conectado	LVI desconectado
JP7	Reset Software On/Off	Arriba: Reset_Soft Off	Abajo: Reset_Soft On
JP8	Actuar sobre XIRQ	Modo normal	Sólo para EPROM del E9

Tabla B.1: Significado de los jumpers de la CT6811.

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.

- **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 software pero se gana el poder utilizar otros programas de comunicaciones. Si el jumper está colocado abajo (on) estará activado el reset software, si está colocado arriba (off) estará desactivado.
- **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 interferir los circuitos de la tarjeta CT6811 es 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 jumper.

B.5 Puertos de expansión

La tarjeta CT6811 dispone de 6 puertos de expansión (ver figura B.4). Cinco de estos puertos coinciden con los cinco puertos del 68HC11 y se han denominado igual: puerto A, puerto B, puerto C, puerto D y puerto E. El sexto puerto contiene otras señales de interés del 68HC11.

En la figura B.4 está representado uno de los conectores de los puertos, que está constituido por dos filas paralelas de 5 pines, a los cuales se podrán conectar cables planos de tipo bus de 10 bits. Se utilizan este tipo de cables por los siguientes motivos:

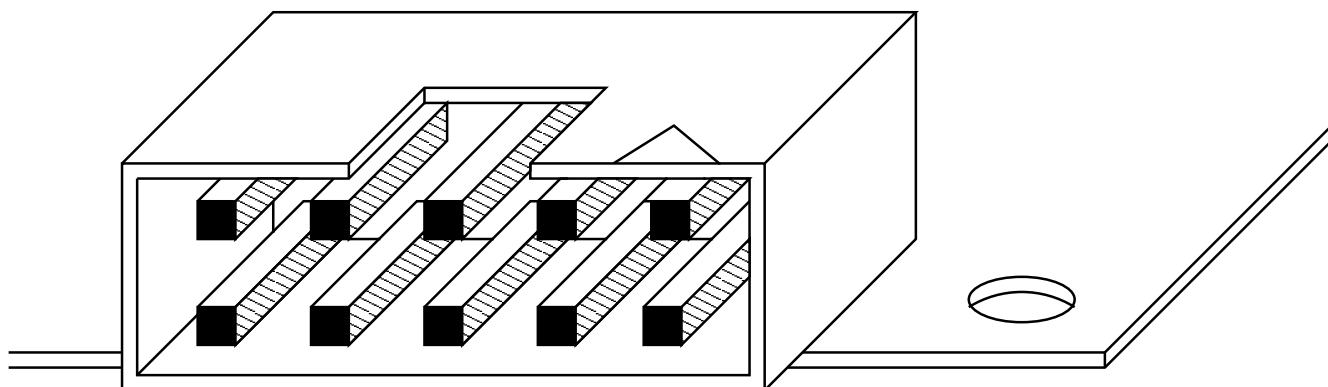


Figura B.4: Conector de los puertos de expansión.

Fiabilidad: Se trata de un cable muy resistente ya que al estar todo integrado en un sólo tipo de cable, se reduce el ruido y el peligro de cortocircuitos.

Multiplicidad: Es posible situar muchos conectores en el mismo cable, lo que lo hace especialmente útil para la conexión de varios periféricos al mismo puerto de expansión.

En la figura B.5 se detalla la correspondencia entre los pines del conector y los del microcontrolador. A continuación se describe cada puerto de la CT6811.

1. **Puerto A:** Este puerto está constituido por los 8 bits del puerto A del 68HC11 más un pin de VCC y otro pin de GND ambos para poder llevar alimentación a los periféricos de una forma cómoda.
2. **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.
3. **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.
4. **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.

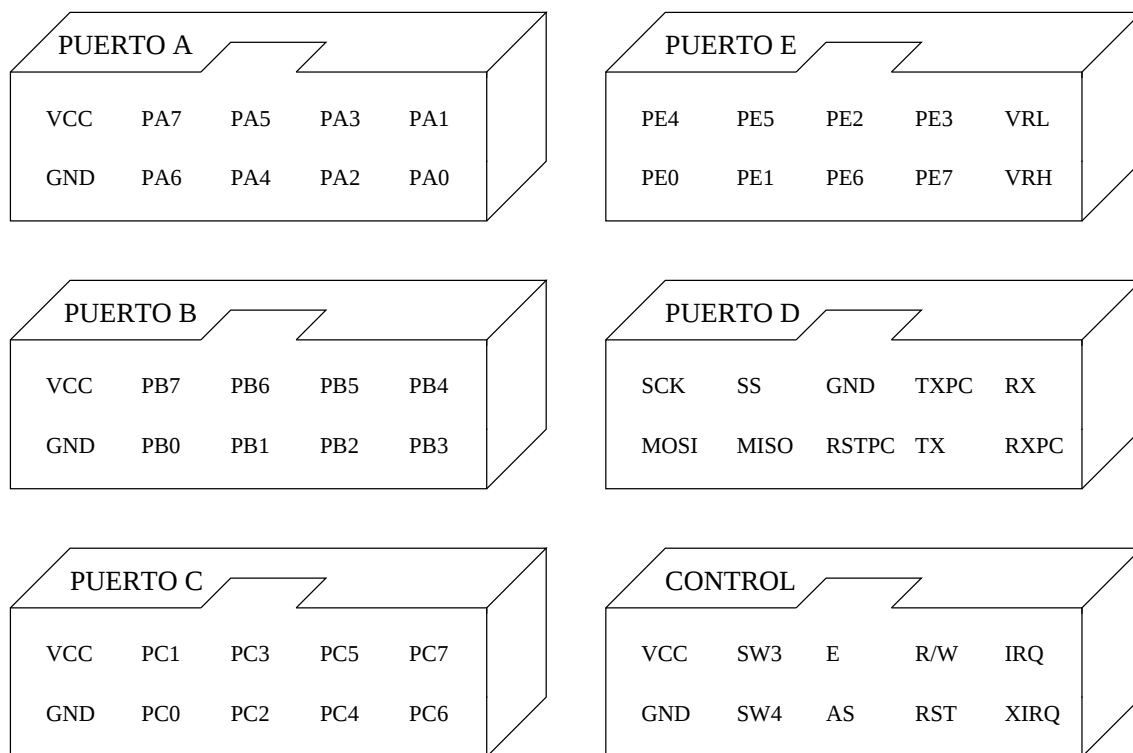


Figura B.5: Señales de los puertos de expansión. Los puertos se miran como se indica en la figura B.4.

5. **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 convertidor. 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.
6. **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.

B.6 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.

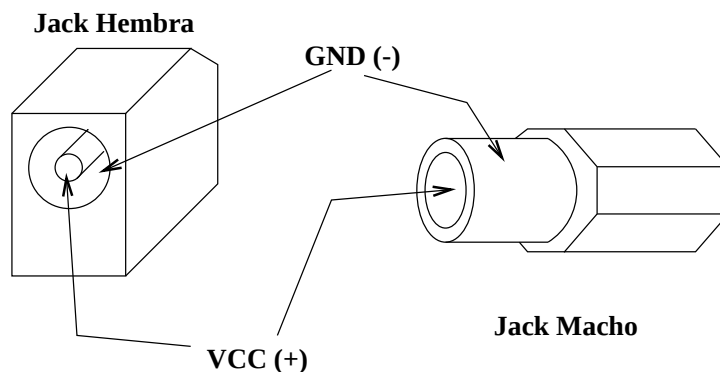


Figura B.6: Jack de alimentación.

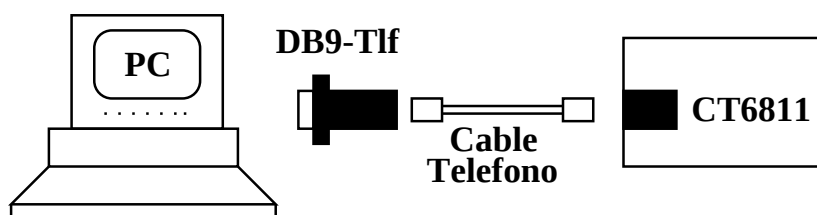


Figura B.7: Conexión PC - CT6811.

En la figura B.6 se ha representado el conector hembra de alimentación y el jack macho cilíndrico. Es importante hacer notar que **la parte exterior del jack macho debe ser GND** (negativa), mientras que **la parte interior debe ser VCC** (positiva). 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 interior.

Si se desea utilizar las clemas de alimentación la polaridad se encuentra indicada en la CT6811. La etiqueta Vcc significa que por ahí hay que introducir la tensión positiva (+), la etiqueta GND indica que por ahí hay que introducir la negativa o masa (-).

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 vez**. O se utiliza el transformador, o se utilizan las bornas.

B.7 Conexión al PC

La conexión al PC se realiza por medio de dos elementos: un cable de teléfono de 4 hilos y un conector DB9-TLF que tiene una entrada para el cable de teléfono por un lado y por el otro lado se conecta directamente a un puerto serie del PC.

El cable de teléfono empleado es de 4 hilos, pero no todos los cables sirven. Existen dos tipos de cables según cómo estén situados los conectores de los extremos (ver figura). Un cable es válido cuando las pestañas de los conectores de teléfono están situadas en el mismo sentido, es decir, o ambas coinciden con la línea marcada en la superficie del cable, o ambas no lo hacen, pero nunca una si y otra no.

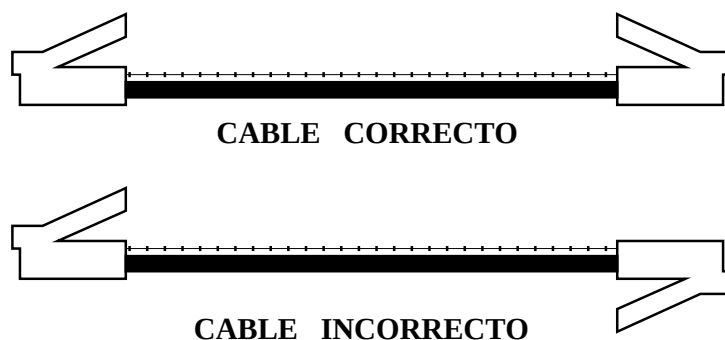


Figura B.8: Cable de teléfono para unir la CT6811 con el PC.

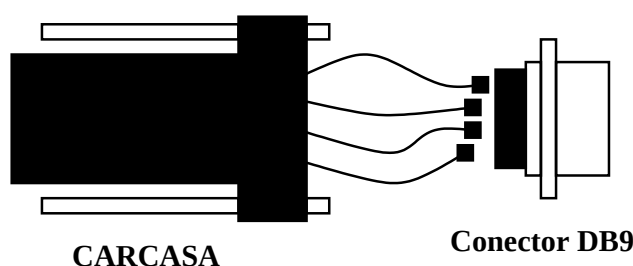


Figura B.9: Conector DB9 - Teléfono.

El cable utilizado es de 4 hilos, por ellos viajarán las señales: TX, RX, GND y DTR. La señal DTR es la utilizada para realizar el reset software de la tarjeta CT6811. El conector DB9-TLF tiene por un lado un conector hembra DB9 y por el otro una hembra de teléfono de 4 vías. Estos conectores se venden desmontados y en su interior hay cuatro cables que tendrán que introducirse en los pines correcto de la salida DB9 del mismo. Generalmente los colores son los indicados en la figura B.10. Pero lo que realmente interesa es realizar las conexiones internas correctamente. Si al desmontar el conector los colores son iguales se puede realizar la asignación tal y cómo indica la figura. Por ejemplo el cable rojo se insertará en el pin 5 del conector DB9. Si los colores no son iguales se tiene que revisar la asignación pin a pin. Por ejemplo, un cable azul que sale del pin 6 de la carcasa (conector teléfono) se tiene que unir con el PIN 2 del conector DB9 (PC).

COLOR	PIN DB9	CARCASA (CABLE)	FUNCIÓN
NEGRO	2	6	RX PC
AMARILLO	3	1	TX PC
VERDE	4	5	RESET
ROJO	5	2	GND

Tabla B.2: Significado de las señales del conector.

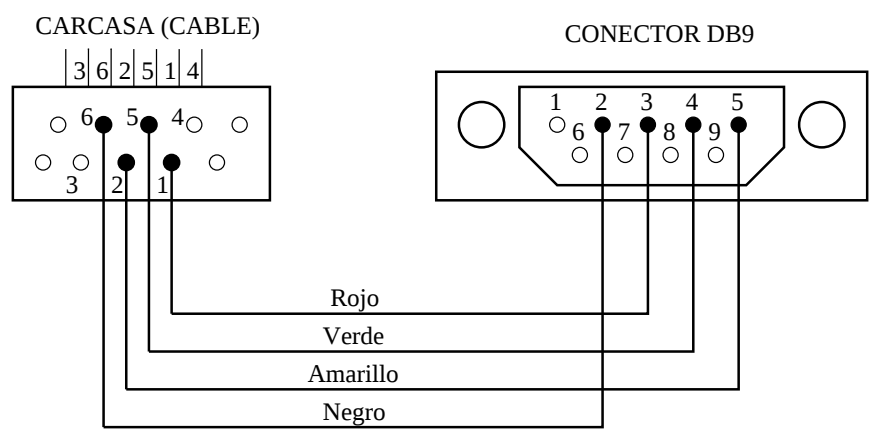


Figura B.10: Unión cable - PC.

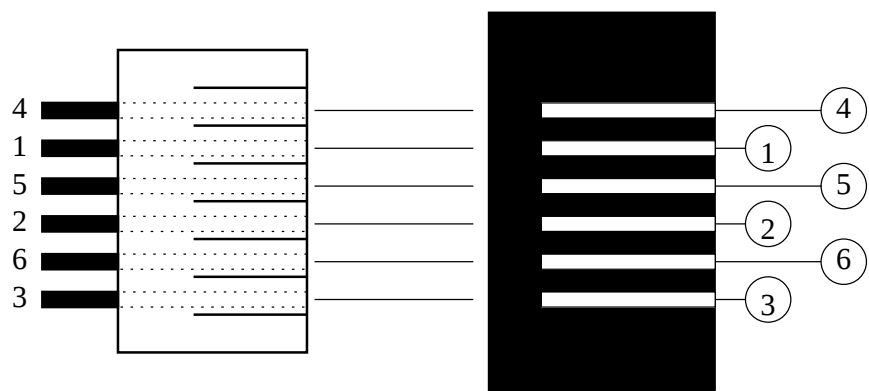


Figura B.11: Unión cable - CT6811.

B.8 Pruebas de funcionamiento

B.8.1 Probando la CT6811 en modo autónomo.

La CT6811 se entrega con un pequeño programa de prueba para comprobar que todo funciona adecuadamente. El programa simplemente hace parpadear el led de la CT6811 intermitentemente. Para ejecutar el programa de pruebas siga los siguientes pasos:

- Coloque los jumpers JP5, JP3. Situar el jumper JP4 en la posición más de la derecha, indicada en la CT6811 como RST. Los demás jumpers pueden estar colocados o quitados, es indiferente.
- Sitúe los switches 1 y 2 hacia abajo. Los switches 3 y 4 pueden estar en cualquier posición.
- Alimente la CT6811, bien con un transformador, bien con pilas.
- Apriete el pulsador para hacer un reset.

Se tendrá que encender intermitentemente el led de la CT6811. Si no funciona correctamente, revise los puntos anteriores y vuélvalo a intentar. Si sigue sin funcionar, puede ser que la EEPROM se haya desprogramado. Haga las pruebas que se indican en el apartado siguiente.

Si todo ha funcionado correctamente, la CT6811 ha funcionado en modo autónomo. En el siguiente apartado se va a probar la CT6811 en modo entrenador.

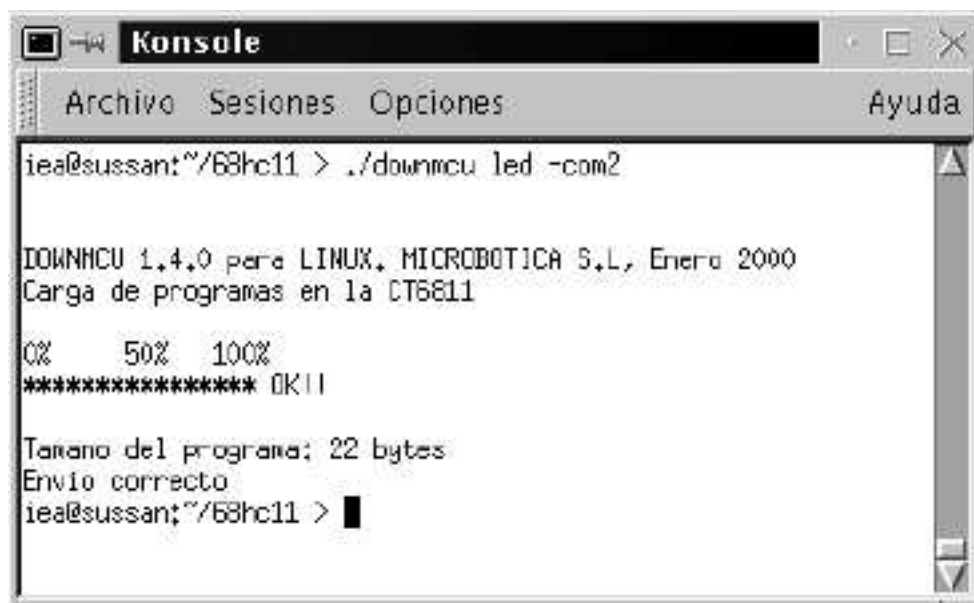
B.8.2 Probando la CT6811 en modo entrenador.

En este apartado se va a cargar un programa en la CT6811 desde el PC utilizando el *downmcu*. Para ello, primero debe configurar la CT6811 adecuadamente:

- Coloque el jumper JP3, JP8 y el JP4 sitúelo en la posición más a la derecha indicada por la etiqueta RST. El jumper JP5 debe estar quitado y el JP7 colocado en la posición de ON (abajo). Los demás jumpers se pueden encontrar conectados o desconectados.
- Sitúe los switches 1 y 2 hacia abajo. Los switches 3 y 4 pueden estar en cualquier posición.
- Conecte la CT6811 al puerto serie del ordenador (COM1 ó COM2) utilizando el cable proporcionado.
- Alimente la CT6811 por medio de un transformador o por medio de pilas

Ahora la CT6811 se encuentra dispuesta para funcionar en modo entrenador. En el PC debe disponer de los siguientes ficheros: el programa *downmcu*, que probablemente esté en */usr/local/bin*¹ y el *ledp.s19* en el directorio de trabajo. Estos son los mínimos ficheros necesarios para realizar estas pruebas.

¹Localización por defecto en las máquinas Linux, en MSDOS estará en el path o en el directorio de trabajo.



```
iea@sussan:~/68hc11 > ./downmcu led -com2

DOWNMCU 1.4.0 para LINUX, MICROBOTICA S.L, Enero 2000
Carga de programas en la CT6811

0%    50%   100%
***** OK!!

Tamaño del programa: 22 bytes
Envío correcto
iea@sussan:~/68hc11 > █
```

Figura B.12: Carga del programa LEDP.S19 en la CT6811 utilizando el *downmcu*.

Primero se va a cargar el programa *ledp.s19* en la CT6811. Este programa simplemente hace parpadear el led de la tarjeta. La extensión *.S19* indica que se trata de un programa ejecutable por el 68HC11. El programa en ensamblador, por si lo quiere ver, se encuentra en el fichero *LEDP.ASM*. En la figura B.12 se muestra lo que hay que teclear y los resultados obtenidos. Se ha supuesto que la CT6811 se encuentra conectada en el puerto COM2. Si se encuentra en el COM1, cambie el parámetro *-com2* por *-com1* (¡Siempre en minúsculas!).

El programa *downmcu* implementa el reset software automáticamente, por eso el jumper JP7 deberá estar puesto en la posición indicada con *ON* en la CT6811. Si se coloca al contrario se desactiva el reset software y cada vez que los programas pidan hacer un reset se deberá pulsar el botón situado en la CT6811 y para que éste funcione el jumper JP4 deberá estar situado a la derecha (posición *RST*).

Si el proceso de carga ha ido bien se apreciará que el LED de la CT6811 está parpadeando y significa que se ha podido enviar un programa a la memoria RAM correctamente. En el siguiente capítulo se escribirán los pasos necesarios para poder grabar los programas en la EEPROM y obtener sistemas autónomos.

B.9 Desarrollo de programas para la CT6811

En este apartado se van a dar unos ejemplos de programación de la CT6811. Para ello se va a utilizar un software concreto: el ensamblador AS11 de Motorola y los programas *downmcu* y *ctdialog*.

Se utilizan las versiones para LINUX, pero se recuerda que también se dispone del software para MSDOS, el cual se maneja de forma similar al explicado aquí.

B.9.1 Filosofía de trabajo

La filosofía empleada para el manejo de la CT6811 es la siguiente. En el PC se programan las aplicaciones para la CT6811 en ensamblador del 68HC11. Estos programas fuente tienen extensión .ASM. Utilizando el ensamblador FREEWARE AS11 de Motorola, los programas se compilan, obteniéndose archivos ejecutables con extensión .S19. Estos son los que se envían a la CT6811.

Para enviar programas a la CT6811 se pueden utilizar varios programas. En los ejemplos de más adelante se utiliza el programa DOWNMCU v.10. Una vez que se ha enviado el programa, el 68HC11 comienza su ejecución. Se puede cargar programas en la CT6811 tantas veces como se quiera. Los programas así cargados son almacenados en la RAM interna del 68HC11.

Una vez que se tiene la aplicación depurada, se graba en la memoria EEPROM del 68HC11 para que se quede permanentemente. Ahora la CT6811 no necesita del PC. Funciona en modo autónomo. Para grabar programas en la EEPROM interna se pueden utilizar varios programas: Pc-bug, ctdialog... En esta sección se utilizará el programa *ctdialog*. Este programa, además de permitir grabar la EEPROM, permite ver el contenido de la memoria del 68HC11, cambiar valores de la memoria, desensamblar ...

B.9.2 Un ejemplo completo

Se presenta un programa fuente que se va a compilar para después ser cargado en la CT6811 y finalmente grabado en la EEPROM. El programa hace parpadear un led y sus fuentes se puede ver en la figura B.13.

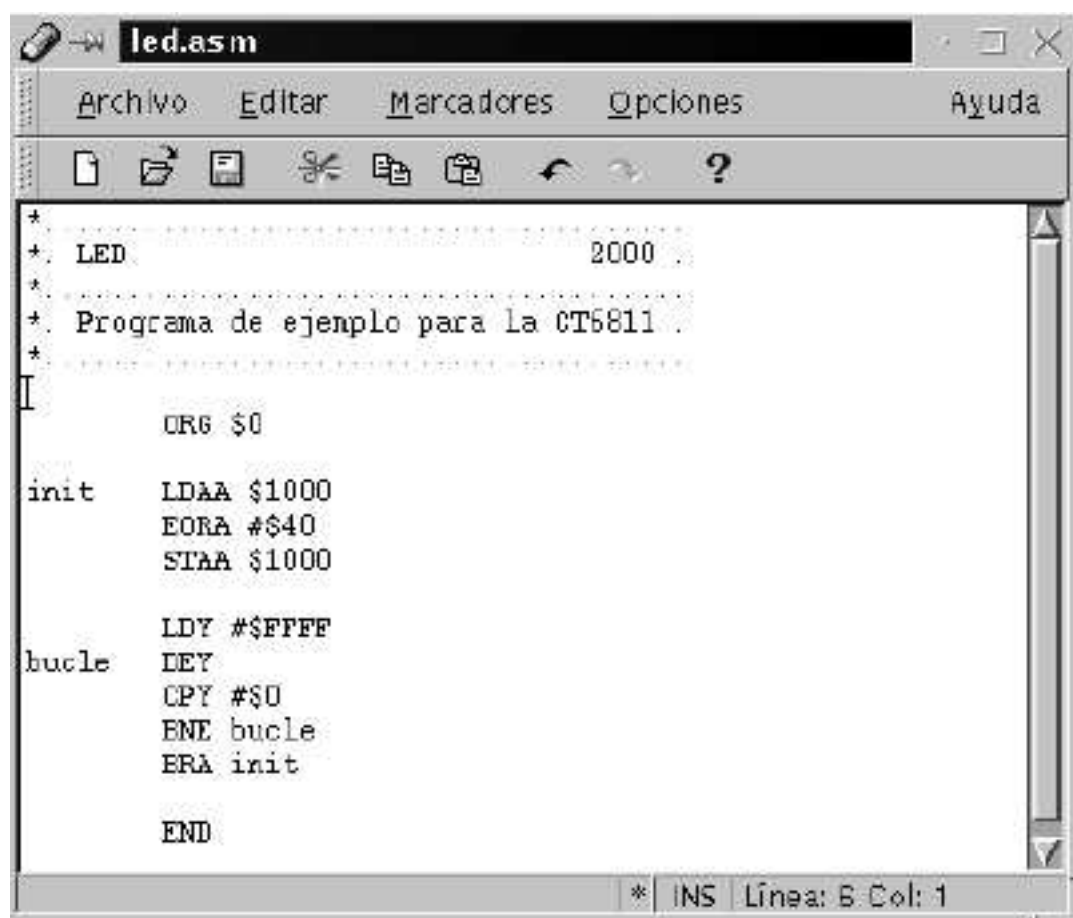
Lo primero que hay que hacer es escribir el programa con algún editor de textos, para luego grabarlo con la extensión ASM (que indica que es el código fuente); en este caso el programa se llamará *LED.ASM*. El siguiente paso es compilar el programa para obtener el archivo *LED.S19*, que es el fichero ejecutable y por lo tanto entendible por el microcontrolador. Para ello se utiliza el ensamblador AS11 de Motorola de la siguiente forma:

```
> as11 led.asm
```

Se obtiene el archivo *led.s19* que se trata de un archivo de texto y por tanto se puede editar (¡Pero no se debe modificar!).

Para probar si funciona el programa se cargará directamente en la CT6811, para ello se utiliza el programa *downmcu* de la forma explicada en la sección anterior. En la figura B.14 se muestran los comandos para compilar el fichero y para luego enviarlo a la CT6811. Se ha supuesto que la CT6811 se encuentra conectada en el puerto serie COM2, si se quiere utilizar el puerto COM1 basta con cambiar el parámetro *-com2* por *-com1* (¡en minúsculas!). En cuanto se recibe el mensaje de 'ENVIO CORRECTO' el programa ya se estará ejecutando en el 68HC11 y se podrá ver cómo parpadea el LED.

Hasta ahora, todos los programas enviados al 68HC11 se han almacenado en la RAM interna del micro, esto significa que al quitar la alimentación estos se perderán. A pesar de que se pueden cargar todas las veces que se quiera, es necesario de disponer de algún mecanismo para poder dejarlos permanentemente grabados la tarjeta independientemente del estado de la alimentación. Esto se consigue enviando los programas a la memoria EEPROM interna del 68hc11, la cual una vez grabada mantiene los datos aunque se apague



The screenshot shows a window titled "led.asm" with a menu bar (Archivo, Editar, Marcadores, Opciones, Ayuda) and a toolbar. The code is as follows:

```
* ..... 2000
* LED
* .....
* Programa de ejemplo para la CT6811
* .....
I
    ORG $0

init  LDAA $1000
      EORA #$40
      STAA $1000

      LDY #$FFFF
bucle DEY
      CPY #$0
      BNE bucle
      ERA init

      END
```

The status bar at the bottom indicates "* INS Línea: 6 Col: 1".

Figura B.13: Listado del programa LED.ASM



```

iea@sussan:~/68hc11 > as11 led.asm
6811 assembler 10-Aug-91
original program by Motorola,
y adaptado por Grupo J&J
iea@sussan:~/68hc11 > ./downmcu led

DOWNMCU 1.4.0 para LINUX, MICROBOTICA S.L., Enero 2000
Carga de programas en la CT6811

0%    50%   100%
***** OK!!

Tamaño del programa: 22 bytes
Envío correcto

iea@sussan:~/68hc11 > █

```

Figura B.14: Proceso para compilar y enviar el programa a la CT6811.

la tarjeta. Para poder enviar los programas a dicha memoria hay que utilizar un software distinto del *downmcu*, en concreto el *ctdialog*.

Antes de nada, hay que modificar el programa fuente para indicar que se quiere situar en la memoria EEPROM. Si el programa a grabar en la EEPROM utilizase variables, habría que situar las variables en la RAM interna y dejar sólo el código en la EEPROM. Como el programa *led* no tiene variables, no hay que preocuparse, pero sí que hay que cambiar la directiva *ORG*, que indica el comienzo del programa. En vez de comenzar en la dirección \$0000, ahora deberá comenzar en la dirección \$b600, que es el comienzo de la memoria EEPROM en el modelo de microcontrolador MC68HC11A1². El nuevo programa creado se denomina *lede.asm* para diferenciarlo del anterior. Una vez compilado este programa, de forma similar a como se ha hecho con el programa *led.asm*, se obtiene el fichero *lede.s19*, que será el que se utilice para grabar la EEPROM.

El programa que se va a emplear es el *ctdialog* (figura B.15). El proceso de carga de este programa depende de la plataforma sobre la que se ejecuta el software. En el caso de LINUX es suficiente con llamar al programa de la siguiente forma:

```
> ctdialog -com2
```

Al igual que antes si en lugar de estar conectada la CT6811 al segundo puerto serie estuviera en el primero se pondría como parámetro *-com1*. En el caso de MSDOS hay que hacer un proceso más largo; Primero hay que enviar a la CT6811 el programa CTSERVER y después se ejecuta el programa CTDIALOG.EXE.

```
> downmcu ctserver -com2
```

²Si el modelo de microcontrolador es el MC68HC11E2 hay que poner *ORG \$F800*.

```

Konsole
Archivo Sesiones Opciones Ayuda

ico@sussan:~/68hc11 > ./ctdialog

Cargando CTSERVER:

0% 50% 100%
*****
OK!
Puerto actual: COM2
Estableciendo conexion con tarjeta..... conexion establecida

CTDIALOG Version 1.4.0 (C) MICROBOTICA, Enero 2000
Teclee HELP para obtener ayuda

bit 0: EEON=1 -> EEPROM activada,
bit 1: ROMON=1 -> ROM activada,
bit 2: NOCOP=1 -> COP desactivado,
bit 3: NOSEC=1 -> SECURITY desactivada, (Solo disponible en modelos
fabricados con la mascara de seguridad).

Registro CONFIG: FF
El modelo de micro se corresponde con la familia Ex
Posicion eeprom: F800 - FFFF

>

```

Figura B.15: Programa CTDIALOG ejecutado en una CT6811 con un microcontrolador E2.

```
> ctdialog -com2
```

En ambos casos el programa *ctdialog* permite grabar la EEPROM del 68HC11, pero también sirve para ver el contenido de la memoria, modificarla, desensamblar, ejecutar programas, reconfigurar el micro, etc.

Para grabar el programa *lede.s19* en la EEPROM hay que utilizar el comando '*eeeprom*' seguido del nombre del fichero:

```
>eeeprom lede
```

Para probar el programa, se puede configurar la CT6811 en modo autónomo (conectando el jumper JP5) y hacer un reset. También es posible realizar un 'salto' a la EEPROM desde el *ctdialog*, para ello se utiliza el comando '*g*' seguido de la dirección a donde se quiere saltar:

```
>g b600
```

Al realizar el salto, el 68HC11 comienza a ejecutar el programa almacenado en la EEPROM. El servidor que estaba en la RAM interna se deja de ejecutar y por tanto se pierde la conexión del *ctdialog* con la CT6811. Para volver a ejecutar el programa habrá que salir utilizando el comando '*quit*', volver a cargar el servidor en caso de estar en MSDOS (bajo LINUX no hace falta) y ejecutar el *ctdialog* otra vez.

El *ctdialog* dispone de otros comandos y entre ellos una ayuda en línea que muestra un resumen de todos:

>help

Otro comando interesante es '*ms*' que permite escribir un byte (segundo argumento) en una posición de memoria (primer argumento). Por ejemplo, la instrucción siguiente encenderá el LED de la CT6811.

>ms 1000 40

El valor 1000 es la dirección en hexadecimal del Puerto A y el valor 40 es el valor en hexadecimal del byte que hace que el bit PA6 se ponga a uno encendiendo LED. Para apagarlo hay que poner en la dirección 1000 el valor 0, es decir:

>ms 1000 0

Por último hay que hacer una puntualización sobre el uso del *ctdialog* bajo MSDOS y con una tarjeta que incorpore el microcontrolador 68hc11E2 en lugar del A1. En este caso, antes de grabar la EEPROM hay que desprotegerla, para ello teclear:

>ms 1035 0

>eprom 'programa'

La primera instrucción desprotege la EEPROM y la segunda la graba. Si lo que se quiere es borrarla en lugar de utilizar '*eprom*' hay que poner '*bulk*'.

B.10 68HC11 y comunicaciones serie: El programa MCBOOT.

El programa *mcboot* es un programa de comunicaciones para comunicarse con la tarjeta CT6811. Además de funcionar como terminal de comunicaciones también permite cargar programas en la RAM interna del 68HC11.

En la depuración de programas y pruebas de periféricos es muy útil poder contar con el teclado y pantalla del PC. Utilizando programas de comunicaciones serie en el 68HC11 se va a poder utilizar la pantalla del PC para representar datos y el teclado del PC para enviar datos al 68HC11.

B.10.1 Programa de ejemplo scihola.asm

A continuación se presenta un pequeño programa para probar las comunicaciones serie. El programa *scihola.asm*, cuyo código está en la figura B.16, simplemente espera a que se pulse una tecla en el PC y responde enviando la cadena "HOLA COMO ESTAS...". Existen dos subrutinas de comunicaciones serie: **enviar** para enviar un carácter por el puerto serie y **leer_car** para esperar a que venga un carácter por el puerto serie. Estas rutinas están relacionadas con el puerto serie del 68HC11 y no se explican en este manual. El bucle principal está constituido por una llamada a la subrutina de *leer_car* para que el micro se quede esperando recibir un carácter por el puerto serie. Una vez recibido el carácter, se envía por el puerto serie la cadena especificada utilizando la función **send_cad**.

Este programa se puede enviar a la CT6811 utilizando el programa de comunicaciones *mcboot*. Antes de cargar el programa hay que seleccionar el puerto en el que está conectada la CT6811, el predeterminado es el COM2, pero pulsado la tecla F4 se puede cambiar a otro. **Asegúrese de que NO está activado el reset software. Para ello pulse F7 hasta que aparezca en la pantalla DTR OFF, ya que siempre que esté DTR ON la CT6811 estará reseteada y no se podrán cargar programas.**

Para cargar un programa pulse la tecla F1 e introduzca el nombre del fichero .S19 a cargar. En este caso será el programa *scihola*, el cual envía al PC la cadena "hola como estas..." cada vez que recibe por el puerto serie la tecla pulsada en el PC.

B.10.2 Programa de ejemplo menú

Con el siguiente ejemplo se pretende mostrar la posibilidad de realizar programas de prueba que ofrezcan diferentes opciones al usuario. En concreto, el programa *menu.asm* presenta en pantalla dos opciones. Según la opción que elija el usuario el micro cambiará el estado del led o volverá a imprimir el menú. A partir de este programa modelo el usuario puede ir practicando y realizando otros más complejos.

```
* +-----+
* | MENU                                MICROBOTICA, 2000 |
* |-----|
* | Programa ejemplo para ser ejecutado en la          |
* | tarjeta CT6811. Este programa se debe cargar      |
* | en la RAM interna del 6811.                        |
* |                                                    |
* | Ejemplo de como manejar un menu de opciones      |
* | para programas interactivos con el usuario        |
* +-----+
```

```
CR      equ 13      * Retorno de carro
LF      equ 10      * Avance de linea
```

```
* Registros del SCI
```

```
BAUD    equ $2B
```

```

* ..... MICROBOTICA, 2000
* Programa de ejemplo para ser ejecutado en la CT6811.
* .....

BAUD equ $2B
SCCR1 equ $2C
SCCR2 equ $2D
SCSR equ $2E
SODR equ $2F

ORG $0
LDX #$1000 * Para acceder a registros del SCI

bucle BSR leer_car * Esperar a que llegue un carácter por SCI
      LDY #hola * Meter en Y la dirección de la cadena hola
      BSR send_cad * Enviar la cadena por el puerto serie
      BRA bucle

leer_car BRCLR SCSR,X,$20 leer_car * Esperar hasta que llegue un carácter
      LDAA SODR,X
      RTS

enviar BRCLR SCSR,X,$80 enviar
      STAA SODR,X
      RTS

send_cad LDAA 0,Y * Meter en A el carácter a enviar
          CMPA #0 * ¿ Fin de la cadena?
          BEQ fin * Si--> retornar
          BSR enviar * NO--> enviar carácter.
          INY * Apuntar a la sig. posición de memoria
          BRA send_cad * Repetir todo

fin RTS

hola FCC "Hola como estas..."
      FCB 0

      END

```

Figura B.16: Código del programa SCIHOLA.

```

SCCR1    equ    $2C
SCCR2    equ    $2D
SCSR     equ    $2E
SCDR     equ    $2F

                ORG $0          * programa para la RAM
                LDX #$1000      * acceder a registros

bucle     LDY #menu
          BSR send_cad        * Sacar menu
wait      BSR leer_car        * Leer tecla
          CMPA #'1'
          BEQ opcion1        * Tecla '1' -> Opcion 1
          CMPA #'2'
          BEQ opcion2        * Tecla '2' -> Opcion 2
          BRA wait

opcion1   LDAA $1000
          EORA #$40          * Cambiar estado bit 6 puerto A
          STAA $1000
          BRA wait

opcion2   BRA bucle          * Volver a sacar el menu

*+-----+
*| Rutina par leer un caracter del puerto serie (SCI) |
*| La rutina espera hasta que llegue algun caracter |
*| ENTRADAS: Ninguna. |
*| SALIDAS: El acumulador A contiene caracter recibido |
*+-----+
leer_car  BRCLR SCSR,X $10 leer_car    * esperar caracter
          LDAA SCDR,X
          RTS

*+-----+
*| Enviar un caracter por el puerto serie (SCI) |
*| ENTRADAS: Acumulador A contiene caracter a enviar |
*| SALIDAS: Ninguna. |
*+-----+
enviar    BRCLR SCSR,X $80 enviar
          STAA SCDR,X
          RTS

*+-----+
*| Enviar una cadena de caracteres por el puerto SCI |

```

```

*| La cadena debe terminar con el caracter 0      |
*| ENTRADAS: Y contiene direccion cadena a enviar |
*| SALIDAS: Acumulador A contiene caracter recibido |
*+-----+
send_cad LDAA 0,Y      * Meter en A caracter a enviar
          CMPA #0      * ¿Fin de la cadena?
          BEQ fin      * Si--> retornar
          BSR enviar   * .NO--> enviar caracter.
          INY          * Apuntar a la sig. posicion
          BRA send_cad * Repetir todo
fin      RTS

```

```

*+-----+
*| DATOS |
*+-----+
menu    FCB CR,LF,LF
        FCC "          MENU DE OPCIONES"
        FCB CR,LF
        FCC "          ====="
        FCB CR,LF,LF
        FCC "  1.- Cambiar de estado el LED"
        FCB CR,LF
        FCC "  2.- Sacar este menu"
        FCB CR,LF
        FCC "  Opcion: "
        FCB 0

        END

```

B.10.3 El MCBOOT y la RAM externa

El *mcboot* tiene otras funciones hasta ahora no mencionadas ya que están desactivadas en la versión de Linux, pero no en la de MSDOS, por eso conviene mencionarlas:

- F2** Cargar programas en la RAM externa. Para utilizarla se debe disponer de la ampliación de memoria externa para la CT6811 y además disponer en el directorio donde se encuentre el *mcboot* el cargador *loader.s19*. La utilización de esta opción es igual que **F1**.
- F3** GAIA. Esta opción necesita las condiciones anteriores, pero en lugar de enviar cualquier programa carga un software especial que permite ver los registros internos, volcar datos, poner *break points*, etc. El *ctdialog* ha venido a sustituir esta opción del *mcboot*.
- F6** Volcar programas. Mediante esta opción se puede volcar un fichero de texto por el puerto serie indicado. La diferencia con **F1** está en que **F6** envía todo el fichero, mien-

```

Konsole
Archivo Sesiones Opciones Ayuda

iea@sussan:~/68hc11/c/ctload.1.4.0/s19 > ctload scihola -com2 -t

CTLOAD      ...  OK ...

Puerto actual      : COM2
Fichero             : scihola.s19
Tamaño             : 60 bytes
Direccion de comienzo: 0
PROGRAMA PARA LA RAM INTERNA

00000000  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
*****XXXXXXXXXXXX OK!!

TERMINAL ACTIVADO, ESC PARA TERMINAR

Hola como estas,,Hola como estas,,

```

Figura B.17: Programa *ctload* actuando de terminal.

tras que F1 sólo la parte correspondiente al código, dejando sin enviar las cabeceras y los bytes de control.

En Linux, para enviar programas a la RAM externa se puede utilizar el *ctload* (ver figura B.17), que funciona igual que el *downmcu*, solo que es capaz de distinguir si el programa es para la RAM interna o externa, cargándolo donde corresponda. Además este programa tiene la opción *'-t'* que permite después de cargar el programa dejar la consola como un terminal. Es decir, todo lo que se reciba se verá en pantalla y todo lo que se teclee se enviará por el puerto serie.

B.11 Utilización de la CT6811 con la familia 68HC11

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 B.3 se muestran los modelos que los autores han utilizado en alguna aplicación de la CT6811.

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,

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
MC68HC11E1	512	0	0	512
MC68HC711E9	512	0	12K	512

Tabla B.3: Modelos de microcontroladores compatibles con la CT6811.

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 TRAN-TOR, también desarrollada por el GRUPO J&J³ 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, incluido 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 podían ser mayores que los de la RAM, por lo tanto no se podían 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

³El Grupo J&J es como se hacían llamar los autores de la CT6811 antes de formar parte de la empresa Microbótica.

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'.

B.12 Características del 68HC11

1. En la figura B.18 se representa la numeración del zócalo PLCC de 52 pines del microcontrolador.
2. En la figura B.19 se representa el patillaje del microcontrolador de 48 pines con encapsulado DIP.
3. En la figura B.20 se representa el patillaje del microcontrolador de 52 pines con encapsulado PLCC.
4. En la figura B.21 se representa el diagrama de bloques del MC68HC11A1.



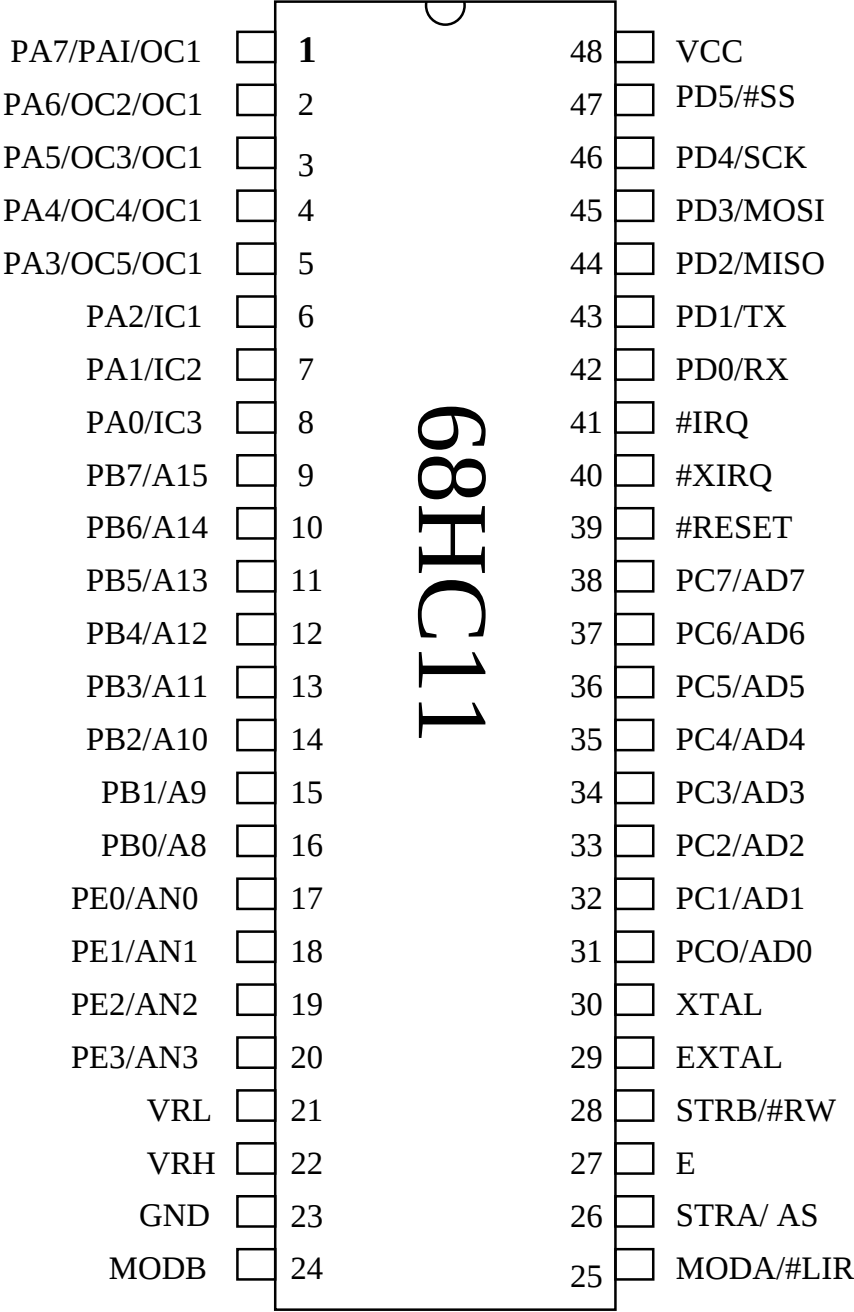


Figura B.19: Patillaje del 68HC11 en formato DIP.

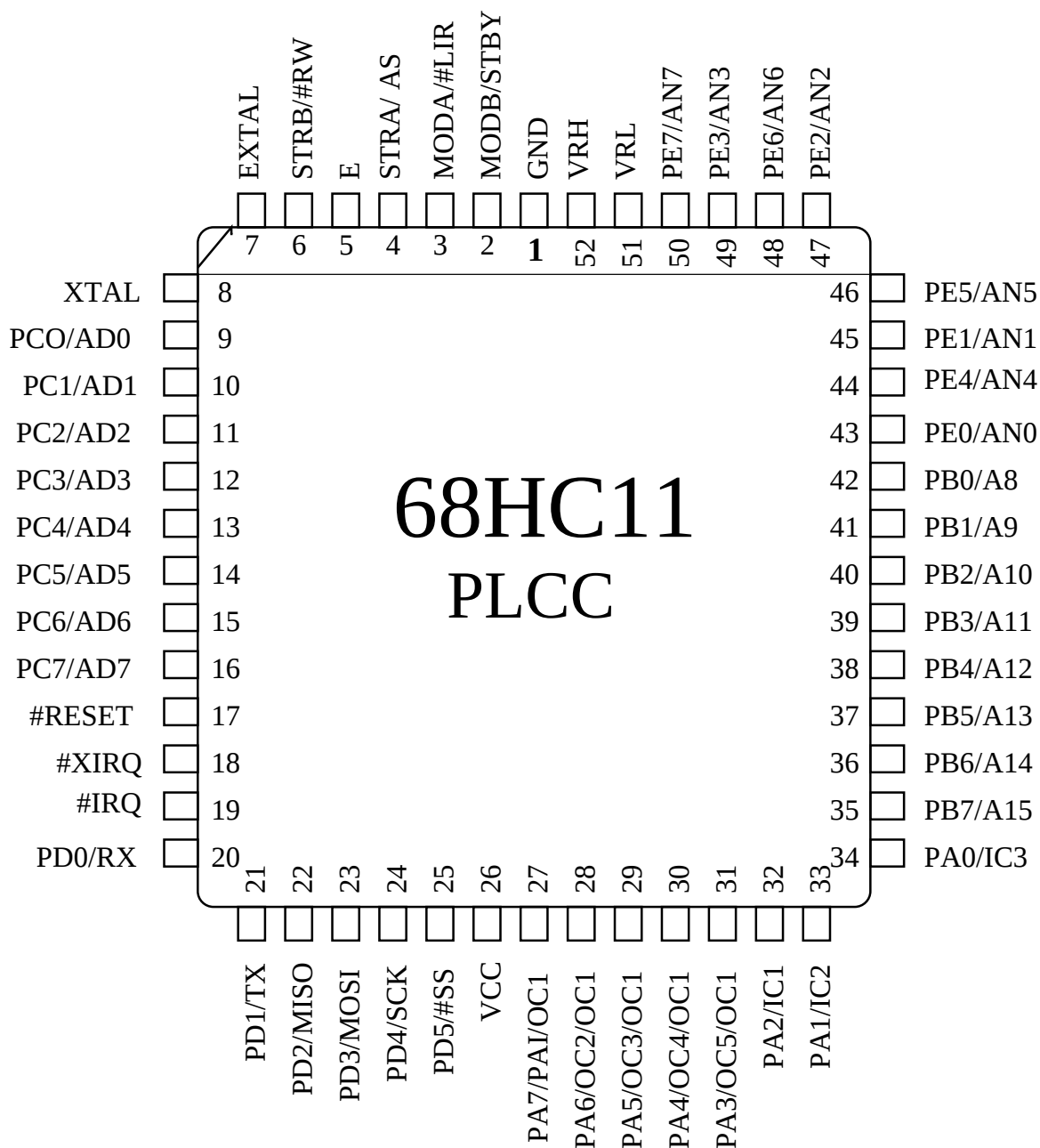


Figura B.20: Patillaje del 68HC11 en formato PLCC:

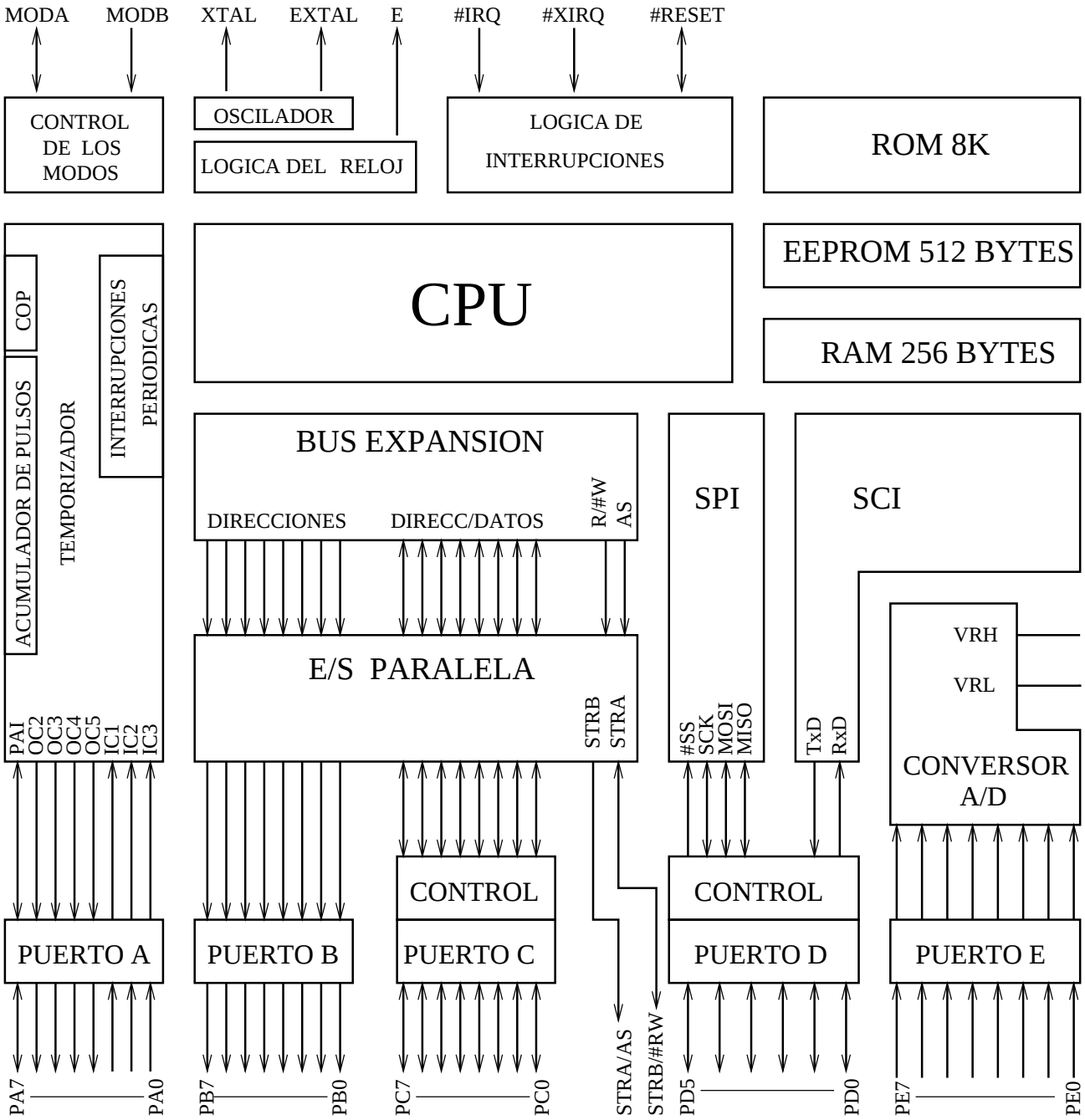


Figura B.21: Diagrama de bloques del MC68HC11A1.