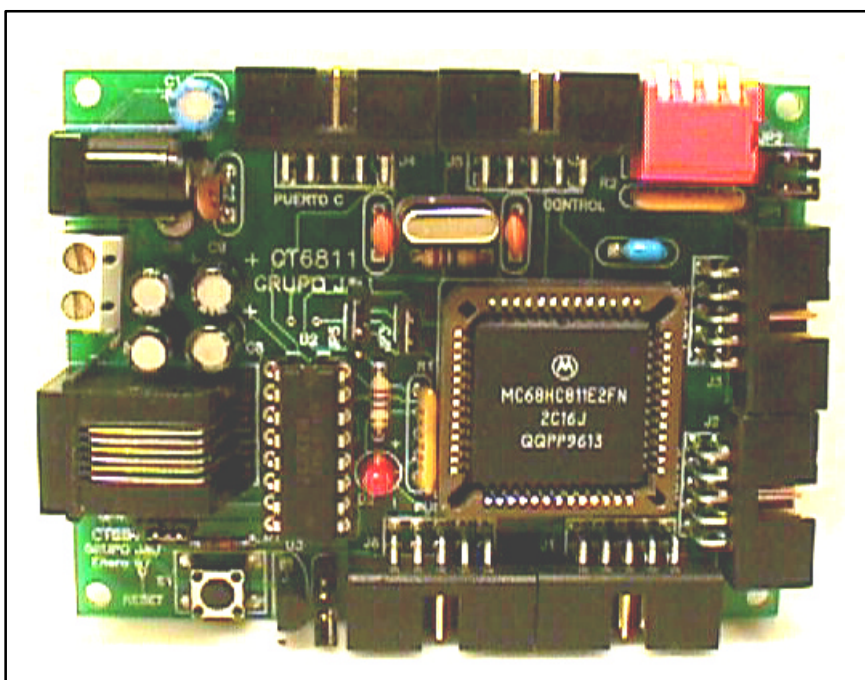


TARJETA DE DESARROLLO

CT6811

MANUAL DE USUARIO



MICROBÓTICA, S.L.

GRACIAS POR SU COMPRA

Gracias por su compra. Usted ha adquirido el sistema de desarrollo CT6811, el cual debe estar compuesto por:

- 1) Tarjeta CT6811.
- 2) Cable del tipo telefónico.
- 3) Adaptador RJ45-DB9.
- 4) Manual de usuario.
- 5) Software de aplicación.

El sistema CT6811 ha superado con éxito las pruebas de control de calidad realizadas en el laboratorio, finalizando las mismas con la carga y grabado de un programa en la memoria interna del microcontrolador.

El software para el control de la CT6811, incluido en el disco, ha sido desarrollado para las plataformas MSDOS y LINUX. Con una interfaz similar, se ha optado por hacer referencia en el manual a la versión de MSDOS por su popularidad.

Cualquier consulta puede ser realizada utilizando el servicio permanente de email, además le invitamos a ver nuestra página WEB.

Email: info@microbotica.es

WEB: www.microbotica.es

Esperamos que este producto sea de su agrado.

MICROBÓTICA, S.L.

0. GUÍA RÁPIDA DE CONEXIÓN AL PC DE LA TARJETA CT6811

- 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. Por defecto viene 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).

Las posiciones anteriores son las que trae la tarjeta CT6811 por defecto.

- 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 5.5 voltios (tensión nominal 5v). **(Se aconseja leer el capítulo 7 del manual de usuario de la CT6811)**
- 7) Teclee `downmcu ctserver -com1 (-com2)`
- 8) Teclee `ctdialog -com1 (-com2)`

CONTENIDO

0.- GUÍA RÁPIDA DE CONEXIÓN.....	4
1.- SOBRE ESTE MANUAL.....	7
2.- INTRODUCCION A LA CT6811.....	7
2.1.- Características de la CT6811.....	8
2.2.- Modos de funcionamiento de la CT6811.....	9
3.- DIAGRAMA DE BLOQUES DE LA TARJETA CT6811.....	10
4.- ASPECTO FISICO DE LA CT6811.....	12
5.- CONFIGURACION DE LA TARJETA.....	16
5.1.- Configuración de los modos del micro.....	16
5.2.- Configuración de los jumpers.....	17
5.3.- Configuración típica de la CT6811 en modo entrenador.....	19
5.4.- Configuración típica de la CT6811 en modo autónomo.....	19
6.- PUERTOS DE EXPANSION.....	21
7.- ALIMENTACION.....	24
8.- CONEXION AL PC.....	26
8.1.- Distintas partes que intervienen en la conexión.....	26
8.2.- El cable de teléfono.....	27
8.3.- El conector DB9-TLF.....	28
9.- PRUEBAS DE FUNCIONAMIENTO.....	31
9.1.- Probando la CT6811 en modo autónomo.....	31
9.2.- Probando la CT6811 en modo entrenador.....	31
10.-DESARROLLO DE PROGRAMAS PARA LA CT6811.....	33
10.1.- Filosofía de trabajo.....	33
10.2.- Un ejemplo completo.....	33

PARTE II:

ENTORNO DE DESARROLLO Y PROGRAMAS EJEMPLO

1.- INTRODUCCION.....	39
2.- CONTROL DE LA CT6811 DESDE EL PC: CTDIALOG.....	39
2.1.- Ejecutando el programa CTDIALOG.....	40
2.2.- Algunos comandos del programa CTDIALOG.....	41
2.3.- Acceso al LED de la CT6811 desde el CTDIALOG.....	42
3.- PROGRAMACION DE LA TARJETA CT6811.....	43

3.1.- Ficheros fuente y ficheros ejecutables.....	43
3.2.- Un primero programa ejemplo: Encender el LED.....	44
3.3.- Programa del LED parpadeante.....	46
4.- GRABACION DE PROGRAMAS EN LA EEPROM	47
5.- 68HC11 Y COMUNICACIONES SERIE: MCBOOT.....	50
5.1.- Programa ejemplo SCIHOLA.ASM.....	50
5.2.- Programa ejemplo MENU.ASM.....	52
6.- EL ENTORNO DE DESARROLLO TOWERBOOT.....	54
 APENDICE A: Modos de funcionamiento del 68HC11.....	 57
APENDICE B: Patillaje del 68HC11.....	59
APENDICE C: Numeración del zócalo PLCC de 52 pines.....	61
APENDICE D: Diagrama de bloques de la CT6811.....	63
APENDICE E: Utilización de la CT6811 con la familia de 68hc11..	65

1.- SOBRE ESTE MANUAL

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. Este manual *no es un curso de aprendizaje*, sino más bien una guía de manejo de la CT6811. Se ha orientado para utilizarlo fundamentalmente como una guía rápida de consulta, resumiendo toda la información en tablas y dibujos.

No se va a enseñar a programar el microcontrolador 68HC11, no obstante, se presentan algunos ejemplos.

2.- INTRODUCCION A LA 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 terminados**. 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 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 ...
- **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...

2.1.- Características de la CT6811

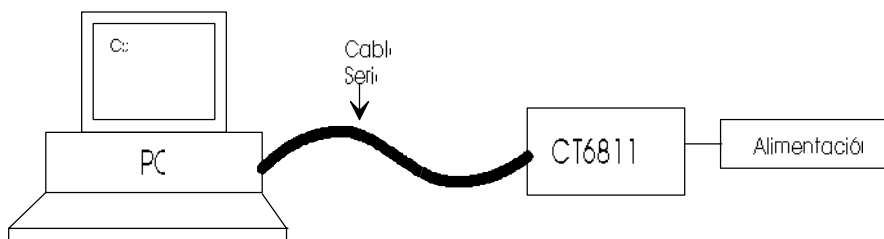
- **Microcontrolador 68HC11:** Por ello incorpora todas las características de este micro:
 - 512 Bytes de EEPROM en los modelos A1 y A8
 - 256 Bytes de memoria RAM interna
 - Temporizador de 16 bits
 - 3 capturadores de entrada
 - 5 comparadores con salida hardware
 - Un acumulador de pulsos
 - Comunicaciones 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

2.2.- Modos de funcionamiento de la CT6811

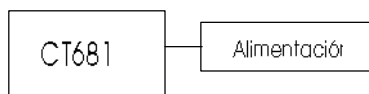
La tarjeta CT6811 se ha diseñado para trabajar en dos modos diferentes: modo autónomo y modo entrenador. Estos modos se configuran colocando o eliminando un jumper.

2.2.1.-Funcionamiento en modo entrenador

En la figura 1.a. se representa de forma esquemática el funcionamiento de la tarjeta CT6811 en su configuración como tarjeta entrenadora.



a) Funcionamiento de la tarjeta CT6811 en modo entrenador



b) Funcionamiento en modo autónomo

Figura 1: Distintos funcionamientos de la tarjeta CT6811

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.

2.2.2. 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 la EEPROM en modo entrenador utilizando un software especial.

El funcionamiento en modo autónomo es muy interesante ya que no se necesita disponer de un PC. Este modo es especialmente indicado para la construcción de microbots autónomos. La CT6811 ha sido diseñada específicamente para funcionar en modo autónomo, reduciendo las dimensiones al máximo para poder realizar microbots pequeños.

3.- DIAGRAMA DE BLOQUES DE LA TARJETA CT6811

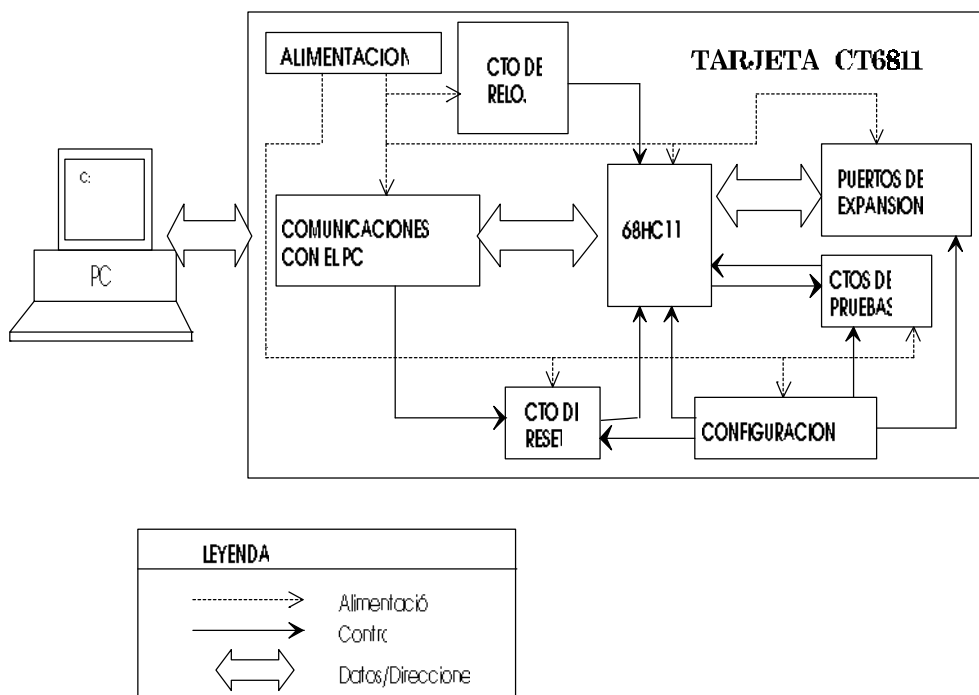


Figura 2: Diagrama de bloques de la tarjeta CT6811

El núcleo central de la tarjeta CT6811 está constituido por el microcontrolador 68HC11A1 de Motorola. Los demás bloques surgen como una extensión del 68HC11. 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 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.
- **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 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.

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

4.- ASPECTO FISICO DE LA CT6811

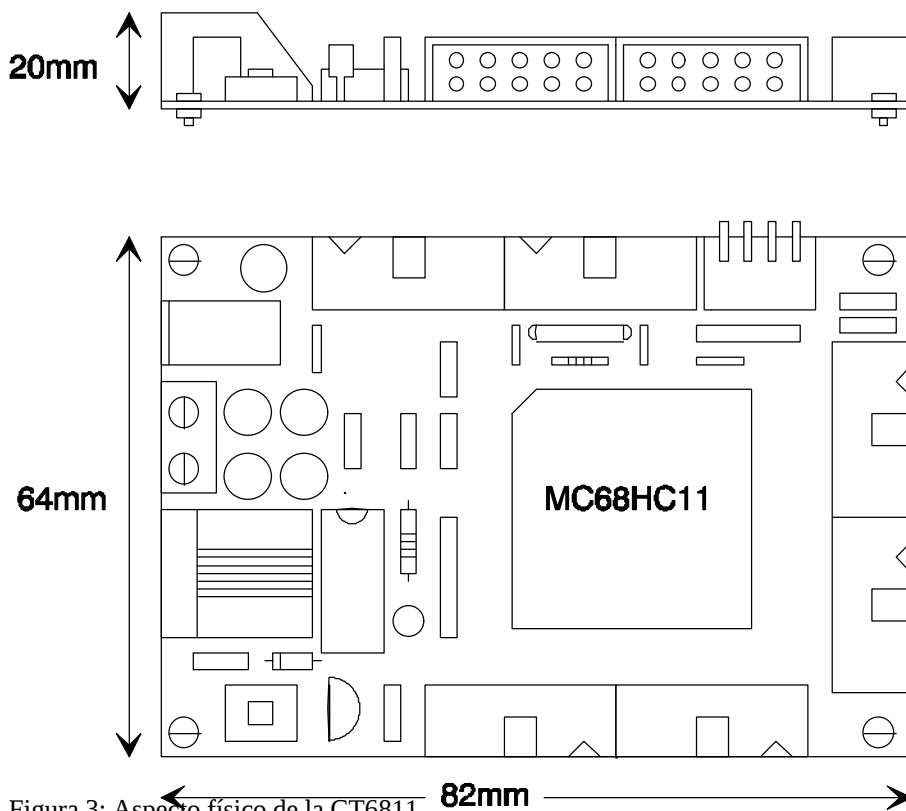


Figura 3: Aspecto físico de la CT6811

En la figura 3 se ha representado el aspecto físico y las dimensiones de la tarjeta CT6811. La tarjeta ha sido optimizada para ocupar la mínima superficie posible. Sus reducidas dimensiones, 8.2x6.4cm, hace que la CT6811 sea una herramienta ideal para el diseño de sistemas empotrados y construcción de microbots. El sistema se ha pensado para ser expandido hacia lo ‘alto’, de tal manera que las expansiones no ocupen superficie sino volumen.

A continuación se muestra la localización de las diferentes partes de interés para el usuario: Jumpers, switches, puertos de expansión, conectores de alimentación, pulsador de reset/IRQ y led de pruebas.

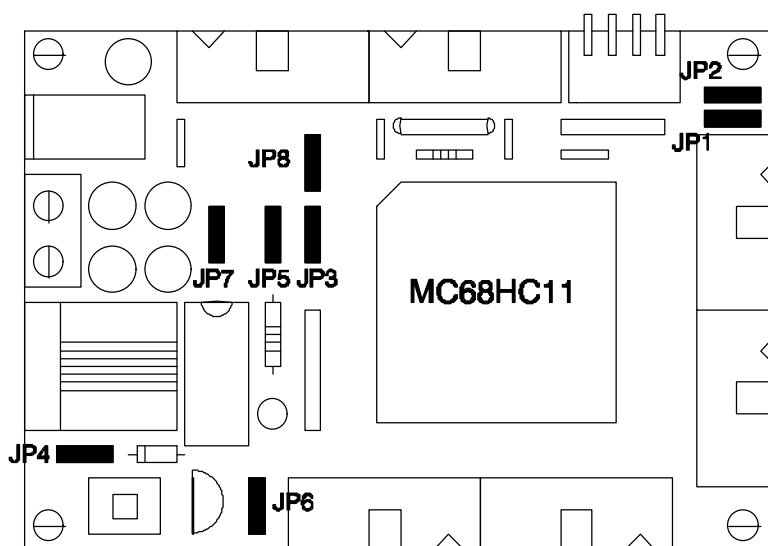


Figura 4: Situación los *jumers*

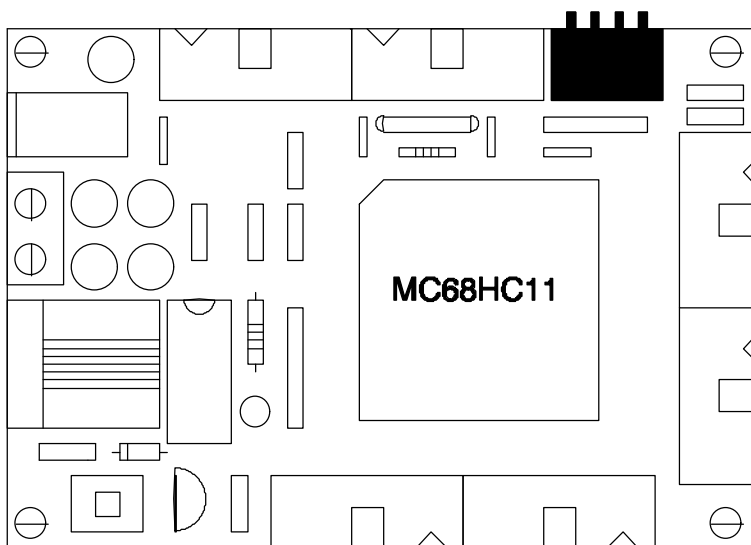


Figura 5: Situación de los switches de configuración

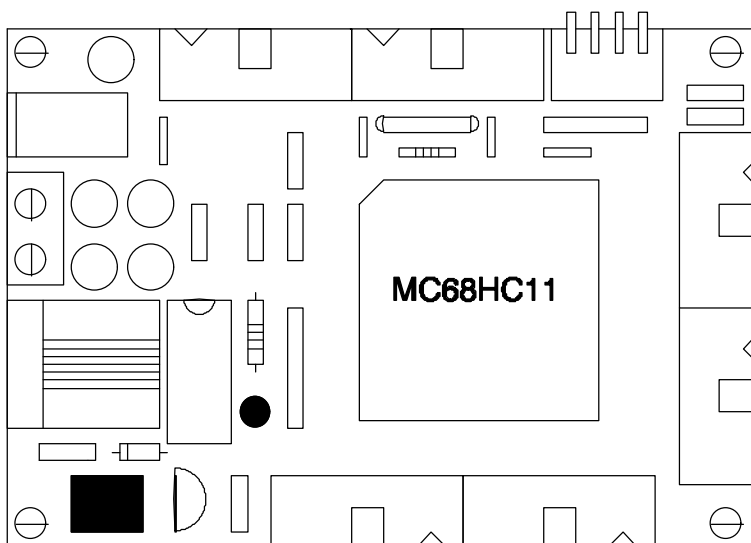


Figura 6: Situación del pulsador y del LED de pruebas

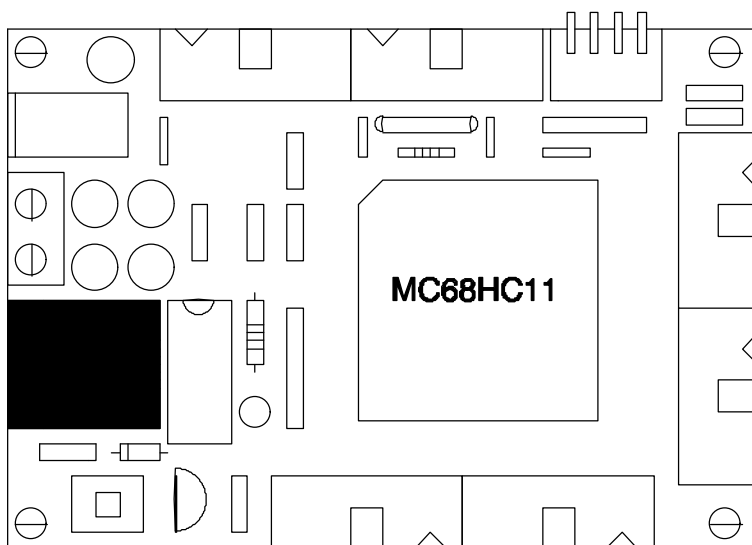


Figura 7: Situación del conector de teléfono para conectar el PC con la CT6811

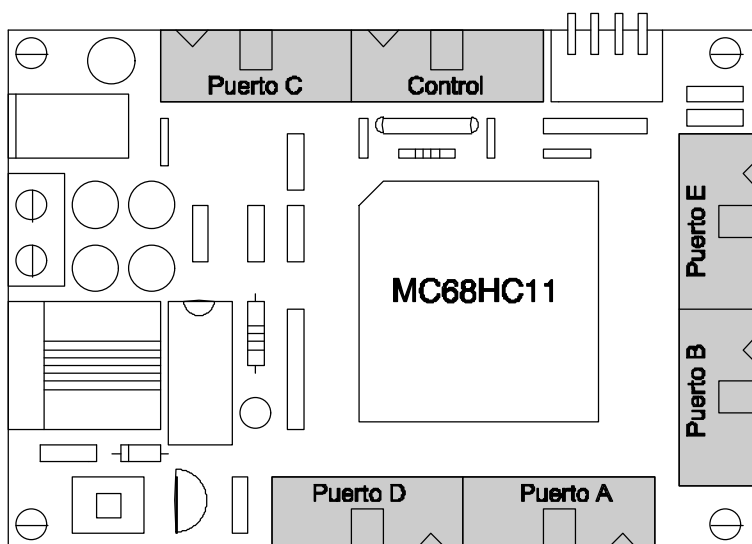


Figura 8: Situación de los puertos de expansión

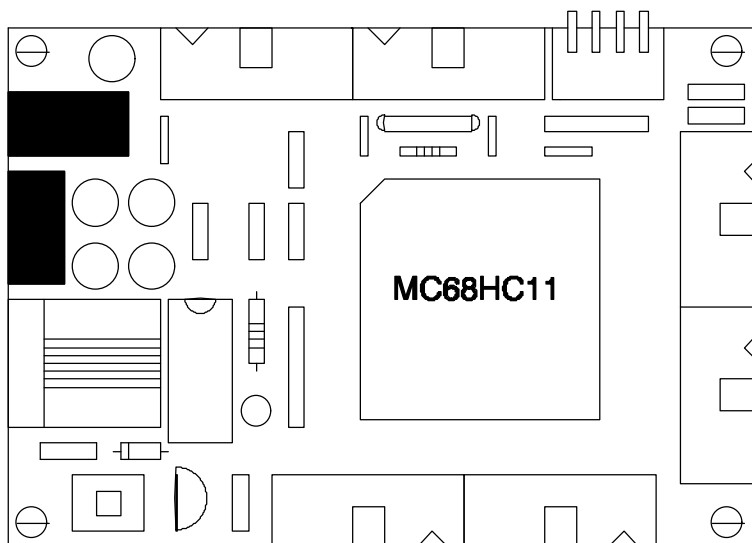


Figura 9: Situación de los conectores de alimentación

5.- CONFIGURACION 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 switches, dos se utilizan para configurar los modos de funcionamiento del 68HC11 y los otros dos quedan disponibles para las aplicaciones del usuario.

5.1.- 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*. En el apéndice A se puede encontrar más información sobre estos modos. El modo bootstrap es el modo más utilizado. Este modo se utiliza para trabajar con la CT6811 como tarjeta entrenadora y también como tarjeta autónoma. Sólo en aplicaciones más complejas se utilizarán los otros modos.

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 10 se encuentra resumida la información sobre la configuración de los modos.

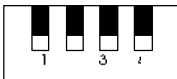
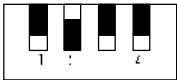
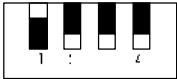
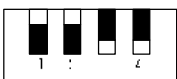
SWITCHES 1 y 2	MODA	MODB	MODO	DESCRIPCION
	C	0	Bootstrap	No acceso memoria externa Modo especial Ejecución programa BOOTSTRAP
	C	1	Single Chip	No acceso memoria externa Vectores interrupción en ROM
	1	0	Special test	Acceso a memoria externa Modo especial Vector interrupción en ROM
	1	1	Expanded	Acceso a memoria externa Vector interrupción en ROM ó RAM

Figura 10: Configuración de los modos mediante los switches de la tarjeta CT6811.

5.2.- 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 través 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.

- **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 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 JP5¹: 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.

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 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

¹ Si la CT6811 dispone del 68hc11E2 mirar el apéndice E.

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 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 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*.

En la figura 11 se ha resumido toda la información sobre la configuración de los jumpers. En el apartado “Configuración de la CT6811 en modo entrenador” se muestra un ejemplo típico de configuración de los jumpers y switches.

5.3.- Configuración típica de la CT6811 en modo entrenador

La configuración típica en modo entrenador es la mostrada en la figura 12: 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 simultáneamente**.

5.4.- 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. Esta configuración se presenta en la figura 13.

En el caso de disponer la CT6811 con el modelo de microcontrolador 68hc11E2, la configuración propuesta cambia. Se recomienda mirar el apéndice E.

JUMPER	FUNCIÓN	CONECTADO	QUITADO
JP1	Actuar sobre VRH	VRH = VCC	VRH accesible desde el bus de expansión
JP2	Actuar sobre VRL	VRL = GND	VRL accesible desde el bus de expansión
JP3	Actuar sobre el LED	LED conectado a PA6	LED desconectado PA6 liberado
JP4	Seleccionar usos del pulsador: IRQ/RESET	El jumper en la izquierda activa la IRQ	El jumper en la derecha activa el pulsador de reset
JP5	Seleccionar modo entrenador o autónomo	Modo autónomo	Modo entrenador.
JP6	Actuar sobre el LVI	LVI conectado	LVI desconectado
JP7	Seleccionar reset software ON / OFF	El jumper abajo activa el reset software	El jumper arriba desactiva el reset software
JP8	Actuar sobre XIRQ	Modo normal	Sólo en caso de programar la EPROM del 68HC11E9

Figura 11: Configuración de los jumpers de la CT6811.

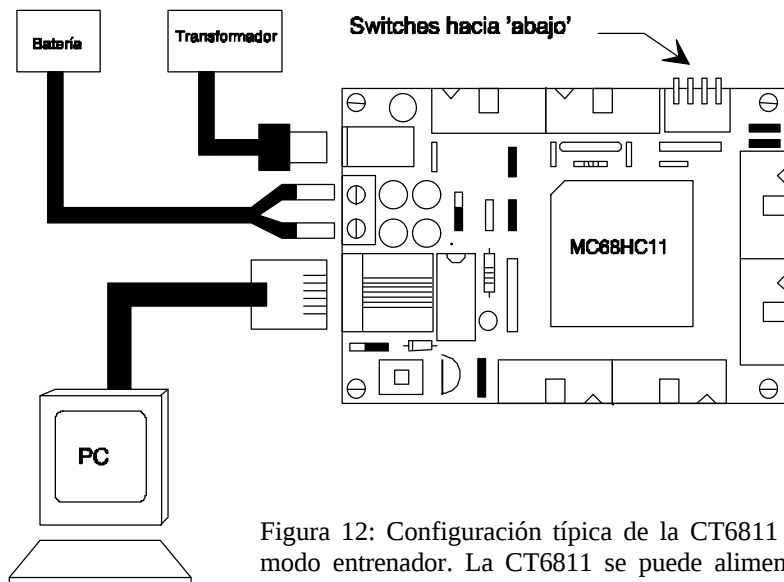


Figura 12: Configuración típica de la CT6811 en modo entrenador. La CT6811 se puede alimentar por medio de un transformador o por baterías.

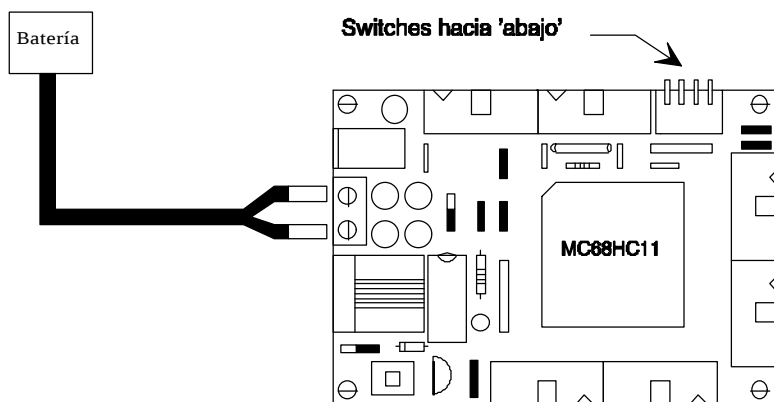


Figura 13: Configuración típica de la CT6811 en modo autónomo.

6.- PUERTOS DE EXPANSION

La tarjeta CT6811 dispone de 6 puertos de expansión. 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.

PUERTO EXPANSION

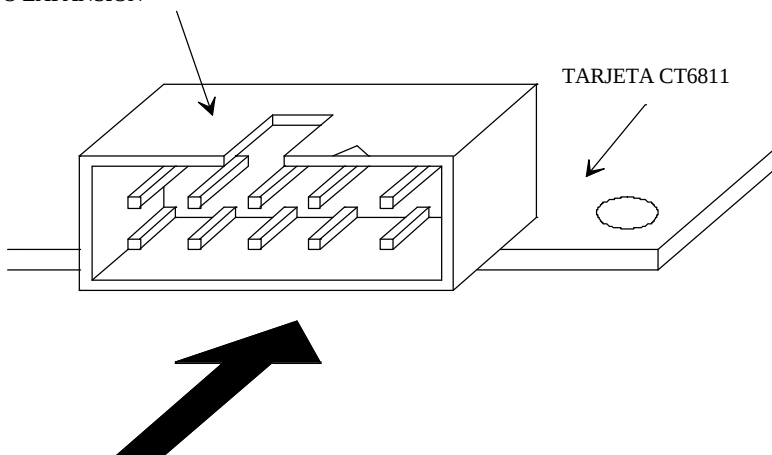


Figura14: Puertos de expansión

En la figura 15 se presentan las señales asociadas a cada puerto. Los puertos están vistos como se indica en la figura 14.

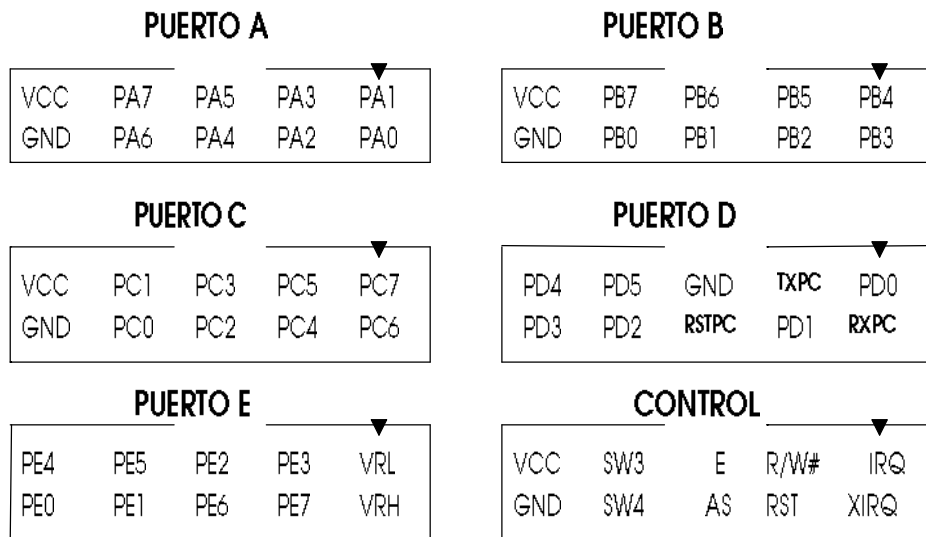


Figura 15: Señales de los puertos de expansión. Los puertos se miran como se indica en la figura 14

Todos los puertos están constituidos por dos filas paralelas de 5 pines. Los puertos se han diseñado para conectar cables planos de tipo bus de 10 bits. Se utilizan este tipo de cables por los siguientes motivos:

- **Fiabilidad:** Se trata de un cable muy fiable, se reduce el ruido y el peligro de cortocircuitos.
- **Coste:** se trata de un cable muy barato y muy común, que se puede comprar en tiendas de electrónica e informática.
- **Múltiples conectores:** 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.

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.

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. Las 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 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.

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.- ALIMENTACION

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.

En la figura 16 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**, mientras que **la parte interior debe ser VCC**. 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.**

En la figura 17 aparecen las distintas vistas de las bornas de alimentación. 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 vez**. O se utiliza el transformador, o se utilizan las bornas.

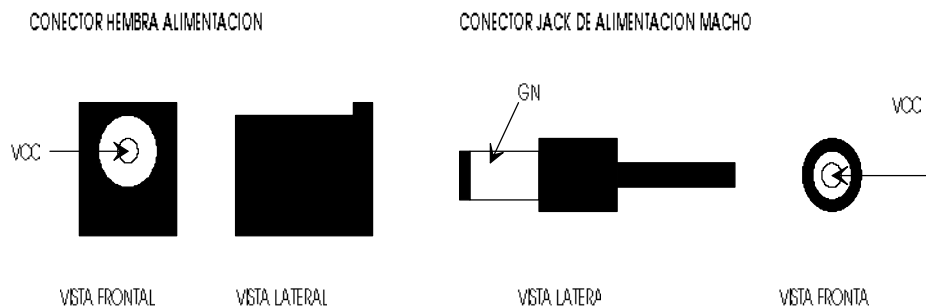


Figura 16: Distintas vistas de los conectores macho y hembra de alimentación mediante un transformador. ¡Cuidado con la polaridad!

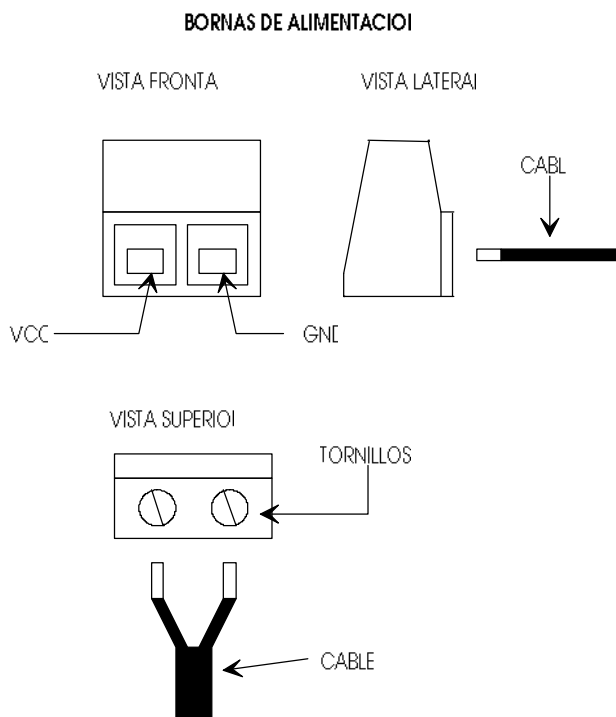


Figura 17: Bornas de alimentación: vistas y polaridad.

8.- CONEXION AL PC

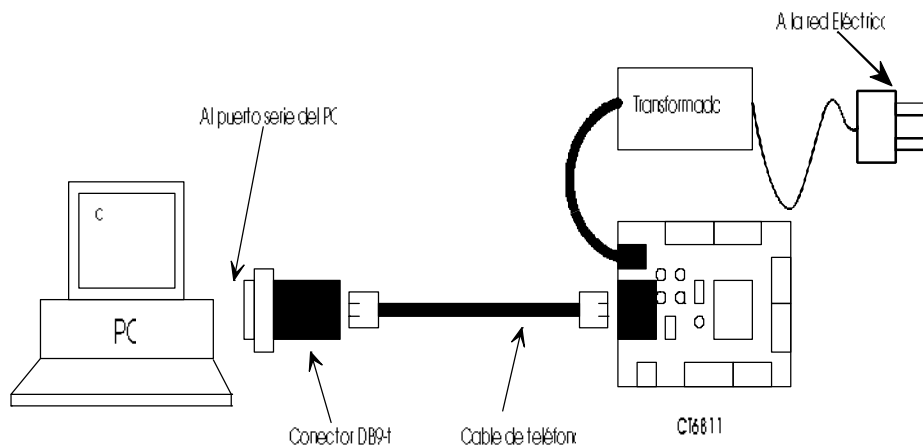


Figura 18: Configuración típica de la CT6811 en modo entrenador y las distintas partes que intervienen en su conexión al pc.

8.1.- Distintas partes que intervienen en la conexión

En la figura 18 aparece la configuración típica de utilización de la CT6811 en modo entrenador. La alimentación se realiza a través de un transformador conectado directamente a la red. La alimentación también se podría tomar de pilas o baterías. 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. La conexión se realiza de esta manera por varios motivos:

- El cable de teléfono es barato y se puede encontrar en muchos sitios.
- El cable de teléfono es muy 'flexible' y se lleva muy bien a cualquier lugar.
- El enganche del cable de teléfono es muy fiable y muy fácil de quitar y poner. El cable puede soportar movimientos bruscos mientras queda sólidamente fijado a la CT6811.
- La conexión al PC es más simple: basta con dejar conectado el conector DB9-TLF en el PC y quitar sólo el cable telefónico.

8.2.- El cable de teléfono

El cable de teléfono empleado es de 4 hilos. No todos los cables sirven. Existen dos tipos de cables, según cómo estén situados los conectores de los extremos. Se han representado en la figura 19.

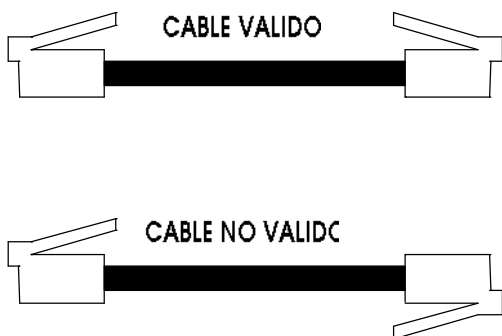


Figura 19: Los dos tipos de cables de teléfono que hay. Sólo uno es válido para trabajar con la CT6811

Para saber si un cable es válido o no hay que situar los dos extremos en paralelo, como se indica en la figura 20. Observando los 4 hilos, que tienen cada uno un color diferente, nos podemos encontrar dos situaciones: que los hilos estén en la misma posición en los dos conectores, o que exista simetría de 'espejo'. Sólo son válidos los cables del segundo tipo. Si según tenemos el cable como se indica en la figura 20 observamos los colores de izquierda a derecha: verde, rojo, amarillo y negro, en el otro conector los debemos observar en orden inverso, también de izquierda a derecha: negro, amarillo, rojo y verde. En la figura 20, se han sustituido los nombres de los colores por números, pues cables diferentes, tienen hilos de diferentes colores.

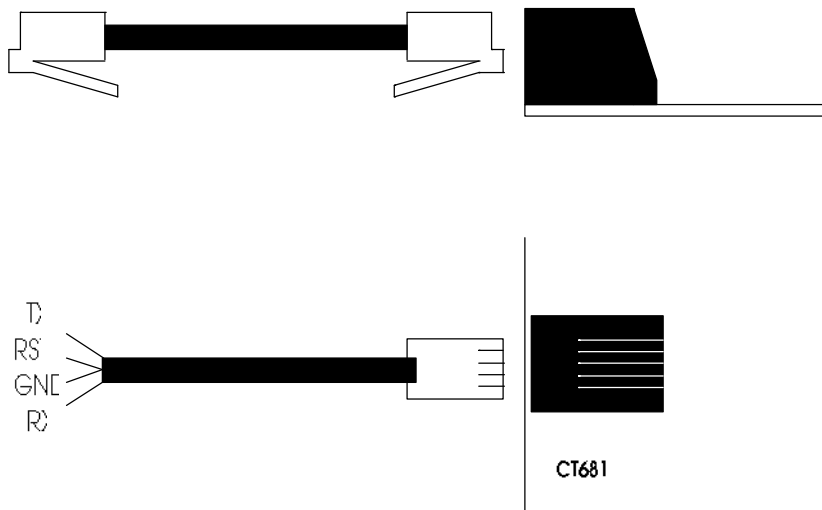


Figura 21: Información transportada por cada hilo del cable de teléfono.

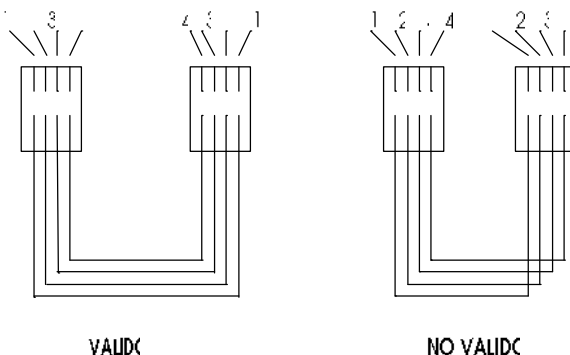
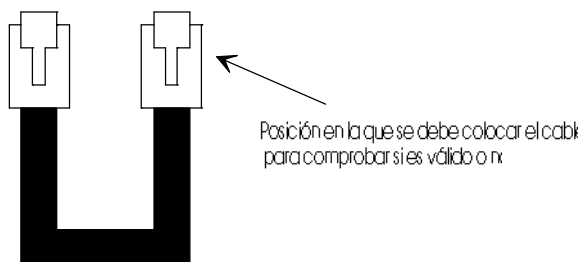


Figura 20: Cómo distinguir un cable válido de uno no válido.

El cable utilizado es de 4 hilos. La información transportada por cada hilo se encuentra en la figura 21. Estas señales son: TX, RX, GND y DTR. La señal DTR es la utilizada para realizar un reset por software de la tarjeta CT6811.

8.3.- El conector DB9-TLF

El conector DB9-TLF es como el dibujado en la figura 22. Por un lado es un conector hembra DB9 y por el otro tiene una hembra de teléfono de 4 vías. Estos conectores se venden desmontados, como se muestra en la figura 23. El cuerpo del conector tiene 4 hilos que se conectan al conector DB9. En la figura 24 se muestra cómo hay que conectar estos cables. Es posible que existan conectores de este tipo con cables de colores diferentes. En la figura 24 se muestra cómo se que conectar los

cables en conectores que sean iguales (mismo fabricante) que el que se proporciona con la CT6811. El conector proporcionado ya se encuentra montado.

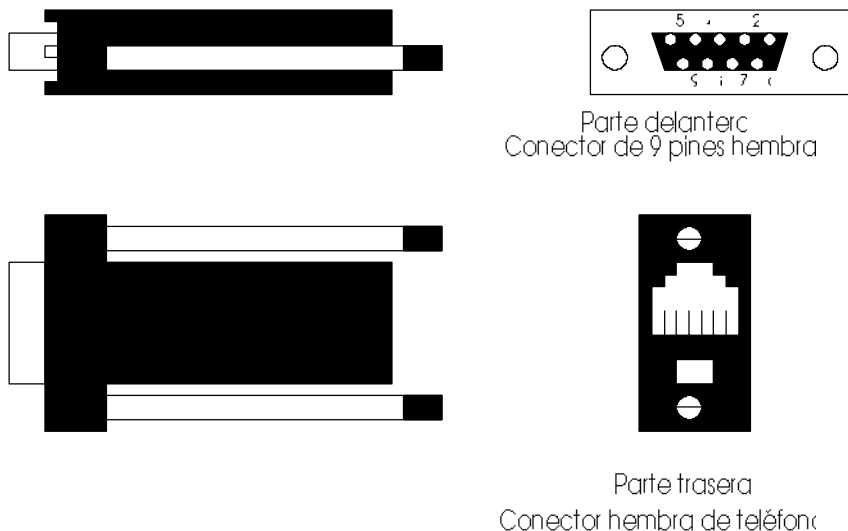


Figura 22: Conector DB9-TLF

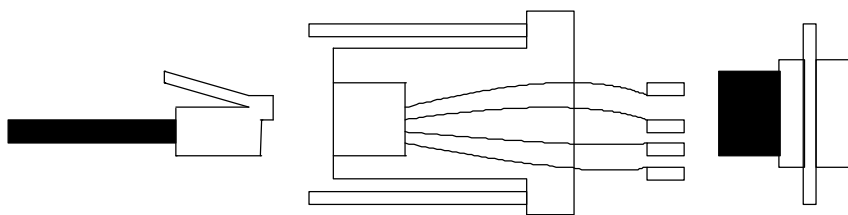
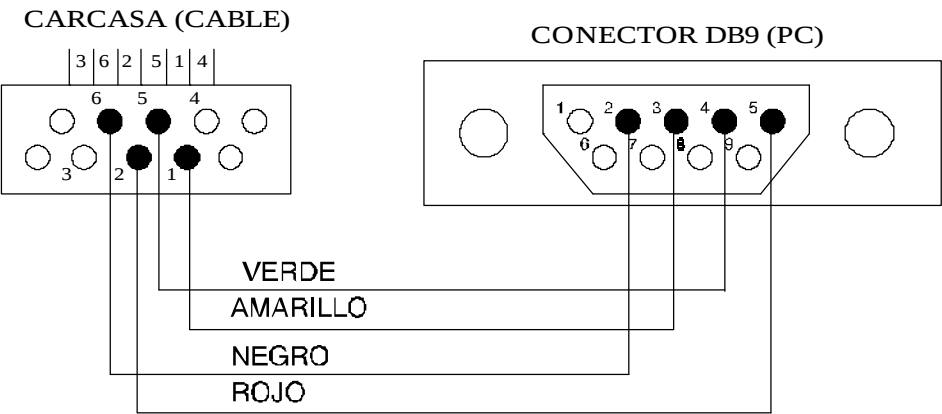


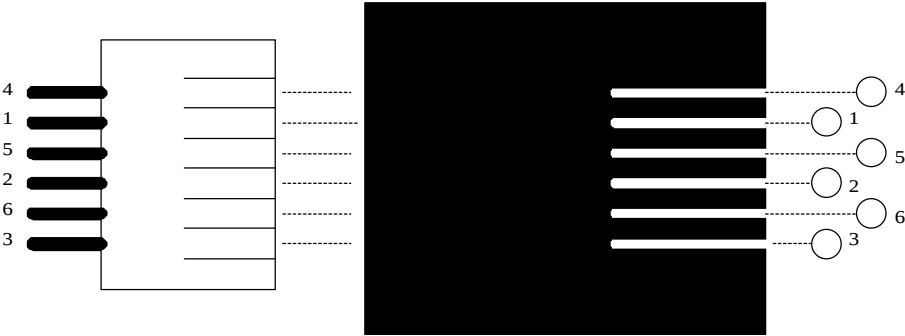
Figura 23: Conector DB9-TLF desmontado

En la página siguiente se muestra la figura 24 que indica el proceso de montaje del conector DB9-TLF. Además se hace un seguimiento de cada cable, desde que sale del conector del PC (DB9) hasta que llega al conector de teléfono situado en la CT6811.

UNIÓN CABLE-PC



UNIÓN CABLE-CT6811



COLOR	PIN DB9	CABLE	FUNCIÓN
NEGRO	2	6	RX pc
AMARILLO	3	1	TX pc
VERDE	4	5	RESET dtr
ROJO	5	2	GND

9.- PRUEBAS DE FUNCIONAMIENTO

9.1.- 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 puntos:

- Coloque los jumpers JP5, JP3, JP8. El jumper JP4 sitúelo en la posición más de la derecha, indicada en la CT6811 como RST. El jumper JP7 sitúelo en la posición más baja, indicada como ON. 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 9.2.

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.

9.2.- Probando la CT6811 en modo entrenador.

En este apartado se va a cargar un programa en la CT6811 desde el PC. Los programas que se van a emplear son: DOWNMCU y CTDIALOG.

Primero debe configurar la CT6811 adecuadamente. Para ello, haga lo siguiente:

- 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: downmcu.exe y ledp.s19. Estos son los mínimos ficheros necesarios para realizar estas pruebas.

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 25 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!).

```
C:\6811\MCB00T2>downmcu ledp -com2  
  
DOWN-MCU. V1.0 (C) GRUPO J&J. Noviembre-1996.  
Envío de programas a la entrenadora  
  
Fichero a enviar: .\ledp.S19  
Puerto serie: COM2  
  
Pulse reset en la entrenadora...  
Transmitiendo:  
>>>>>>>>>>>>>>>>>. . . . .  
. . . . .  
. . . . .  
. . . . .OK!  
Envío correcto  
Tamaño del programa: 22 bytes  
  
C:\6811\MCB00T2>
```

Figura 25: Carga del programa LEDP.S19 en la CT6811 utilizando el programa DOWNMCU.

Los programas descritos implementan el reset software por eso el jumper JP7 debe estar puesto en la posición más baja posible, marcada como '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 de reset de la CT6811. Para que este funcione el jumper JP4 tiene que estar situado a la derecha (RST).

10.- 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 MSDOS, pero se recuerda que también se dispone del software para LINUX, el cual se maneja de forma similar al explicado aquí.

10.1.- 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 archivos 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 del Grupo J&J. 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 del Grupo J&J. Este programa, además de permitir grabar la EEPROM, permite ver el contenido de la memoria del 68HC11, cambiar valores de la memoria, desensamblar ...

10.2.- Un ejemplo completo

En esta sección se va a presentar 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 el led de la CT6811. El programa fuente se puede ver en la figura 26.

El siguiente paso es compilar el programa para obtener el archivo LEDP.S19, que es el fichero ejecutable. Para ello se utiliza el ensamblador AS11 v.1.0.3 de Motorola. En las siguientes figuras se muestra los comandos que hay que teclear y los resultados obtenidos. Los comandos tecleados por el usuario se han escrito en **negrita**. En la figura 27 primero se muestran todos los ficheros necesarios y después se compila el programa ledp.asm. Se obtiene el archivo ledp.s19 que se trata de un archivo de texto y por tanto se puede editar (¡Pero no se debe modificar!).

```

;+-----+
;| LEDP. Grupo J&J                                     MICROBÓTICA |
;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811. |
;| Este programa se debe cargar en la RAM interna del 6811. |
;+-----+
;| Simplemente se enciende y se apaga el led de la tarjeta CT6811. |
;+-----+

                ORG $000

comienzo
                LDAA $1000
                EORA #$40          ; Cambiar de estado el bit PA6
                STAA $1000

dec            LDY #$FFFF          ; Realizar una pausa
                DEY
                CPY #0
                BNE dec

                BRA comienzo      ; Repetir el proceso
                END

Figura 26: Listado del programa LEDP.ASM

```

Ahora hay que enviar el programa LEDP.S19 a la CT6811. En la figura 28 se muestran los comandos. 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.

Los programas cargados en el 68HC11 se encuentran almacenados en la RAM interna del micro. Se pueden cargar los programas todas las veces que se quiera, sin ninguna restricción. Una vez que se tiene el programa depurado, se graba en la EEPROM. En el ejemplo, se trata de un programa muy sencillo y su depuración es muy fácil. A continuación se indica cómo grabar el programa LEDP en la EEPROM.

Antes de nada, hay que modificar el programa fuente para indicar que lo queremos 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 ledp no tiene variables, no hay que preocuparse. En la figura 29 se muestra el listado del programa ledp.asm modificado para ser grabado en la memoria EEPROM. El único cambio que ha habido que hacer es en 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 microcontrolador 68HC11A1. Si utilizamos otro modelo se deberá averiguar esta posición de inicio. El nuevo programa creado se denomina ledpe.asm. La ‘e’ del final se ha puesto para indicar que se trata de un programa para la eeprom. Una vez compilado este programa, de forma similar a como se ha hecho con el programa ledp.asm, obtenemos el fichero ledpe.s19. Este será el fichero que utilizemos para grabar en la EEPROM.

```
C:\6811\CT6811>dir

Volumen en unidad C es GRUPO JJ
Número de serie de volumen es 395D-1CD5
Directorio de C:\6811\CT6811

.           <DIR>      02-16-97   2:13a
..          <DIR>      02-16-97   2:13a
DOWNMCU  EXE      40087 11-30-96 12:11a
CTSERVER S19       608 12-30-96  8:10p
CTDIALOG EXE     76249 12-28-96  2:28a
LEDP     ASM       915 12-16-96  1:20a
AS11     EXE     19584 01-31-92  2:58p
          7 archivo(s)      137443 bytes
          780173312 bytes libres

C:\6811\CT6811>as11 ledp.asm
Freeware assembler ASxx.EXE Ver 1.03.

Number of errors 0

C:\6811\CT6811>dir

Volumen en unidad C es GRUPO JJ
Número de serie de volumen es 395D-1CD5
Directorio de C:\6811\CT6811

.           <DIR>      02-16-97   2:13a
..          <DIR>      02-16-97   2:13a
DOWNMCU  EXE      40087 11-30-96 12:11a
CTSERVER S19       608 12-30-96  8:10p
CTDIALOG EXE     76249 12-28-96  2:28a
LEDP     ASM       915 12-16-96  1:20a
AS11     EXE     19584 01-31-92  2:58p
LEDP     S19        68 02-16-97   2:14a
          8 archivo(s)      137511 bytes
          780238848 bytes libres

C:\6811\CT6811>_
```

Figura 27: Generación del fichero ledp.s19

El programa que se va a emplear es el CTDIALOG. En la figura 30 se muestra cómo se carga este programa. Primero hay que enviar a la CT6811 el programa CTSERVER, y después se ejecuta el programa CTDIALOG.EXE. Además de permitir grabar la EEPROM del 68HC11, con el CTDIALOG podremos ver el contenido de la memoria, modificarla, desensamblar , ejecutar programas...

```
C:\6811\CT6811>downmcu ledp -com2  
  
DOWN-MCU. V1.0 (C) GRUPO J&J. Noviembre-1996.  
Envío de programas a la entrenadora  
  
Fichero a enviar: .\ledp.S19  
Puerto serie: COM2  
  
Pulse reset en la entrenadora...  
Transmitiendo:  
>>>>>>>>>>>>>>>>>>. ....OK!  
Envío correcto  
Tamaño del programa: 22 bytes  
  
C:\6811\CT6811>
```

Figura 28: Cargando el programa LEDP.S19 en la CT6811.

Si se ejecuta el programa CTDIALOG y el servidor CTSERVER no se ha cargado previamente, o se ha cargado mal, aparecerá lo que se indica en la figura 31. En el prompt del CTDIALOG aparecerá un asterisco, indicando que no hay conexión con la tarjeta.

Para grabar el programa ledpe.s19 en la EEPROM hay que utilizar el comando EEPROM seguido del nombre del fichero. Los pasos a seguir se encuentran explicados en la figura 32: Primero se graba el programa en la EEPROM y después se desensambla el contenido de la eeprom para asegurarse de que se ha grabado correctamente. 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, como se ha hecho en el ejemplo de la figura 33. 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 CTDIALOG habrá que salir utilizando el comando quit, volver a cargar el servidor y volver a ejecutar el CTDIALOG.

```

;+-----+
;| LEDPE. Grupo J&J.                                MICROBÓTICA |
;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811.    |
;| Este programa se debe Grabar en la EEPROM del 6811.          |
;+-----+
;| Simplemente se enciende y se apaga el leddela tarjetaCT6811. |
;+-----+
;
;          ORG $B600          ; ¡Memoria EEPROM!
;
comienzo
;          LDAA $1000
;          EORA #$40
;          STAA $1000          ; Cambiar bit PA6 de estado
;
;          LDY #$FFFF          ; Realizar una pausa
dec
;          DEY
;          CPY #0
;          BNE dec
;
;          BRA comienzo      ; Repetir proceso
;
;          END

```

Figura 29: Programa ledp preparado para ser grabado en la EEPROM.

C:\6811\CT6811>downmcu ctserver -com2

DOWN-MCU. V1.0 (C) GRUPO J&J. Noviembre-1996.

Envío de programas a la entrenadora

Fichero a enviar: .\ctserver.S19

Puerto serie: COM2

Pulse reset en la entrenadora...

Transmitiendo:

[illegible]

Envío correcto

Tamaño del programa: 250 bytes

```
C:\6811\CT6811>ctdialog -com2
```

CTDIALOG Versión 1.0 (C) GRUPO J&J. Diciembre 1996

Teclee HELP para obtener ayuda

Puerto actual: COM2

```
Estableciendo conexión con tarjeta..... conexión establecida
```

>

Figura 30: Carga del programa CTDIALOG.

```
C:\6811\CT6811>ctdialog -com2
```

```
CTDIALOG Versión 1.0 (C) GRUPO J&J. Diciembre 1996
```

```
Teclee HELP para obtener ayuda
```

```
Puerto actual: COM1
```

```
Estableciendo conexión con tarjeta..... CONEXION NO ESTABLECIDA
```

```
*>_
```

Figura 31: El CTDIALOG no se ha conectado con el servidor.

```
>eeprom ledpe
Fichero a grabar en memoria EEPROM: LEDPE.S19
Transmitiendo: -----»
Grabación terminada
Número bytes grabados: 22
```

```
>dasm b600
```

```
B600 LDAA 1000
B603 EORA #40
B605 STAA 1000
B608 LDY #FFFF
B60C DEY
B60E CPY #0000
B612 BNE B60C
B614 BRA B600
B616 STX FFFF
B619 STX FFFF
B61C STX FFFF
B61F STX FFFF
B622 STX FFFF
B625 STX FFFF
B628 STX FFFF
B62B STX FFFF
B62E STX FFFF
B631 STX FFFF
B634 STX FFFF
```

```
>g b600
Conexión perdida
```

```
*>quit
Programa terminado
```

```
C:\6811\CT6811>_
```

Figura 32: Grabación en eeprom del programa ledpe, desensamblado de la eeprom y salto a la eeprom.

PARTE II:

ENTORNO DE DESARROLLO Y PROGRAMAS

EJEMPLO

1.- INTRODUCCION

En esta parte se van a mostrar ejemplos de programas para la tarjeta CT6811. El objetivo no es el de enseñar a programar el microcontrolador 68HC11, sino el de

aprender a manejar el software básico para programar la tarjeta CT6811, así como mostrar en detalle una serie de ejemplos de programación. Se da por hecho que el lector sabe algo de programación en ensamblador. Si no es así, no importa. A partir de los ejemplos el lector puede aprender bastante, aunque debe recurrir a otros textos para profundizar en la programación del 68HC11.

2.- CONTROL DE LA CT6811 DESDE EL PC: CTDIALOG

Para tener un primer contacto con la tarjeta CT6811 se va a presentar un programa, el **CTDIALOG**, con el que vamos a poder ‘dialogar’ con el 68HC11. Este programa nos permite acceder a todos los recursos del 68HC11 desde el PC: memoria RAM, registros de control, memoria EEPROM... Este programa servirá más adelante para comprobar que todos los periféricos conectados a la CT6811 funcionan adecuadamente: movimiento de los motores, lectura de los sensores, encendido y apagado de LED’s conectados...

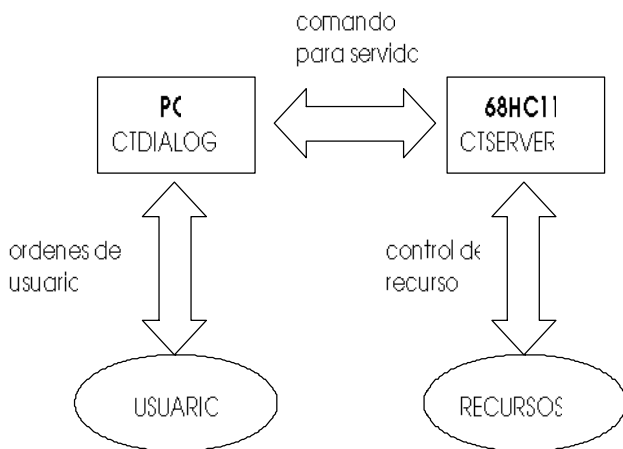


Figura 33: Control de los recursos del 68HC11 utilizando el programa CTDIALOG.

Antes de pasar a ejecutar el CTDIALOG, se va explicar un poco por encima cómo funciona. El programa se divide en dos partes: una parte se ejecuta en el 68HC11, la otra parte se ejecuta en el PC. El programa que se ejecuta en el 68HC11 se denomina CTSERVER. Se trata de un ‘**servidor de información**’. Según los comandos que le envíe el PC por el puerto serie, el CTSERVER actuará en consecuencia. El programa que se ejecuta en el PC actúa como interfaz de usuario, presentando la información de una forma inteligible para el usuario y convirtiendo las órdenes del usuario a comandos que entiende el CTSERVER. El esquema se muestra en la figura 33.

Un poco más adelante, se va a mostrar con ejemplos cómo se puede encender el led de la tarjeta CT6811 utilizando el programa CTDIALOG. El usuario tecleará un comando en el PC. Este comando, si es correcto, será procesado por el CTDIALOG y se transformará en un comando que el CTSERVER entienda. El programa CTSERVER convertirá el comando en las instrucciones necesarias para encender el LED de la CT6811.

2.1.- Ejecutando el programa CTDIALOG

La ejecución del programa CTDIALOG para ‘dialogar’ con la tarjeta CT6811 se divide en dos partes. Primero es necesario ejecutar el programa servidor CTSERVER en la tarjeta CT6811. Después habrá que ejecutar el programa CTDIALOG en el PC. En la figura 34 se muestra un ejemplo de ejecución del CTDIALOG para el **COM2**. Para utilizar el puerto serie COM1 sólo habrá que sustituir el parámetro -com2 por el parámetro -com1. Para enviar cualquier programa a la CT6811 y que se ejecute, se utilizará el programa DOWNMCU. Este programa toma como argumentos el nombre del programa a enviar a la CT6811 y el puerto serie en el que está conectada la CT6811.

En la figura 34 se han representado en **negrita** los comandos tecleados por el usuario. Si todo ha funcionado correctamente, al ejecutar el programa CTDIALOG deberá aparecer un prompt ‘>’ indicando que el programa está esperando las órdenes del usuario. Si no ha conseguido enviar correctamente el programa CTSERVER a la tarjeta CT6811 compruebe los siguientes puntos:

- Que la tarjeta CT6811 está correctamente configurada en **modo entrenador**.
- Que la tarjeta CT6811 está correctamente conectada a un puerto serie del PC (Com1 ó com2)
- Que la tarjeta CT6811 está correctamente alimentada
- Que el puerto especificado en el programa DOWNMCU coincide realmente con el puerto al que está conectada la tarjeta CT6811

[illegible]

Figura 34: Ejecución del programa CTDIALOG.

2.2.- Algunos comandos del programa CTDIALOG

En la figura 35 se muestran un ejemplo de utilización de alguno de los comandos del CTDIALOG. El comando **ckc** (ChecK Conexion) comprueba que la conexión con el programa CTSERVER es correcta y que el sistema se encuentra listo para ejecutar comandos. El siguiente comando, **MD**, sirve para volcar bloques de datos de la memoria. En el ejemplo se ha empleado **MD 0 1**. El argumento 0 indica que queremos volcar bloques a partir de la dirección 0 del 68HC11. El argumento 1 indica que queremos volcar 1 bloque. Cada bloque está constituido por 16 bytes. Por tanto, lo que conseguimos con el comando **MD 0 1** es ver el contenido de las posiciones de memoria 0 hasta la \$0F del 68HC11. Con el comando **MS 0 FA** lo que hacemos es decirle al 68HC11 que almacene el valor hexadecimal \$FA en la posición de memoria 0. Para comprobar que efectivamente así ha sido, volvemos a ver el contenido de la posición 0. Efectivamente comprobamos que se ha almacenado el valor indicado por nosotros. Hay que tener cuidado al utilizar el comando **MS** con la memoria interna,

puesto que el programa CTSERVER se encuentra en la RAM interna del 68HC11, entre las posiciones de memoria 0 y \$FF. Si almacenamos un valor en alguna de estas posiciones, estamos modificando el programa CTSERVER, lo que puede ocasionar que no se ejecute correctamente y se pierda la conexión.

```
>ckc
Puerto actual: COM2
Estado de la conexión: ESTABLECIDA

>md 0 1

      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000  20 05 00 03 00 0B 00 CE 10 00 18 CE FF FF 18 09  ....

1 bloque volcado

>ms 0 fa
>md 0 1

      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000  FA 05 00 03 00 0B 00 CE 10 00 18 CE FF FF 18 09  ....

1 bloque volcado

>_
```

Figura 35: Ejemplo de uso de algunos comandos del CTDIALOG

2.3.- Acceso al LED de la CT6811 desde el programa CTDIALOG

Una de las características más atractivas del programa CTDIALOG es que nos permite acceder a todos los puertos de E/S del 68HC11, con lo que podremos acceder a los periféricos que se encuentren conectados a ellos.

En la tarjeta CT6811, el LED de pruebas se encuentra conectado el bit 6 del puerto A del 68HC11. Para encender el LED simplemente habrá que activar este bit desde el CTDIALOG. El puerto A se encuentra en la posición de memoria **\$1000**. Una escritura en esta posición de memoria, provoca que se activen los bits correspondientes de salida del puerto A. Una lectura permite leer el estado de los bits de entrada del puerto A. El bit 6 del puerto A es siempre de Salida. En la figura 4 se muestra un ejemplo de encendido y apagado del LED. Para encender mandamos el valor \$40 al puerto A (que se encuentra en la dirección \$1000). Para apagarlo mandamos el valor

\$00. El se encenderá siempre y cuando el valor enviado tenga un '1' en el bit 6. Por ejemplo, los valores \$FF, \$C0, \$72... también encenderá el led.

En la figura 36 después de encender el LED se mira la dirección 1000 para comprobar el valor almacenado. Después de apagarlo se comprueba que se ha almacenado el valor 0.

```
>ms 1000 40
>md 1000 1

      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
1000  40 00 03 00 00 FE 00 00 03 00 00 00 00 0D B9  ....

1 bloque volcado

>ms 1000 0
>md 1000 1

      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
1000  00 00 03 00 00 FE 00 00 03 00 00 00 00 18 38  ....

1 bloque volcado

>_
```

Figura 36: Encendido y apagado del LED.

3.- PROGRAMACION DE LA TARJETA CT6811

Hasta ahora se ha mostrado cómo acceder a los recursos de la CT6811 desde el PC utilizando el programa CTDIALOG. Ahora se van a presentar ejemplos de programas en ensamblador del 68HC11 que controlan directamente los recursos de la CT6811.

3.1.- Ficheros fuente y ficheros ejecutables

Los programas que se van a realizar están en lenguaje ensamblador del 68HC11. La extensión que se utilizará para estos archivos será **.ASM**. Estos ficheros se compilarán utilizando el ensamblador Freeware AS11 de Motorola. Una vez compilado el programa, se tendrá un archivo con extensión **.S19**. Estos serán los que se envíen a la tarjeta CT6811.

Los programas se realizan en el PC con un editor de texto normal. Se compilan y se envían al 68HC11 para ser ejecutados. Los programas se cargarán en los 256 bytes de memoria **RAM interna** del 68HC11. Una vez que los programas funcionan correctamente y han sido depurados, se graban en la memoria **EEPROM interna** del 68HC11. A partir de entonces, la tarjeta ya no depende del PC y se convierte en un **sistema autónomo**.

3.2.- Un primer programa ejemplo: Encender el LED de la tarjeta CT6811

En la sección 2.3 se ha encendido el LED de la CT6811 desde el PC utilizando el programa CTDIALOG. Ahora se encenderá desde un programa en ensamblador. El programa es el más simple de todos. Simplemente habrá que almacenar el valor \$40 en la posición de memoria \$1000, que se corresponde con el puerto A. El programa se presenta en la figura 37.

```

;+-----+
;| LEDON.  Grupo J&J                                MICROBÓTICA. |
;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811.      |
;| Este programa se debe cargar en la RAM interna del 6811.      |
;|                                                                  |
;| Simplemente se enciende el led de la tarjeta CT6811.          |
;+-----+

ORG $000

LDAA #$40    ; Almacenar en el acumulador A el valor $40
STAA $1000   ; Enviar el valor $40 a la posición de memoria $1000

inf BRA inf    ; Bucle infinito

END

```

Figura 37: Programa LEDON.ASM que enciende el led de la tarjeta CT6811

Si el lector no ha programado nunca en lenguaje ensamblador el programa le puede resultar un poco ‘extraño’. Sin embargo es muy simple y se va a ir explicando poco a poco. Las primeras ocho líneas son comentarios. Los comentarios se realizan utilizando el carácter de punto y coma ‘;’. Todo lo que se encuentre a la derecha del signo ‘;’ es ignorado por el ensamblador. Si se trabaja en LINUX se utilizará el carácter asterisco ‘*’ para indicar línea de comentario.

La primera instrucción que nos encontramos es **ORG \$0000**. Realmente, no es una instrucción del 68HC11, sino que se trata de una **directiva** del ensamblador que indica

que se quiere situar el programa a partir de la dirección \$0000 en adelante. Se utiliza esta dirección porque coincide con el comienzo de la memoria RAM interna del 68HC11. El carácter '\$' sirve para indicar que el número que viene a continuación se encuentra en **hexadecimal**.

La siguiente instrucción es "LDAA #\$40". Esta instrucción almacena el valor \$40 en el registro A del 68HC11. Un registro se puede ver como una 'variable' del 68HC11. Delante del número \$40 se ve el signo '#'. Este carácter indica que el número \$40 hace referencia al número \$40 y **no a la posición de memoria \$40**. Hay que tener mucho cuidado con esto porque suele provocar bastantes errores en los programas. La instrucción LDAA \$40 carga en el acumulador A el contenido de la dirección de memoria \$40. La instrucción LDAA #\$40 carga en el acumulador A el número \$40 en hexadecimal, recuerde que lleva el carácter '\$' delante.

La instrucción **STAA \$1000** almacena el valor contenido en el acumulador en la posición de memoria \$1000, que es donde se encuentra el PUERTO A. Al ejecutarse esta instrucción se activa el bit 6 del puerto A y el LED se enciende.

La última instrucción se trata de un bucle infinito. Es una instrucción de salto que se llama a sí misma. Esta instrucción se ha incluido para tener 'controlado' el 68HC11. Si no la colocamos, el 68HC11 no sabe que el programa a finalizado y continúa ejecutando instrucciones. ¿Qué instrucciones ejecuta?, pues no se sabe, porque el contenido de la memoria RAM puede ser cualquiera. Para evitar que se ejecuten instrucciones 'peligrosas' incluimos un bucle infinito para finalizar el programa.

```
C:\6811\CT6811>as11 ledon.asm
Freeware assembler ASxx.EXE Ver 1.03.
```

```
Number of errors 0
```

```
C:\6811\CT6811>downmcu ledon -com2
```

```
DOWN-MCU. V1.0 (C) GRUPO J&J. Noviembre-1996.
Envío de programas a la entrenadora
```

```
Fichero a enviar: .\ledon.S19
Puerto serie: COM2
```

```
Pulse reset en la entrenadora...
```

```
Transmitiendo:
```

```
>>>>>>.....
.....
.....OK!
```

```
Envío correcto
```

```
Tamaño del programa: 7 bytes
```

```
C:\6811\CT6811>_
```

Figura 38: Ensamblado del programa LEDON.ASM y envío a la tarjeta CT6811

Ya se tiene el programa **LEDON.ASM**. Ahora hay que enviarlo a la CT6811 para ejecutarlo. En la figura 38 se muestran todos los pasos a seguir. Primero se compila el programa utilizando el ensamblador AS11 de Motorola. Después se envía utilizando el programa DOWNMCU.

Para que el programa funcione correctamente asegúrese de que la tarjeta CT6811 se encuentra configurada en modo entrenador y que el jumper JP3 está colocado para dejar habilitado el LED.

En cuanto se termina de cargar el programa, puede verse cómo se enciende en led. Si lo quiere apagar, apriete el pulsador de reset.

3.3.- Programa del LED parpadeante

En la figura 39 se presenta un programa un poco más complicado. El programa hace parpadear el led de la CT6811 a una frecuencia fija.

```

;+-----+
;| LEDP. |
;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811.
;| Este programa se debe cargar en la RAM interna del 6811.
;|
;| Simplemente se enciende y se apaga el led de la tarjeta CT6811.
;+-----+

                ORG $000

comienzo
                LDAA $1000
                EORA #$40          ; Cambiar de estado el bit PA6
                STAA $1000

dec             LDY #$FFFF        ; Realizar una pausa
                DEY
                CPY #0
                BNE dec

                BRA comienzo     ; Repetir el proceso

                END

```

Figura 40: Programa LEDP.ASM.

Las tres primeras instrucciones son las encargadas de cambiar de estado el bit 6 del puerto A, lo que provocará que el led cambie de estado (de encendido a apagado o de apagado a encendido). Para hacer eso se carga en el acumulador A el valor que hay en el puerto, se aplica la función lógica XOR al bit 6 (instrucción EORA= Exclusive OR acumulador A) y se deposita el nuevo valor en el puerto. A continuación hay que realizar una pausa para que el led permanezca en el mismo estado durante el tiempo suficiente para que el ojo humano lo pueda apreciar. Con **LDY #\$FFFF** se carga el máximo valor posible en el registro de 16 bits Y. Este registro va decrementando en una unidad. Cuando se hace igual a cero se termina la pausa y se salta al comienzo del programa. Si el registro Y es distinto de cero, se continua decrementando. En este programa aparecen varias instrucciones nuevas: CPY, BNE, LDY y EORA. La instrucción **CPY #0** realiza una comparación del registro Y con el valor 0. La instrucción **BNE** comienzo realiza una bifurcación a la dirección indicada si el resultado de la última comparación no se ha cumplido (BNE = Branch if Not Equal). La instrucción LDY sirve para cargar valores en el registro Y.

El programa se carga de la misma manera que el programa LEDON. El lector debería compilarlo y ejecutarlo.

Ejercicio 1: Se propone que el lector modifique el programa LEDP.ASM para que el led parpadee a mayor frecuencia. ¿Qué ocurre si la frecuencia de parpadeo es muy elevada?.

Ejercicio 2: En el programa LEDP.ASM, el led está igual de tiempo encendido que apagado. Modificar este programa para que esté más tiempo apagado que encendido.

4.- GRABACION DE PROGRAMAS EN LA EEPROM USANDO EL CTDIALOG

El 68HC11A1 dispone de una memoria EEPROM interna de 512 bytes. Se encuentra situada a partir de la dirección **\$b600**. En esta sección se va a modificar el programa LEDP.ASM para que se pueda grabar en la memoria EEPROM. Si se dispone del 68HC11E2 la eeprom empezará en la dirección \$F800, por lo tanto al aplicar los ejemplos de este apartado habrá que cambiar el valor \$b600 por el \$F800. Más información sobre la CT6811 con el microcontrolador 68HC11E2 se encuentra en el apéndice E.

En la figura 40 se muestra el programa LEDPE.ASM que ya ha sido modificado para grabarse en la EEPROM. El único cambio introducido ha sido que se ha sustituido **ORG \$0000 por ORG \$B600**. Es decir, se quiere que el programa comience a partir de la dirección \$b600 en vez de a partir de la dirección \$0000. También se ha modificado el valor del retardo de espera para que el led se encienda y se apague a mayor velocidad.

```

;+-----+
;|          LEDPE.          (C)MICROBÓTICA.          Noviembre 1996          |
;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811.          |
;| Este programa se debe grabar en la EEPROM del 6811.          |
;|                               |
;| Simplemente se enciende y se apaga el led de la tarjeta CT6811.          |
;+-----+
;
;          ORG $B600          ; ¡Memoria EEPROM!
comienzo
;          LDAA $1000
;          EORA #$40
;          STAA $1000          ; Cambiar bit PA6 de estado
;
;          LDY #$8000          ; Realizar una pausa
dec
;          DEY
;          CPY #0
;          BNE dec
;
;          BRA comienzo          ; Repetir proceso
;
;          END

```

Figura 40: Programa LEDPE.ASM

Para grabar este programa en la memoria EEPROM se va a utilizar el programa CTDIALOG. En la figura 41 se muestran todos los pasos a seguir. Primero se compila el programa LEDPE.ASM. Después se ejecuta el programa CTDIALOG, cargando previamente el programa servidor en el 68HC11. Finalmente se graba el programa utilizando el comando **EEPROM LEDPE** del CTDIALOG.

Para probar el programa hacemos una instrucción de salto (G B600) a la eeprom para comprobar que el programa se ejecuta correctamente. El led comenzará a parpadear y en el CTDIALOG nos presentará un mensaje indicando que se ha perdido la conexión puesto que ya no se está ejecutando el programa servidor sino el programa que acabamos de grabar en la EEPROM.

Utilizando el comando **DASM b600** podemos desensamblar la memoria eeprom y ver si el programa se ha grabado correctamente. Para poder ejecutar este comando hay que volver a restablecer la conexión. Para ello hay que salir del programa, volver a cargar el servidor y ejecutar el CTDIALOG.

[illegible]

Figura 41: Grabación del programa LEDPE.ASM en la EEPROM

5.- 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 prueba 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.

5.1.- Programa de ejemplo scihola.asm

A continuación se presenta un pequeño programa de prueba de comunicaciones serie que 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 a que se reciba 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** para enviar una cadena completa.

Este programa se presenta como ejemplo de programa en el 68HC11 que sirve para interactuar con un usuario. Más adelante se utilizarán las comunicaciones serie para mostrar al usuario un menú de opciones desde el cual podrá controlar la CT6811.

Este programa se transmitirá a la CT6811 utilizando el programa de comunicaciones **MCBOOT**. Una vez cargado el programa, seleccione el puerto en el que tiene conectada la CT6811. Por defecto se utilizará el **COM1**. Pulsado la **tecla F4** se puede cambiar a otro puerto serie. **Asegúrese de que NO está activado el reset software**. Para ello **pulse F7** hasta que aparezca en la **pantalla DTR OFF**. **Siempre que esté DTR ON, la CT6811 estará 'reseteada' y no se podrán cargar programas**. Salvo que se anule el reset software colocando el jumper JP7 arriba, es decir posición off.

Para cargar un programa pulse la **tecla F1**. Introduzca el nombre del fichero .S19 a cargar. En este caso será el programa scihola. El programa sacará un mensaje diciendo que pulse el botón de reset. Apriete el pulsador de la tarjeta CT6811 para comenzar a cargar el programa. Una vez cargado, cada vez que pulse una tecla, se enviará por el puerto serie hasta la CT6811 y ésta responderá enviando la cadena "hola como estas..." que se visualizará en la pantalla del PC.

```

;+-----+
;|  SCIHOLA.    GRUPO J&J.                                MICROBÓTICA  |
;+-----+
;|  Programa ejemplo para ser ejecutado en la tarjeta CT6811.
;|  Este programa se debe cargar en la RAM internadel 6811.
;|  Se envía una cadena por el puerto serie al pulsarse una tecla.
;+-----+

; Registros del SCI

SCSR    equ    $2E
SCDR    equ    $2F

        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

;+-----+
;|  Rutina para leer un carácter del puerto serie (SCI)
;|  La rutina espera hasta que llegue algún carácter
;|  ENTRADAS: Ninguna.
;|  SALIDAS: El acumulador A contiene el carácter recibido
;+-----+
leer_car BRCLR SCSR,X $10 leer_car ; Esperar a que llegue un carácter
        LDAA  SCDR,X
        RTS

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

;+-----+
;|  Enviar una cadena de caracteres por el puerto serie.
;|  La cadena debe terminar con el carácter 0
;|  ENTRADAS: Registro Y contiene dirección cadena a enviar
;|  SALIDAS: El acumulador A contiene el carácter recibido
;+-----+
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

;+-----+

```

```

;| DATOS |
;+-----+
hola      FCC "Hola como estas.."
          FCB 0

          END

```

5.2.- 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 realizará una u otra opción. Las opciones de ejemplo que se han implementado son: cambiar el estado del led y volver a imprimir el menú. Con este programa modelo, el usuario lo puede ir ampliando para adaptarlo a sus necesidades.

```

;+-----+
;| MENÚ.          GRUPO J&J.                                MICROBÓTICA |
;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811.
;| Este programa se debe cargar en la RAM interna del 6811.
;| Ejemplo de cómo manejar un menú de opciones para programas interac-
;| tivos con el usuario.
;+-----+

```

```

CR      equ 13      ; Retorno de carro
LF      equ 10      ; Avance de línea

```

; Registros del SCI

```

SCSR    equ $2E
SCDR    equ $2F

```

```

LDX #$1000      ; Para acceder a registros del SCI

```

```

bucle   LDY #menu
        BSR send_cad      ; Sacar menú
wait    BSR leer_car      ; Leer tecla
        CMPA #'1'
        BEQ opcion1      ; Tecla '1'--> Opción 1 del menú.
        CMPA #'2'
        BEQ opcion2      ; Tecla '2'--> Opción 2 del menú.
        BRA wait

```

```

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

```

```

opcion2 BRA bucle      ; Volver a sacar el menú

```

```

;+-----+
;| Rutina par leer un carácter del puerto serie (SCI)
;| La rutina espera hasta que llegue algún carácter
;| ENTRADAS: Ninguna.
;| SALIDAS: El acumulador A contiene el carácter recibido
;+-----+
leer_car BRCLR SCSR,X $10 leer_car    ; Esperar a que llegue un carácter
        LDAA SCDR,X
        RTS

```

```

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

```

```

;+-----+
;| Enviar una cadena de caracteres por el puerto serie.
;| La cadena debe terminar con el carácter 0
;| ENTRADAS: Registro Y contiene dirección cadena a enviar
;| SALIDAS: El acumulador A contiene el carácter recibido
;+-----+
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

```

```

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

        END

```


6. EL ENTORNO DE DESARROLLO TOWERBOOT (twb2.exe)

El Towerboot es un entorno de desarrollo que engloba varios programas para facilitar el desarrollo de aplicaciones diversas con microcontroladores u otros dispositivos, como FPGA's.

Sus opciones principales están distribuidas en cuatro menús: Mcu, Fpga, Utl y Tx a los que se puede acceder bien con las teclas F's que aparecen indicadas en la pantalla principal, o bien con ALT + la mayúscula resaltada en el nombre de cada menú. Se puede volver hacia atrás pulsando ESCAPE y moverse por los distintos menús en vertical y horizontal con los cursores. El programa arranca directamente en modo terminal, recibiendo información por el puerto serie.

Se va a analizar el contenido de cada menú en la versión actual Towerboot 2.1:

1.- Menú MCU.

Incorpora tres opciones que están implementadas mediante cuatro programas desarrollados íntegramente por el Grupo J&J en lenguaje C, para el sistema operativo MS-DOS. Pueden ser ejecutados directamente en la línea de comandos desde fuera del Towerboot.

DOWNMCU: Como se ha explicado antes, permite cargar programas en el formato .s19 de Motorola en la RAM interna del microcontrolador. Dispone de un menú de ayuda que aparece con el comando HELP, y que corresponde a las opciones posibles cuando se ejecuta desde fuera del Towerboot ya que, en caso contrario, este último pide el nombre del fichero y del puerto de comunicaciones.

Nombre fichero: cuando desde el Towerboot se pide el nombre del fichero a cargar, cabe la posibilidad de pulsar ENTER para ver un listado de archivos .S19 en el directorio actual.

Puerto COM: la opción pulsada se mantiene en las siguientes cargas por lo que sólo es necesario pulsar ENTER si no se desea cambiar el puerto.

CTDIALOG: Como ya ha sido explicado, permite al usuario ver los contenidos de la memoria del microcontrolador, modificarlos, acceder a los registros de control, grabar la memoria EEPROM interna, cargar programas en la RAM externa, desensamblar código,... en definitiva, 'dialogar' con la CT6811. También posee una ayuda propia que aparece con el comando HELP.

Para que el CTDIALOG funcione correctamente, es necesario haber cargado previamente en la CT6811 el programa **CTSERVER** que es un pequeño servidor al

que se le solicitan servicios por medio del puerto serie. Cuando se ejecuta la opción CTDIALOG desde el Towerboot, éste carga automáticamente el servidor en el microcontrolador.

El conjunto de programas CTSERVER, CTDIALOG y DOWNMCU permiten manejar al 100% todos los recursos de la tarjeta CT6811.

CT293: Programa de aplicación enfocado hacia el control de microbots. Permite visualizar y cambiar el estado de los bits del puerto A, así como del estado de dos motores, que bien pudieran estar conectados a la CT6811.

2.- Menú FPGA.

Una ampliación de la CT6811 es posible mediante la incorporación de otra 'planta', constituyendo así la estructura de 'torre' para la que fue diseñada la tarjeta y que le dota de gran versatilidad. Se pueden añadir muchas plantas que bien pueden ser una etapa de potencia, una planta de memorias o incorporar otros dispositivos como una FPGA.

Para este último caso, el Grupo J&J ha desarrollado varios programas con el fin de cargar software en la FPGA, al igual que en el caso del microcontrolador, y son DOWNLCA, DOWNSPI, RBT2ASM y un programa de test. En la versión actual del Towerboot, estos programas no están incluidos pero si alguien está interesado en ellos, puede ponerse en contacto con nosotros.

3.- Menú UTL.

En este menú presenta varias opciones :

•**Editor:** Llamada al editor del MSDOS para el caso en que se quiera modificar un programa fuente *.asm, o bien editar cualquier otro fichero de texto. Se sale de este editor pulsando ALT-X.

•**Compilador:** el compilador AS11 (Freeware assembler Asxx.exe ver 1.03) para generar los ficheros *.s19 a partir del código en ensamblador *.asm.

•**Archivos:** presenta un listado de los archivos del tipo indicado por el usuario que se encuentren en el directorio actual. Se puede avanzar y retroceder en la lista con los cursores.

•**Canales:** hemos llamado canales a las distintas vías de ejecución de archivos .bat que ofrece el Towerboot. Pueden resultar muy útiles cuando se están llevando a cabo

las mismas acciones continuamente, ya que escribiendo en los archivos "canal1.bat", "canal2.bat" y "canal3.bat" las órdenes deseadas, sólo es necesario acudir a la opción "canales" del Towerboot para ejecutarlas. También da la posibilidad de utilizar otros programas (como el editor EDIT del MS-DOS, para el que lo prefiera al QQ), incluyendo una llamada a los mismos en alguno de los archivos .bat.

4.- Menú TX.

Éste es un menú de configuración, en el que se pueden modificar las siguientes opciones:

- Puerto de comunicaciones : com 1, 2, 3 o 4.
- Baudios : 1200, 7680 o 9600.
- DTR: on / off.

Los parámetros seleccionados aparecen indicados en la línea superior de la pantalla principal del Towerboot. Para acceder a las distintas opciones de este menú, además de las teclas F's y ALT-T, cabe la posibilidad de modificar el estado del DTR con la tecla directa indicada al respecto, para mayor comodidad.

APENDICE A: 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, como se muestra en la figura 42:

- **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.

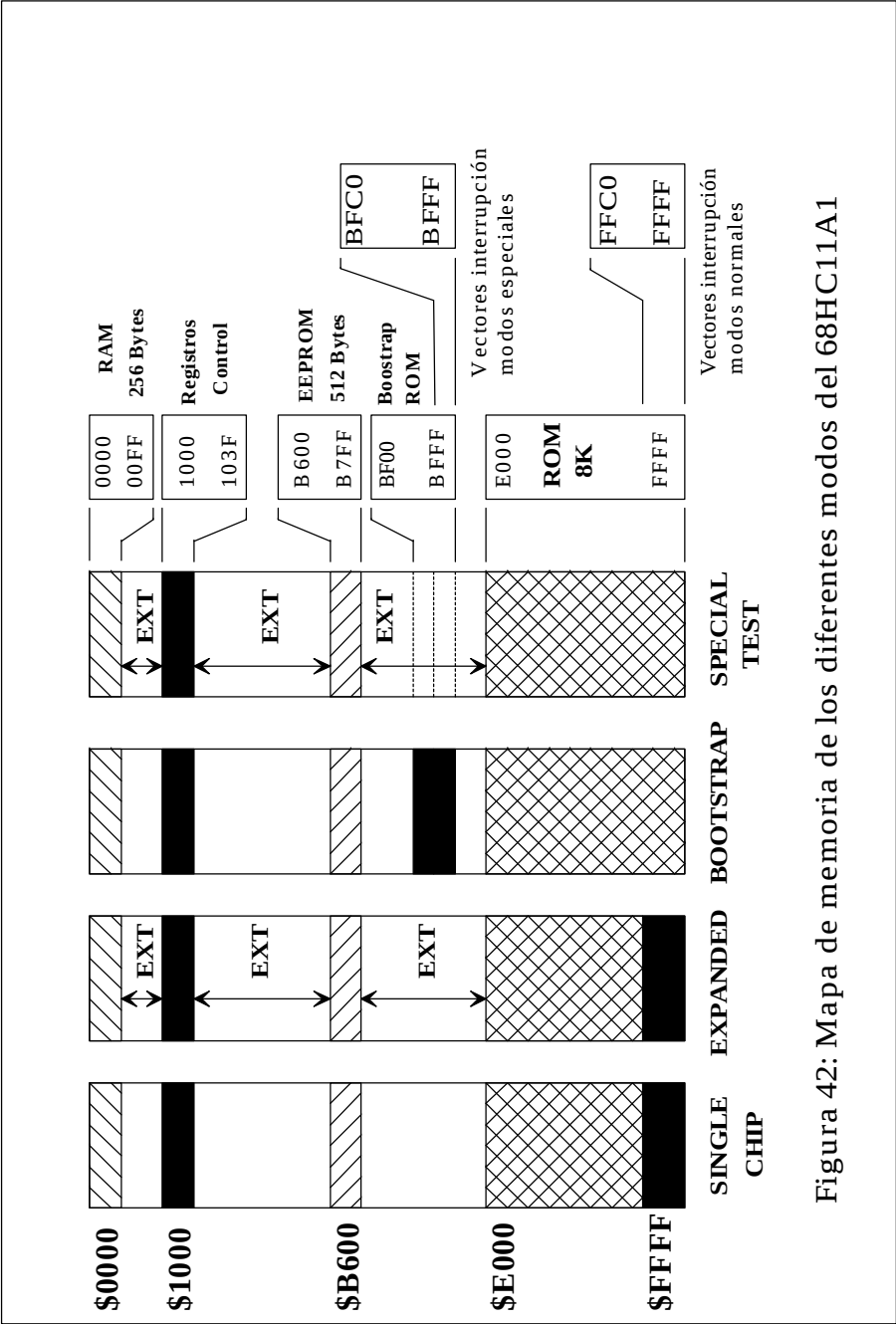
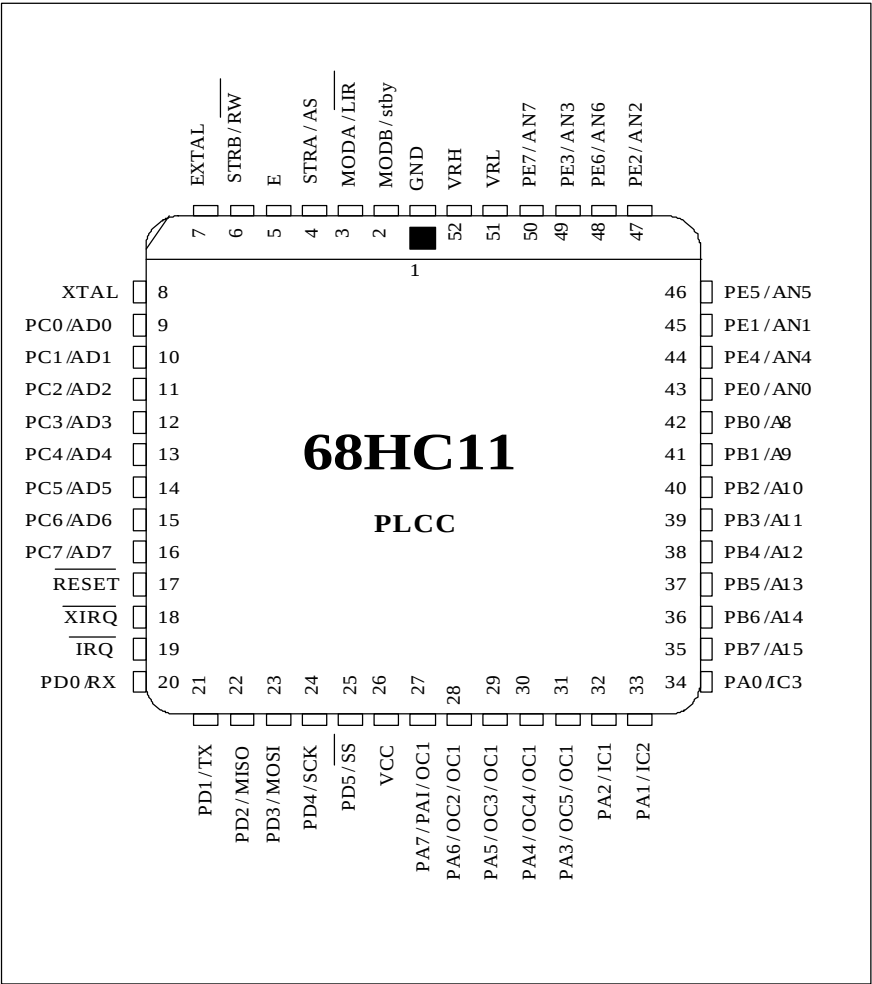
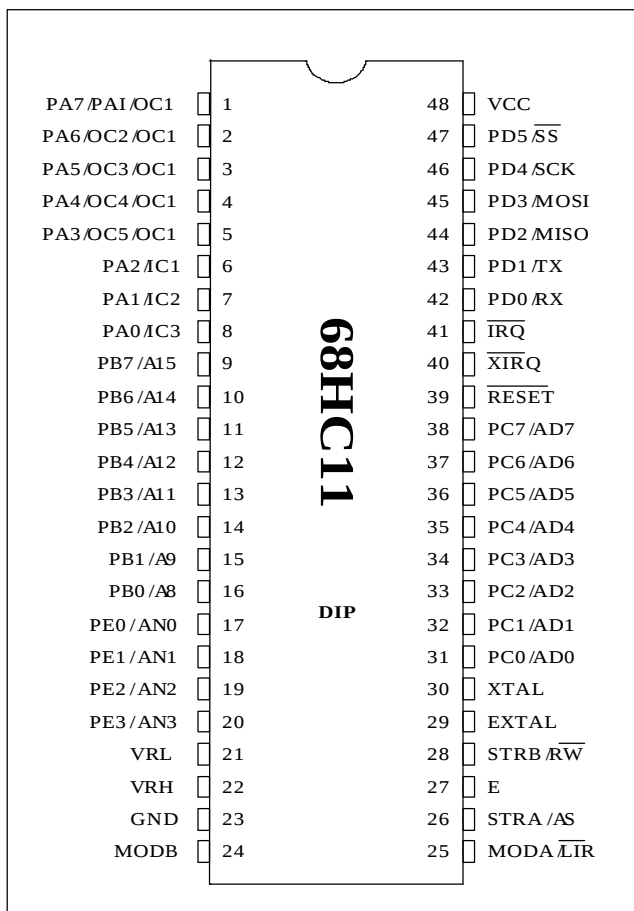


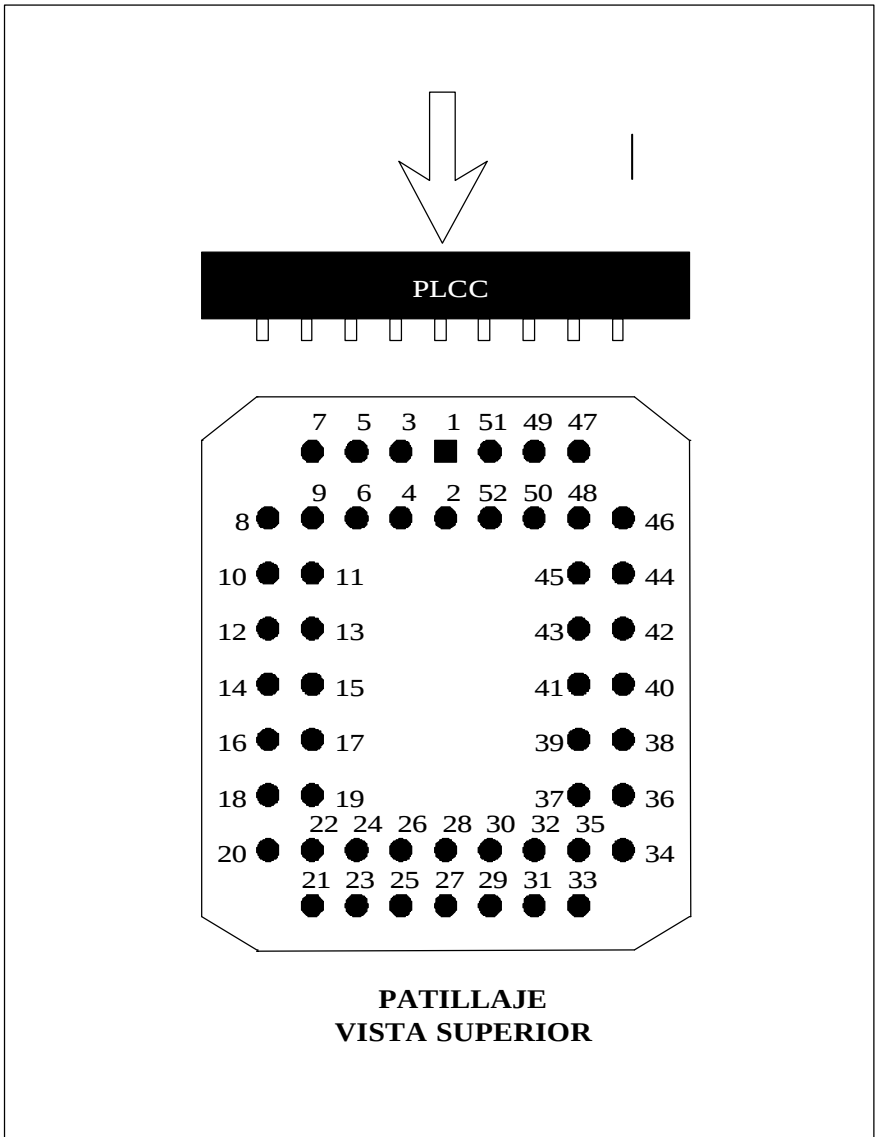
Figura 42: Mapa de memoria de los diferentes modos del 68HC11A1

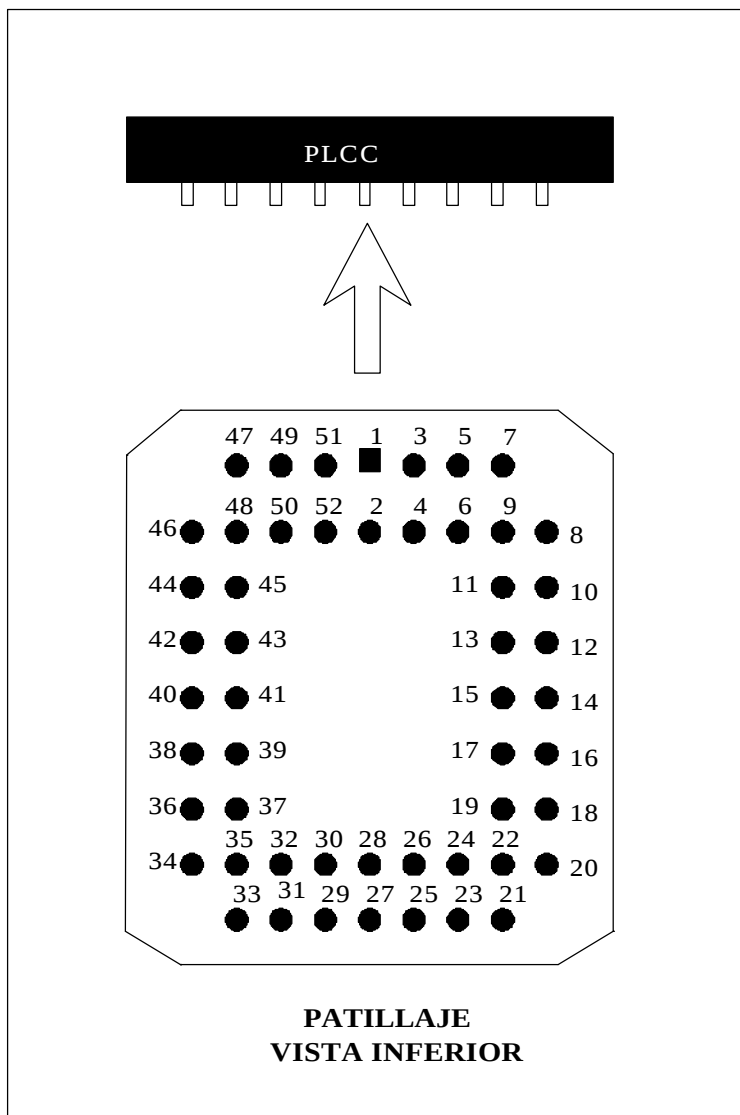
APÉNDICE B: PATILLAJE DEL 68HC11



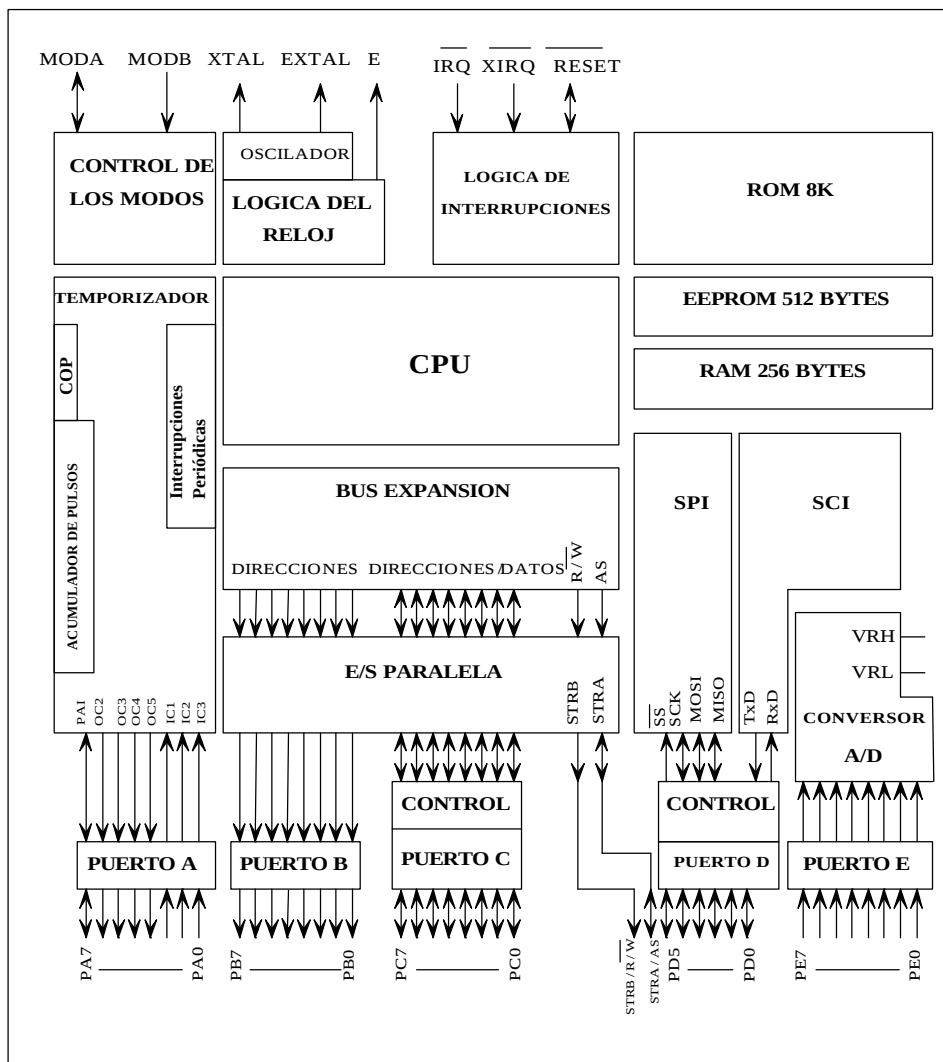


APÉNDICE C: Numeración del zócalo PLCC de 52 pines





APENDICE D: Diagrama de bloques del 68HC11



APÉNDICE E: 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 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, 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 (Ver figura 10 en la página 17), 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 tras pasarlo 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'.