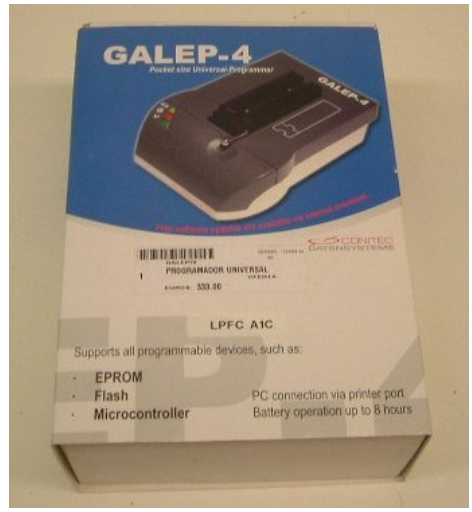
Ejemplo de utilización del programador universal CONITEC DATASYSTEMS GALEP-4 con el microcontrolador Microchip PIC 16F690 y el compilador CCS PCWH 3.249



Versión 1.0

Autor: Marcos Morales Pallarés

Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona
Laboratorio de Proyectos. Unidad de Electrónica
Universitat Politècnica de Catalunya

Barcelona, noviembre del 2006

Índice

1. Objetivo de este documento

El programador universal GALEP-4 de la firma CONITEC DATASYSTEMS está disponible en el laboratorio de PFCs para aquellos/as proyectistas que lo necesiten. El gran número de circuitos integrados que permite programar lo convierte en una herramienta muy versátil.

En muchas ocasiones los estudiantes necesitan volcar programas en microcontroladores, grabar datos en EEPROMs o programar PALs. Estos dispositivos pueden ser programados con el GALEP-4. Sólo precisamos seleccionar el integrado a emplear, generar el archivo en formato hexadecimal que queramos volcar y programar el integrado.

En el caso de no disponer de los drivers necesarios para programar algún chip, siempre podremos consultar la web del fabricante (http://www.galep.com/) y comprobar la existencia de actualizaciones y soporte para nuevos dispositivos.

Para que sirva de pauta y a modo de ejemplo, proporcionamos al lector este manual. En él se recogen los pasos necesarios para llevar a cabo el montaje más básico que existe para todo microcontrolador: el control de encendido de un LED.

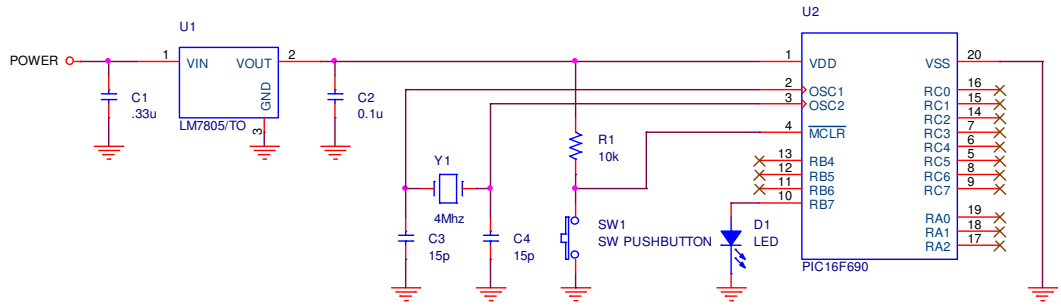
Para ello se ha escogido el PIC16F690, de la casa Microchip. La familia 16FXXX es reconocida por su versatilidad, facilidad de uso y robustez. Nos hemos decantado por el compilador CCS PCWH, pues se trata de una de las herramientas más empleadas al programar código en lenguaje C para los PIC, además de poseer gran cantidad de librerías y ejemplos.

El documento está estructurado a modo de guía de 'pasos' a realizar para llevar a cabo este ejemplo en concreto con éxito. Dada la naturaleza de este documento, no se ha creído conveniente entrar en detalles de funcionamiento.

Tomando este manual como punto de partida, no debería resultar difícil para el lector realizar procesos similares para otros microcontroladores. Obviamente, si lo que el lector precisa es volcar datos a otros circuitos integrados como por ejemplo EPROMs o GALs, deberá emplear otras herramientas software, aunque podrá guiarse de la sección dedicada exclusivamente al programador GALEP-4 y el volcado de datos, teniendo en cuenta que algunos de los parámetros aquí expuestos variarán.

2. Aplicación de ejemplo a montar y programar

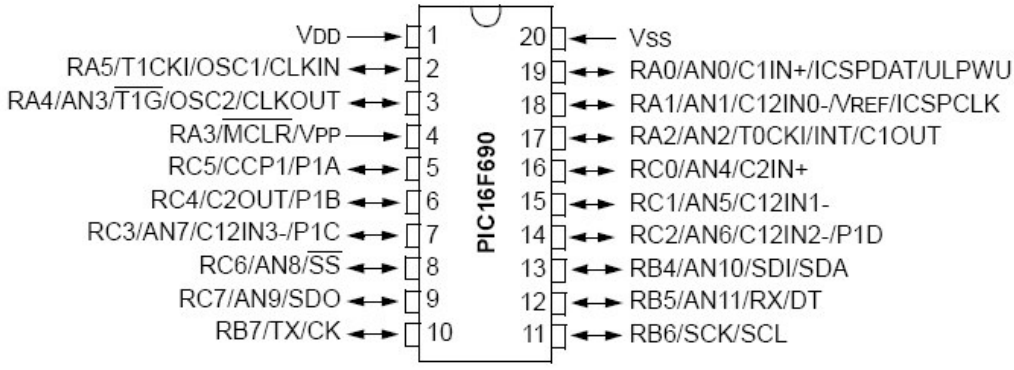
La siguiente figura muestra el circuito que implementaremos en nuestra protoboard.



Aspectos a observar:

- Alimentación regulada por un 7805 a 5 V. Los condensadores de desacoplo de 0.33 uF y 0.1 uF resultan imprescindibles.
- Empleo de un cristal de cuarzo de 4 MHz. En general se trata de la velocidad mínima empleada en todo montaje con microcontrolador.
- Botón de reset.
- LED conectado al pin RB7. Obsérvese la conexión directa al pin del micro, sin emplear ninguna resistencia limitadora de corriente. En general los PIC proporcionan la corriente necesaria para alimentar un diodo LED de forma directa. Esto nos sirve para nuestro ejemplo, aunque en la práctica se recomienda el uso de dichas resistencias.
- No hemos incluido un condensador de desacoplo para el micro, aunque para aplicaciones más complejas puede llegar a ser imprescindible.

A continuación se muestra la distribución de pines proporcionada en el datasheet del fabricante. Consúltese en caso de tener alguna duda.



3. Procedimiento

Los subapartados que se exponen a continuación deben seguirse de forma lineal.

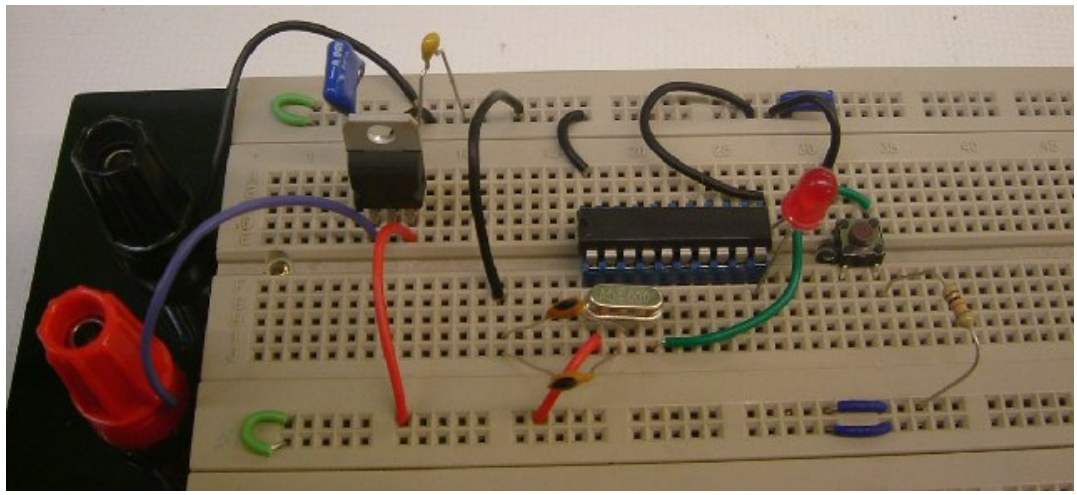
No hace falta decir que, en el caso de tener ya instalado el compilador CCS PCWH o el software del programador universal GALEP-4 podremos prescindir de los apartados 3.2 y 3.3, respectivamente.

De todos modos, si la versión del driver que estamos utilizando no soporta el dispositivo que queremos programar, será necesario consultar en la web del fabricante si existe la actualización correspondiente. En el momento de elaborar este manual, ha sido necesario realizar dicha actualización. Puede tomarse el punto 3.3 como ejemplo para realizar esta tarea.

Dentro de cada apartado observaremos que los pasos se encuentran numerados. En cada uno de ellos se incluye una breve explicación de lo que se debe realizar o, en su defecto, una imagen explicativa por sí sola. Los apartados se encuentran separados por barras negras horizontales.

3.1. Montaje del circuito en protoboard

1. El aspecto del circuito una vez montado es el siguiente:

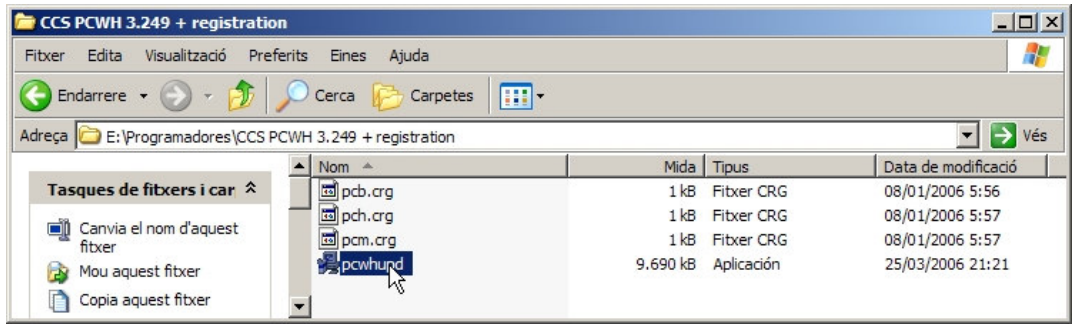


Obsérvese el empleo de un zócalo de 20 pines para nuestro micro. Resulta muy recomendable su uso puesto que al programar varias veces el micro y comprobar su funcionamiento estaremos insertándolo y extrayéndolo de la protoboard continuamente, con lo que podríamos dañar físicamente alguna de sus patitas.

Nuestro objetivo será volcar y ejecutar un programa que encienda y apague el LED de forma intermitente e indefinida. El porqué de realizar un montaje y un programa tan simple es porque nos permite comprobar de un modo sencillo que el proceso de compilación y volcado se ha realizado de forma correcta.

3.2. Instalación de CCS PCWH 3.249

1. Ejecutar 'pcwhund.exe':

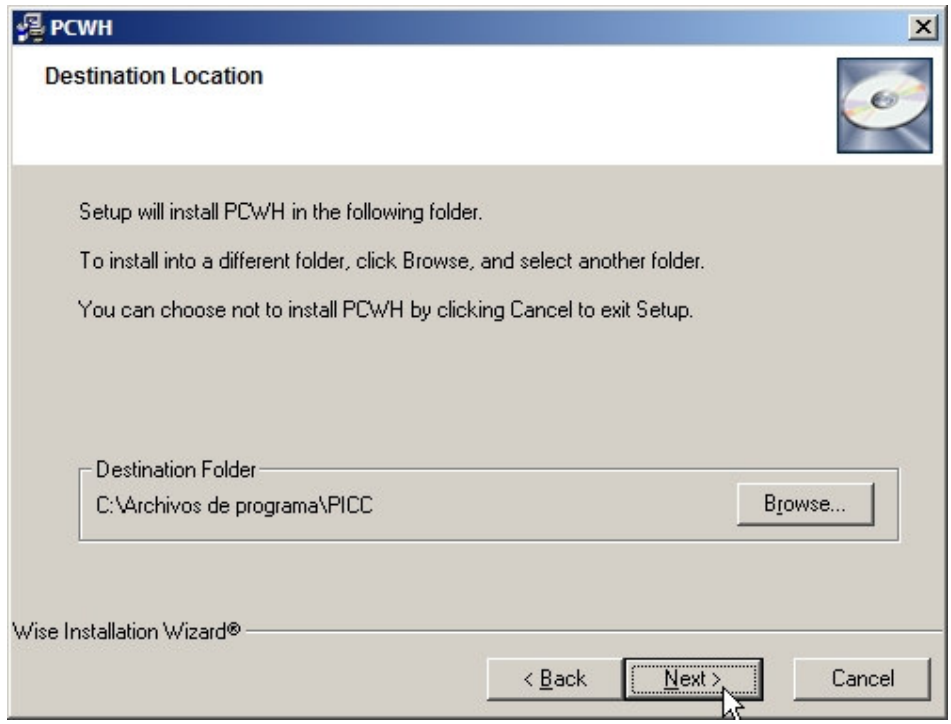


---

2. Clic en 'Next' > Clic en 'Next' > Clic en 'Next'

---

3. Seleccionar la carpeta destino para instalar el programa:

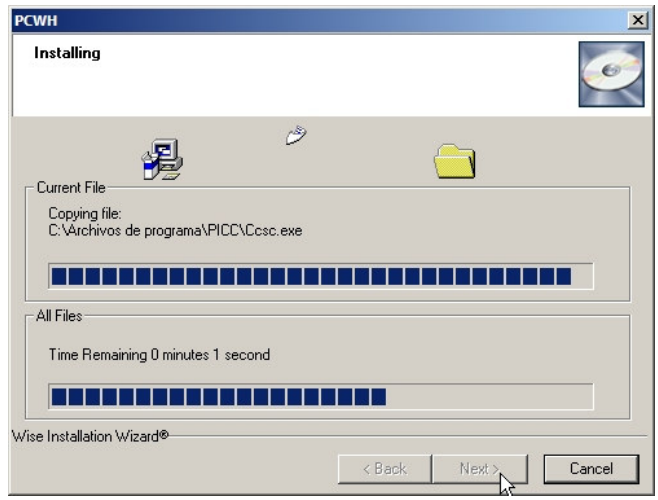


---

4. Clic en 'Next'.

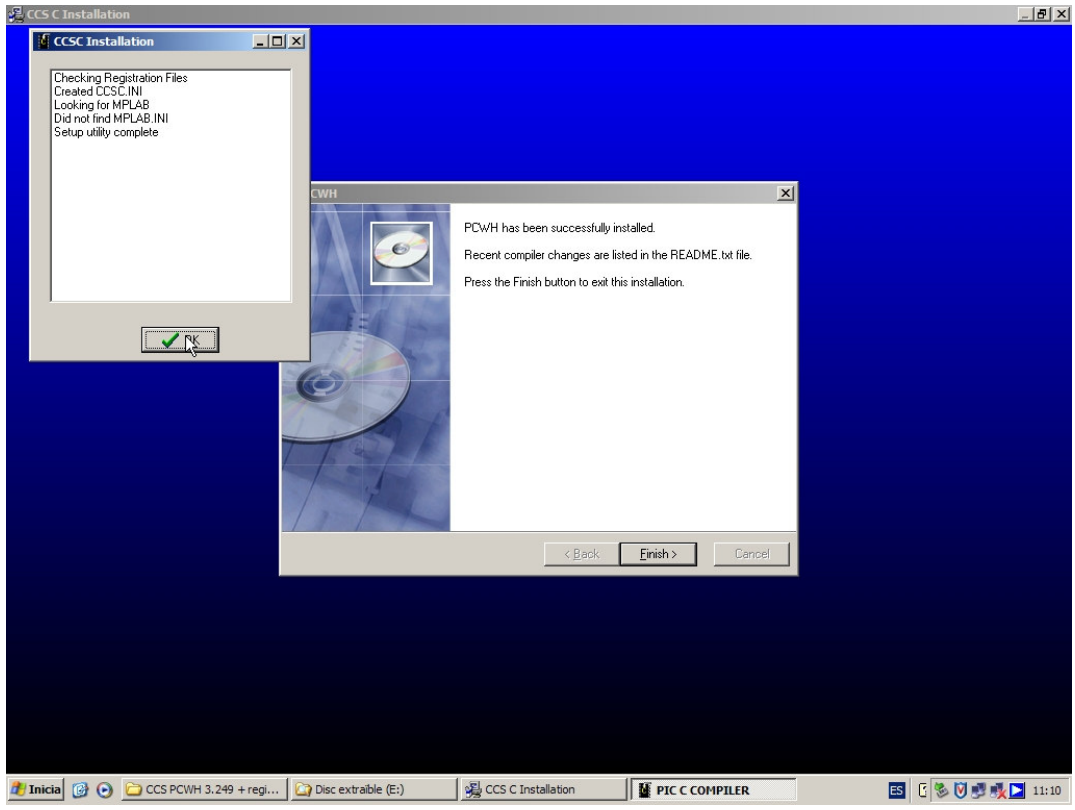
---

5. Esperar a que acabe la copia de archivos:



---

6. Si todo ha ido bien, aparecerán los siguientes cuadros informativos:



Con lo que finalizamos con la instalación de CCS PCWH 3.249.

3.3. Instalación del GALEP-4

1. El pack del programador universal GALEP-4 incluye los siguientes elementos:
    - Módulo programador.
    - CD con el software de utilización (versión de fábrica).
    - Cable para el puerto paralelo.
    - Fuente de alimentación.

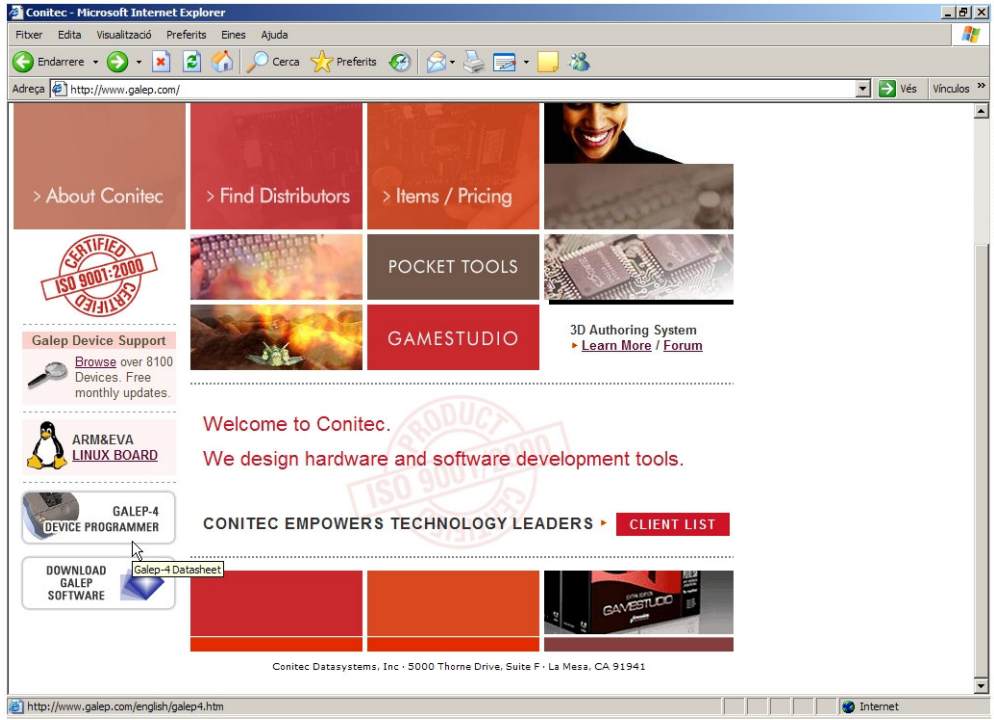
En caso de faltar algún elemento será preciso avisar a uno de los responsables del laboratorio.

---

2. Desde nuestro navegador, introducimos la dirección de la web del fabricante:
http://www.galep.com/

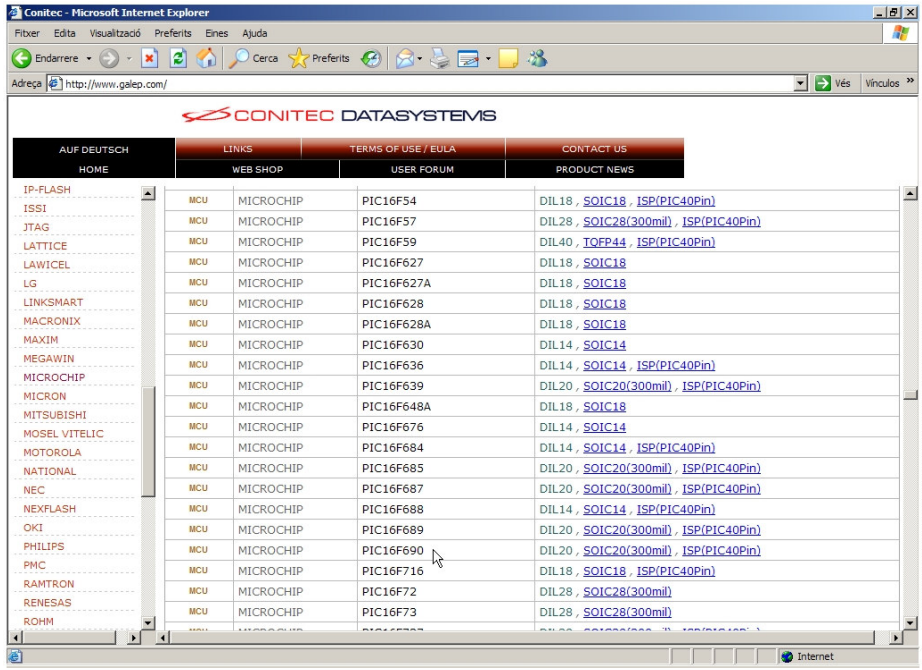
---

3. Click en 'GALEP 4 DEVICE PROGRAMMER'.



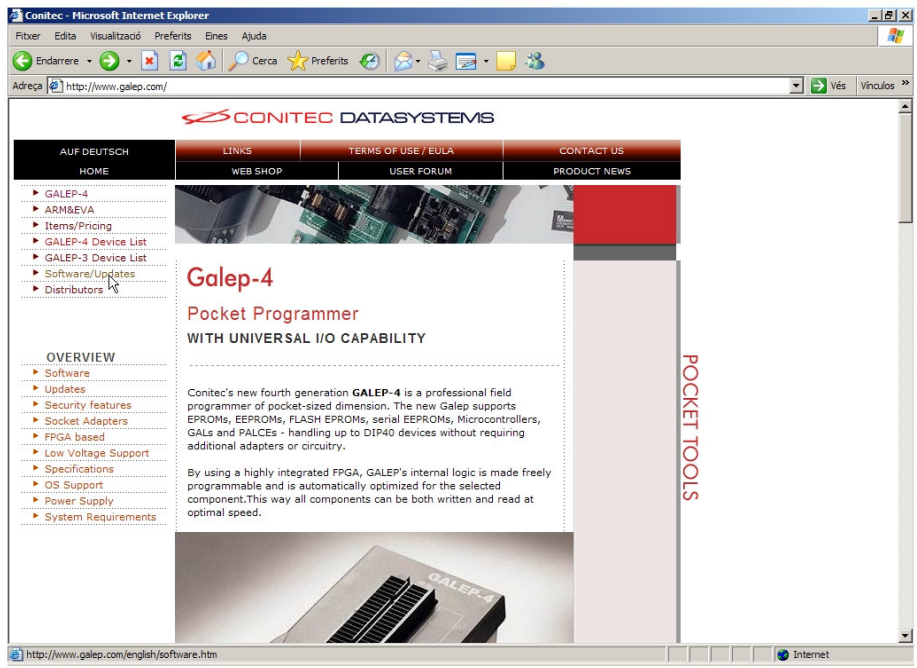
4. Vamos a averiguar si el fabricante da soporte a nuestro micro. Para ello hacemos click en 'GALEP-4 Device List'.

5. Buscamos el 16F690: primero debemos buscar 'MICROCHIP' en la lista de la izquierda. En la de la derecha nos desplazamos hasta encontrar la referencia PIC16F690. En el caso de no ser así no nos quedaría más remedio que cambiar el micro o, en su defecto, el programador (por otro de los disponibles en el laboratorio).



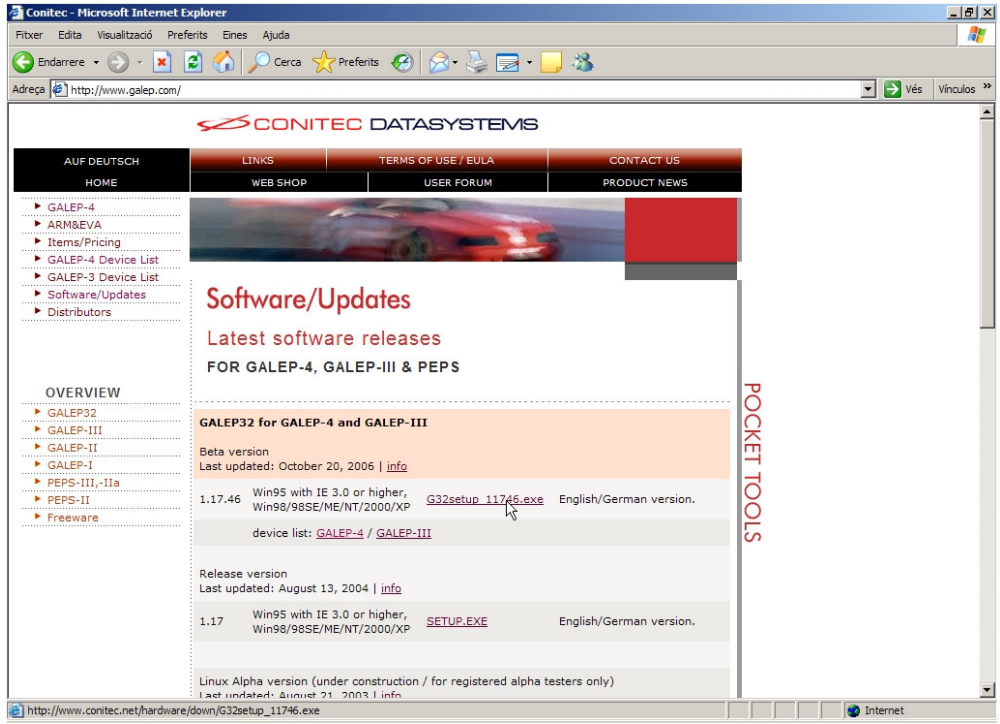
---

6. Procedemos a descargar la última versión del driver. Para ello volvemos a la pantalla inicial y clicamos en 'Software/Updates'.



---

7. Click en el link 'G32Setup_11746.exe'. Obviamente el nombre variará a medida que surjan nuevas actualizaciones.



---

8. Una vez descargado el archivo de instalación lo ejecutamos.

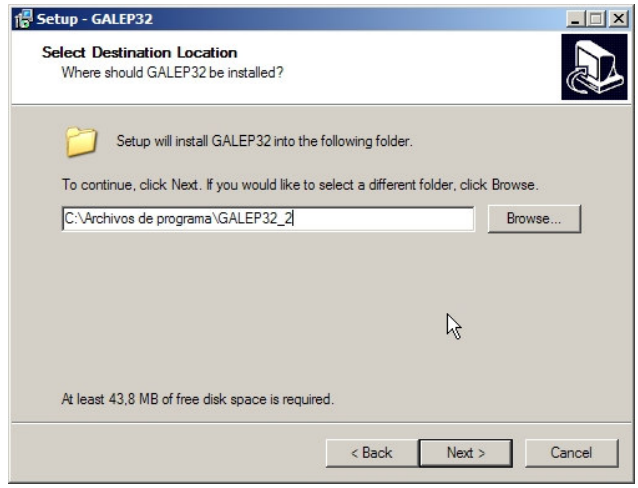
---

9. Desde la pantalla de bienvenida:

Click en 'Next' > Click en 'Next'.

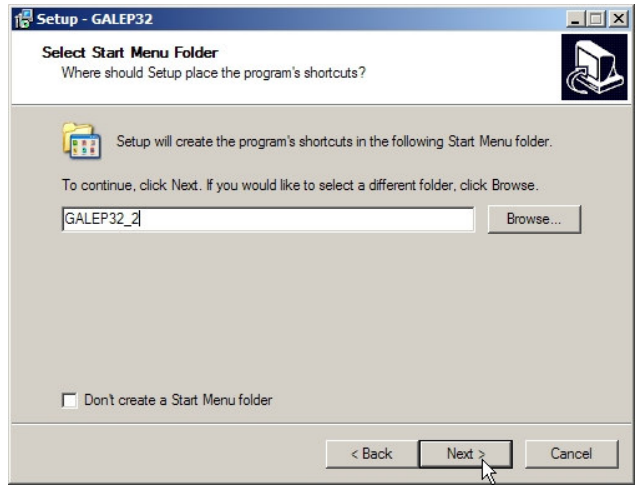
---

10. Introducimos la ruta de instalación. Si teníamos instalada una versión anterior, le indicamos un path distinto. (Click en 'Next').



---

11. Ídem para la carpeta de accesos directos del menú Inicio. (Click en 'Next').



---

12. Click en 'Next' > Click en 'Next'.

---

13. Esperamos a que finalice la copia de archivos.

---

14. Click en 'Finish'.

3.4. Creación del programa a ejecutar en el PIC16F690

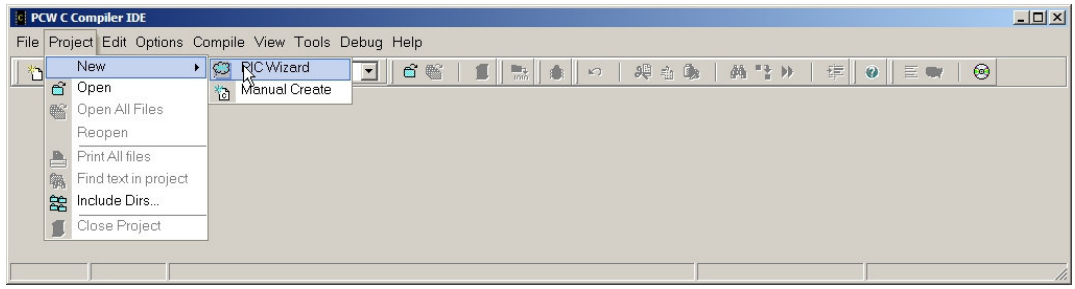
1. Iniciamos el CCS PCW PIC C Compiler:

Inicio > Programas > PIC-C > PIC C Compiler

---

2. Ejecutamos el asistente para la creación de proyectos:

Project > New > PIC Wizard

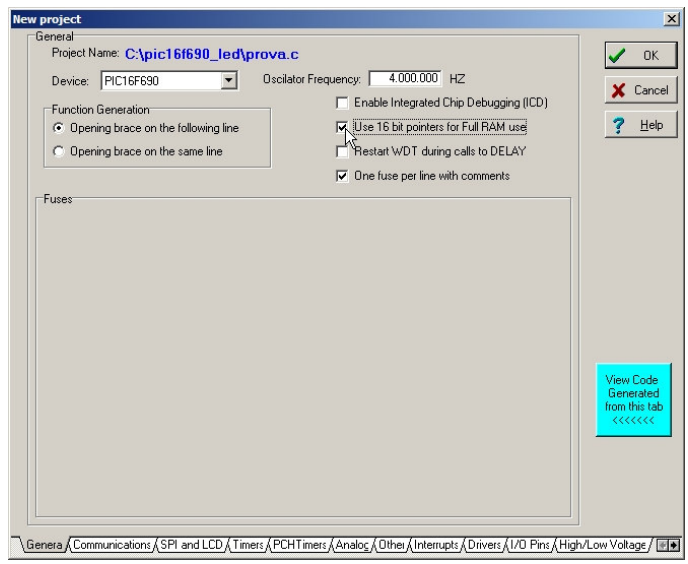


---

3. Se nos preguntará por el nombre y ubicación de nuestro proyecto. Es recomendable dedicar un directorio exclusivamente al proyecto, pues el número de archivos generados puede ser considerable.

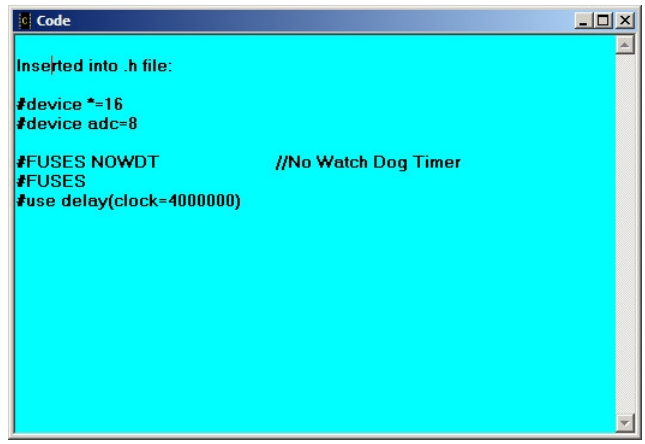
---

4. Aparecerá el asistente abierto por la pestaña 'General':
En 'Device' seleccionamos nuestro modelo de microcontrolador (PIC16F690).
Introducimos la frecuencia de trabajo de nuestro cristal oscilador (4000000).
Seleccionamos el uso de punteros de 16 bits para poder utilizar toda la RAM del micro.



---

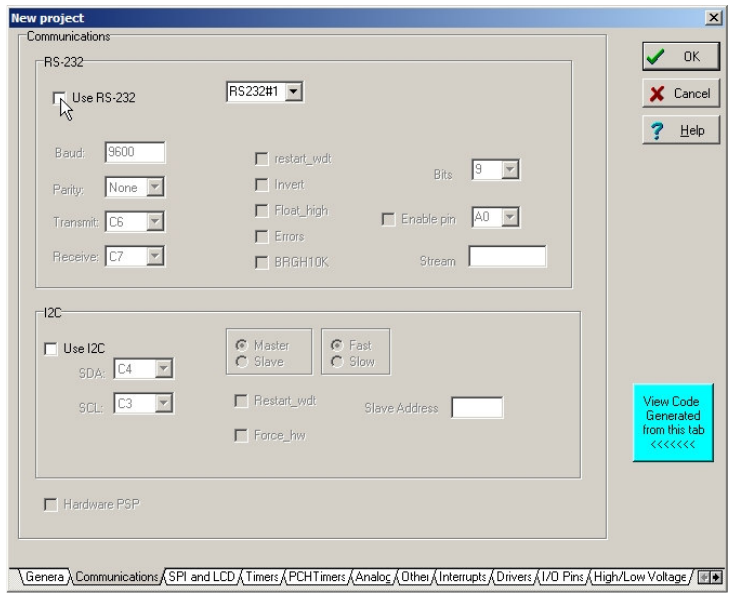
5. Si clicamos en 'View Code Generated from this tab' observaremos la ventana inferior.



En ella podemos observar el código C que el asistente generará (y dónde lo insertará) teniendo en cuenta las opciones seleccionadas de la pestaña del asistente en la que nos encontramos.

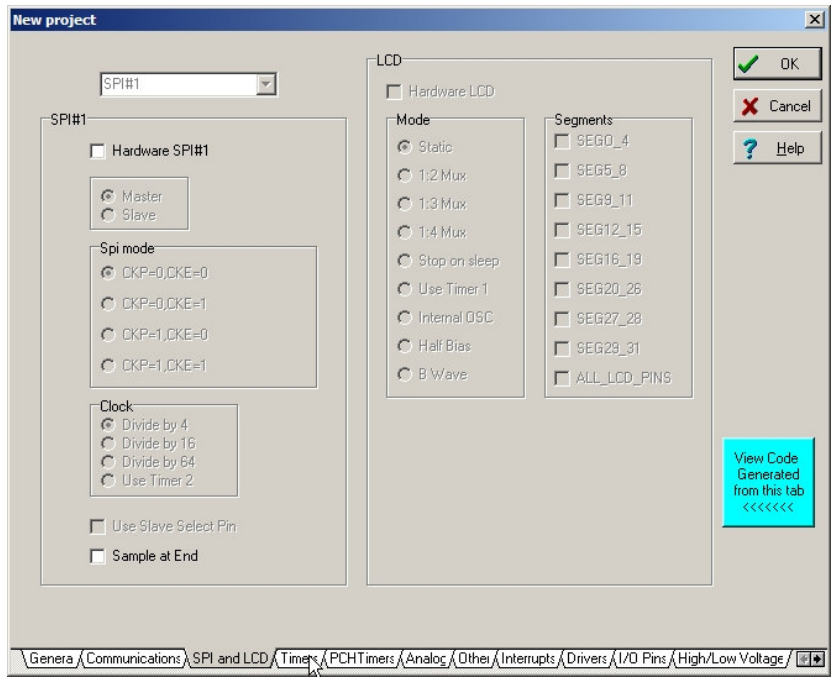
---

6. Pestaña 'Communications':



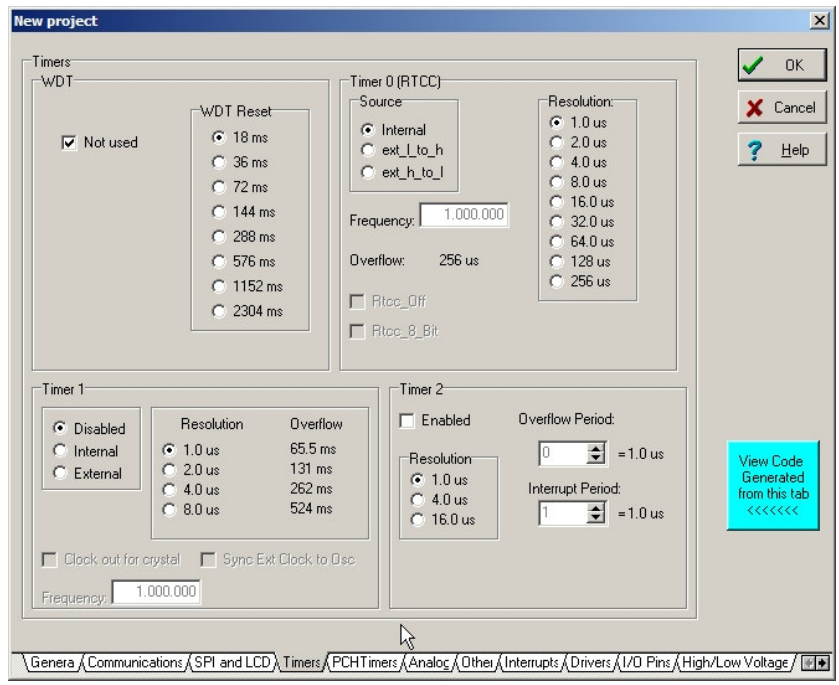
---

7. Pestaña 'SPI and LCD':

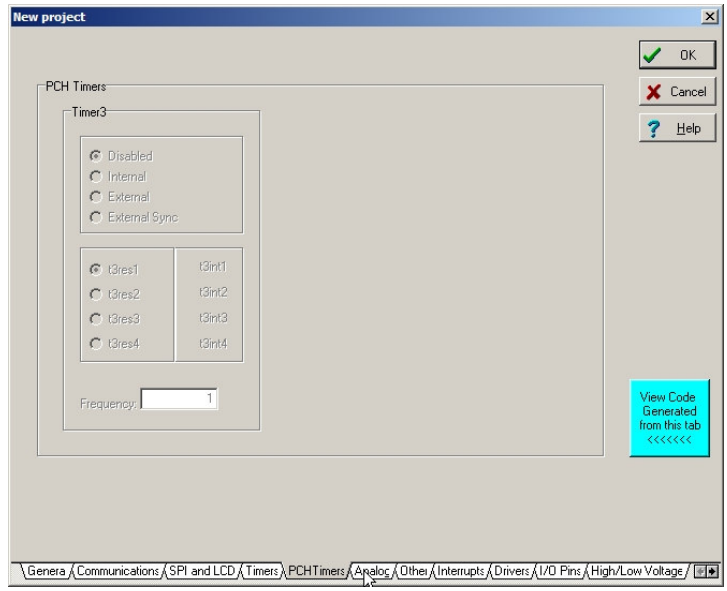


---

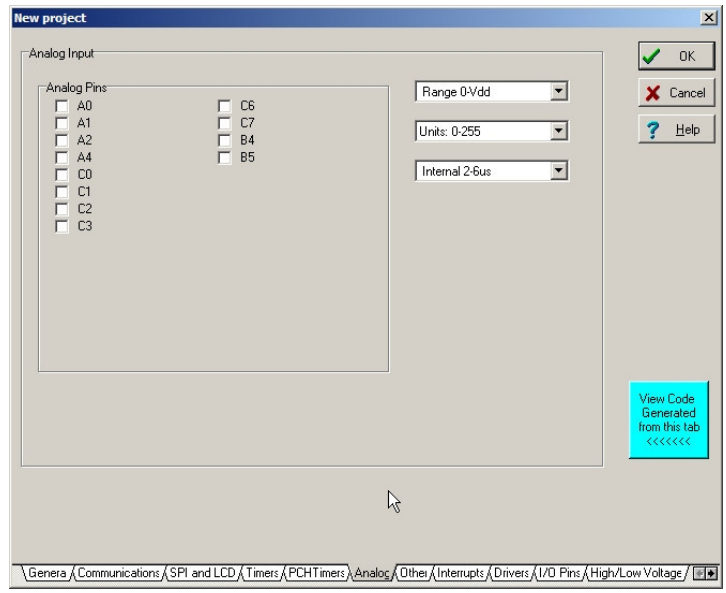
8. Pestaña 'Timers':

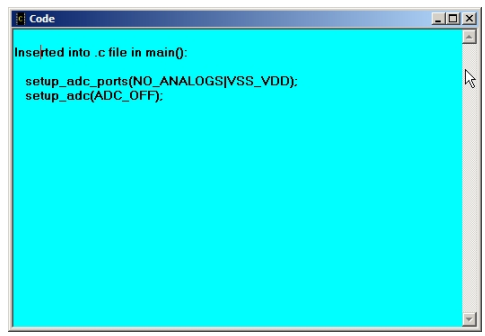


9. Pestaña ‘PCH Timers’:

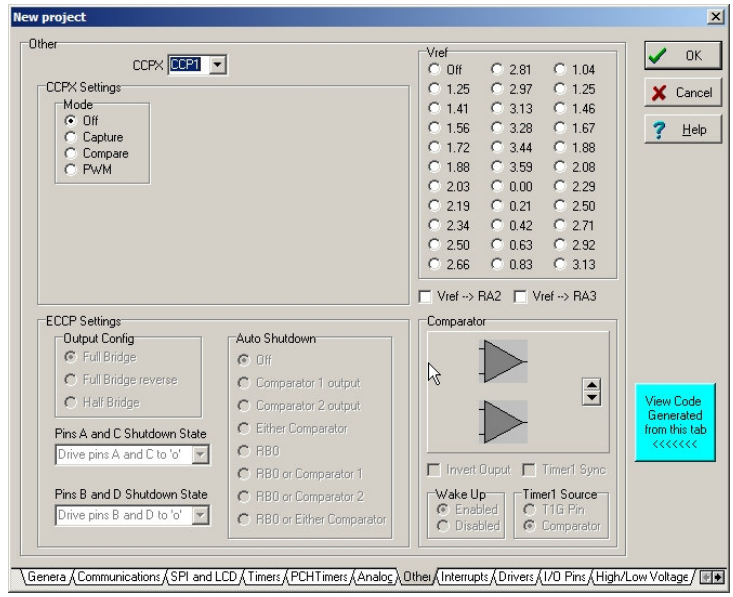


10. Pestaña ‘Analog’:

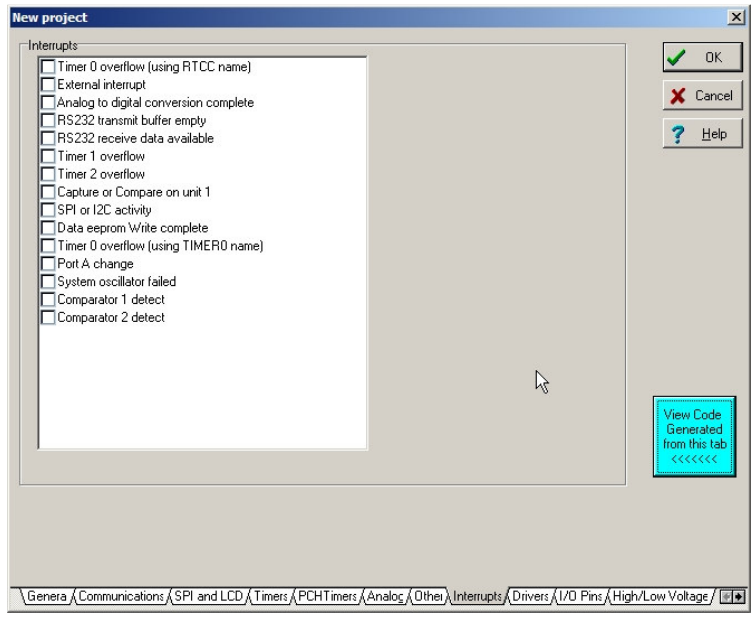




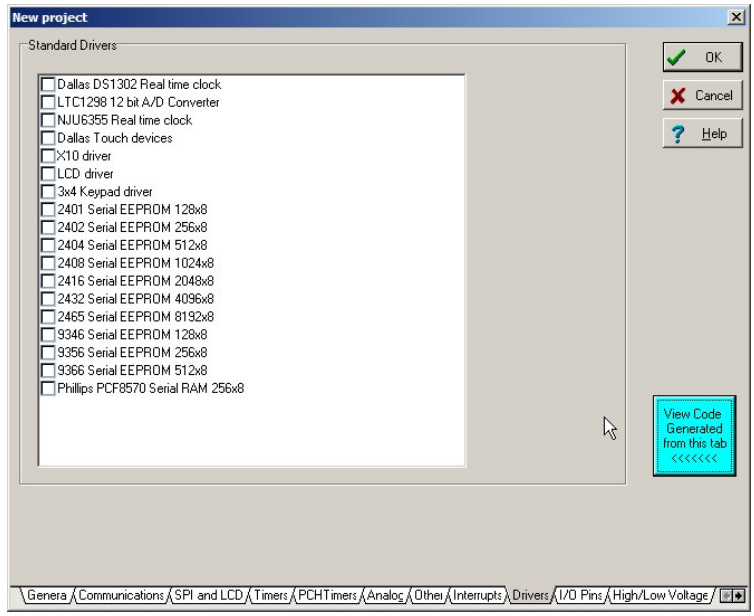
11. Pestaña ‘Other’:



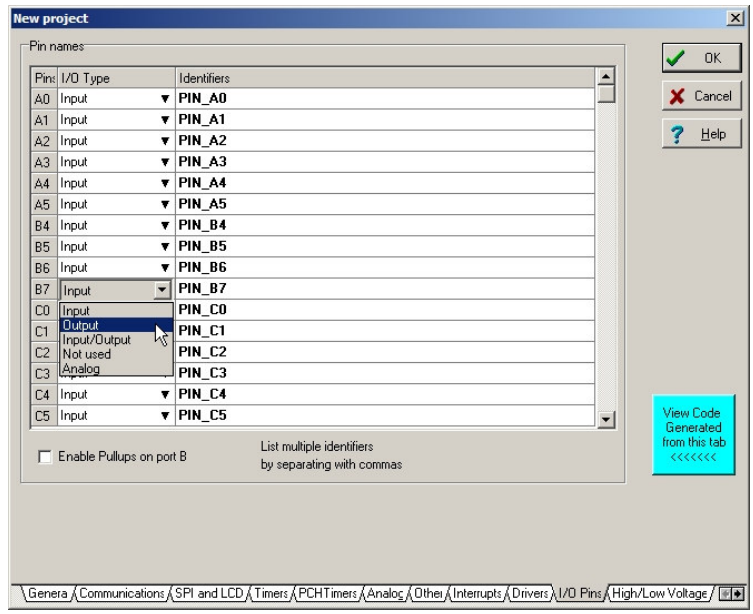
12. Pestaña ‘Interrupts’:



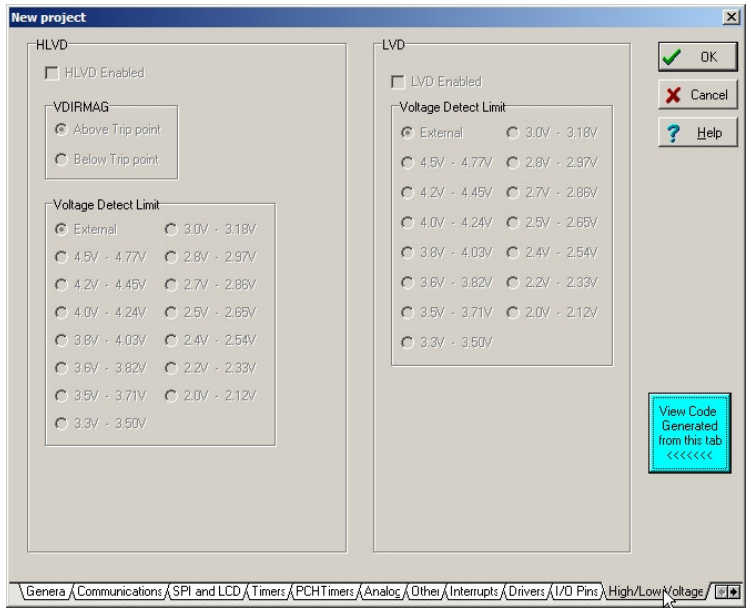
13. Pestaña 'Drivers':



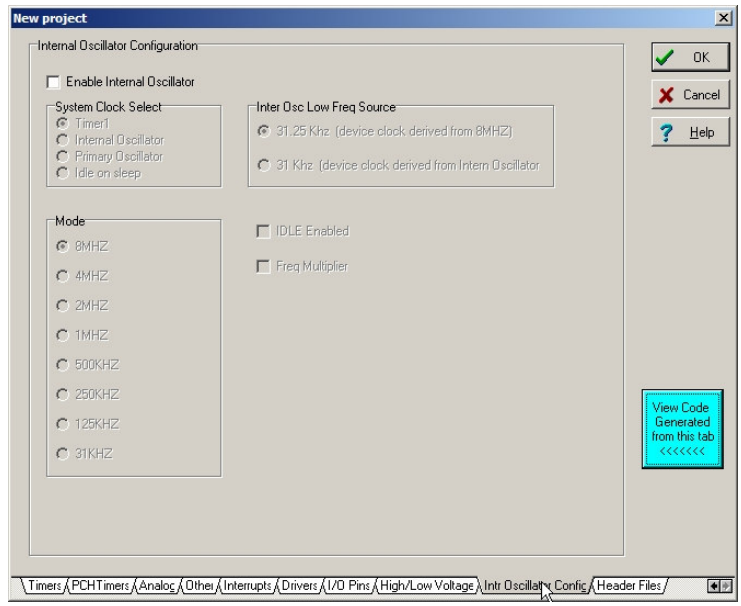
14. Pestaña 'I/O Pins':



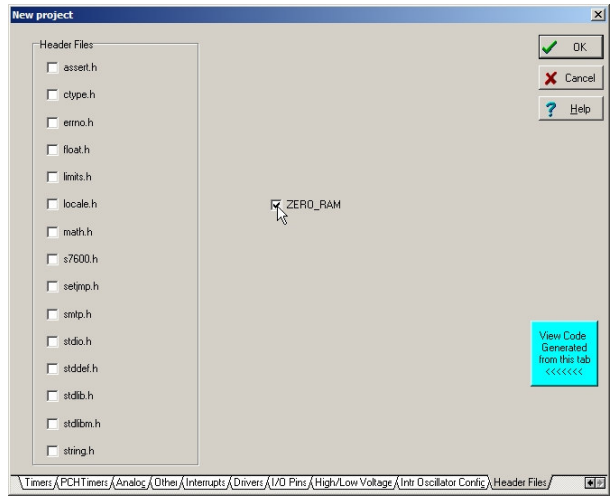
15. Pestaña 'High/Low Voltage':

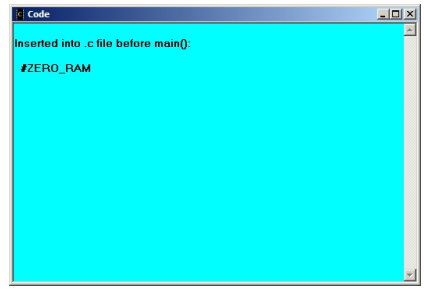


16. Pestaña 'Intr Oscillator Config':



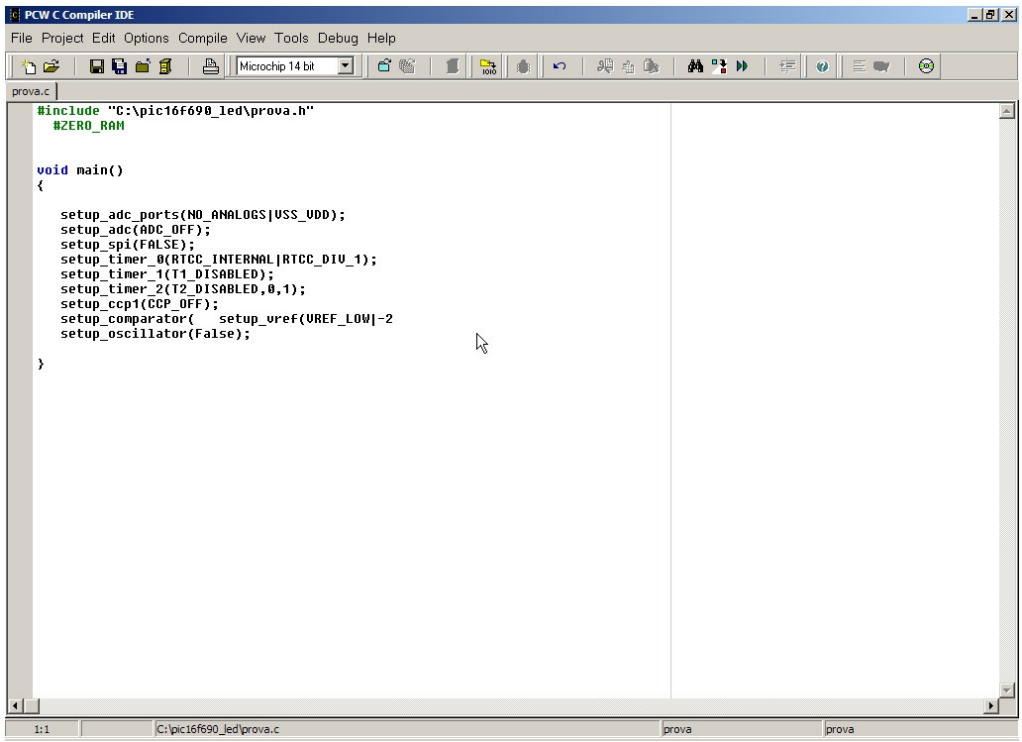
17. Pestaña 'Header Files':





---

18. Finalmente clicamos en 'OK' y obtendremos el siguiente código:



Podemos realizar varias observaciones:

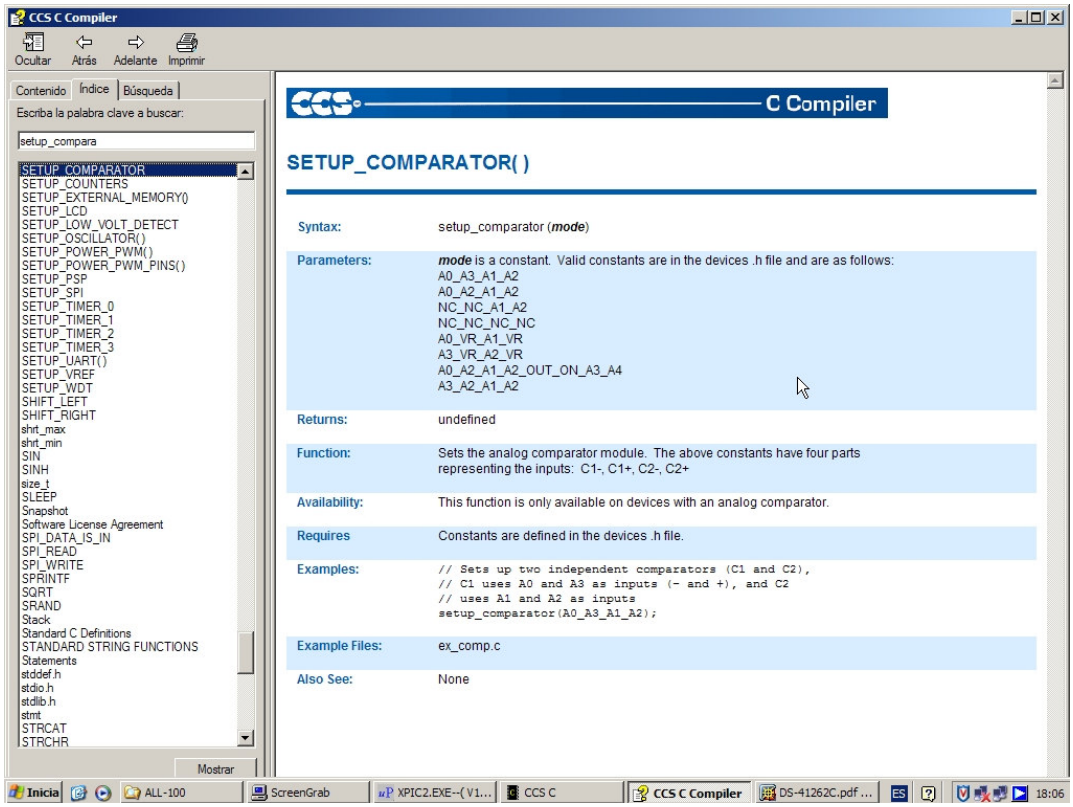
- Nuestro proyecto se ha creado con un archivo de código principal 'prova.c', y un archivo cabecera 'prova.h'.
- En la rutina principal (main) podemos observar una serie de llamadas a funciones de configuración que proporciona el propio compilador: setup_xxx.
- El asistente ha redactado un par de instrucciones con errores de sintaxis: setup_comparator(x) y setup_vref(x). Se trata de un bug de CCS PCWH que deberemos solucionar.

---

19. Consultamos la ayuda para conocer la sintaxis de setup_comparator(x):

Help > Contents

Clicamos en la pestaña 'Índice' y accedemos a la entrada SETUP_COMPARATOR de la lista.



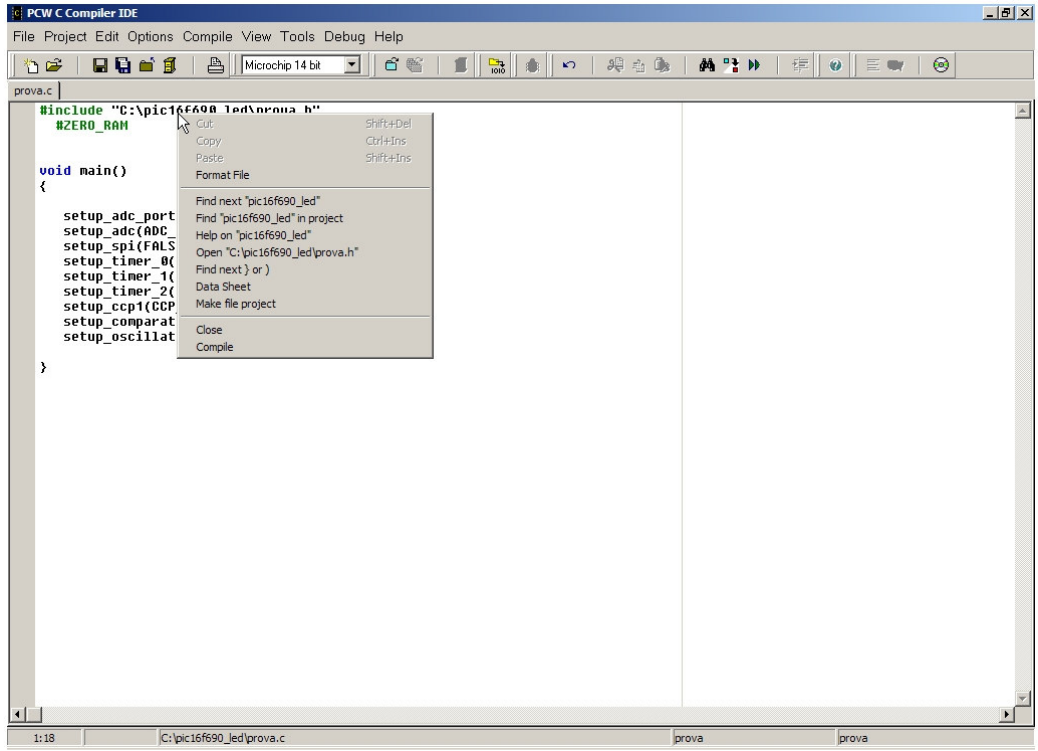
Como se puede observar, la función setup_comparator consta de un único parámetro 'mode', que especifica la configuración de los comparadores del micro. Como no vamos a emplear ninguno, usaremos la constante NC_NC_NC_NC.

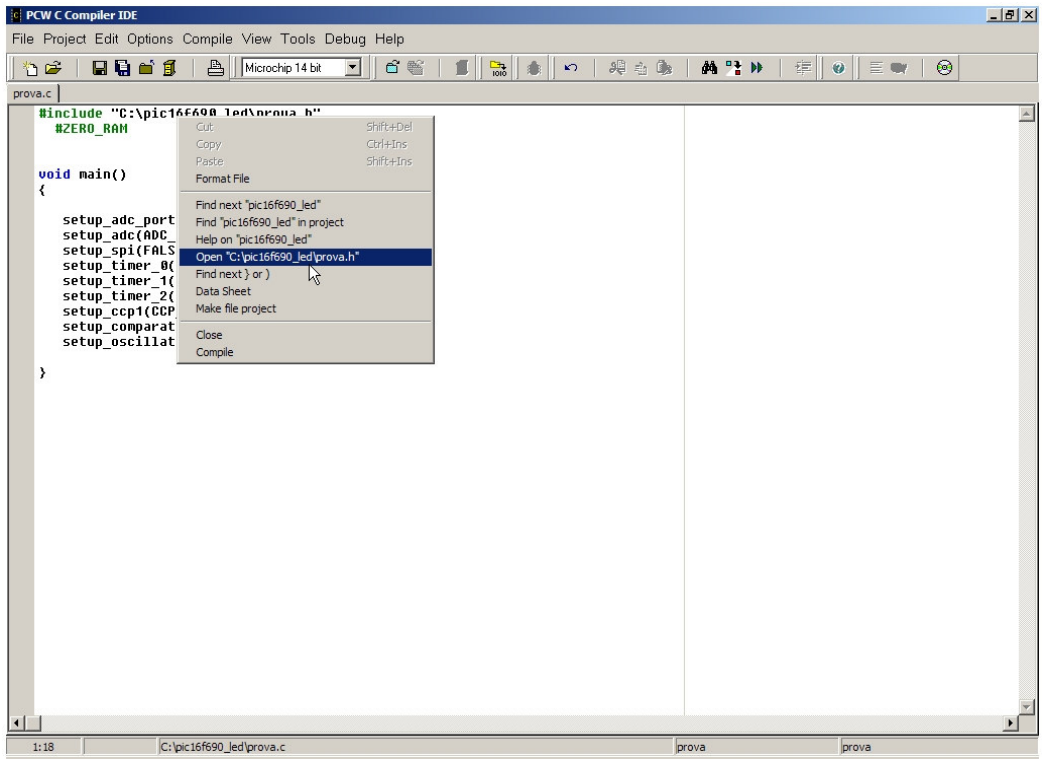
---

20. El paso anterior se puede realizar también accediendo al fichero de cabecera específico del 16F690, y consultando la sección que contiene las definiciones correspondientes a los comparadores, y consultando a su vez en el datasheet el valor de los valores numéricos que las definiciones nos indican.

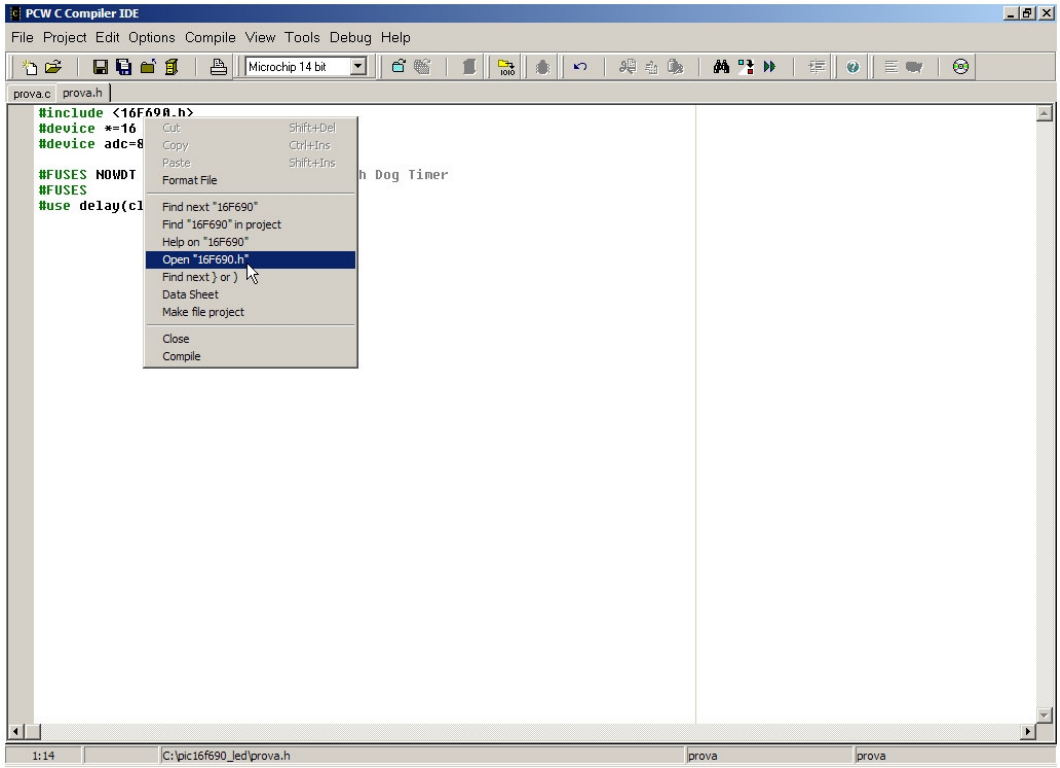
Para acceder al fichero 16F690.h podemos hacer:

- Clic con el botón derecho en el fragmento de código que incluye prova.h.
- Clic en Open '…\prova.h'.

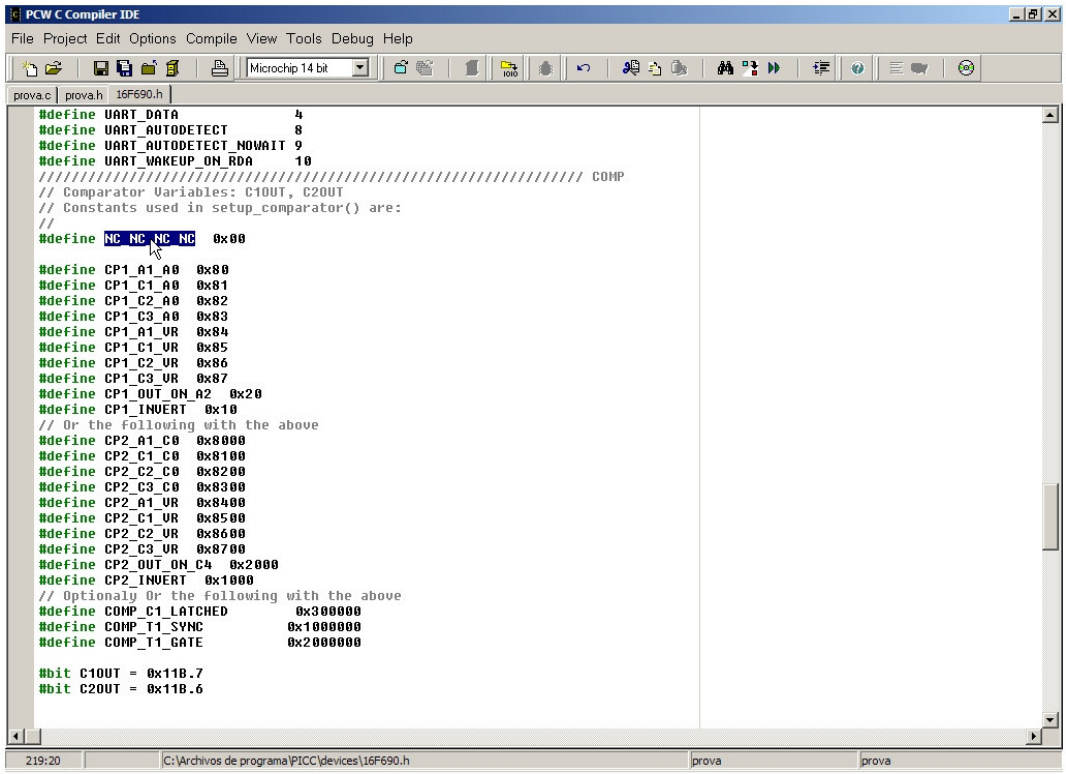




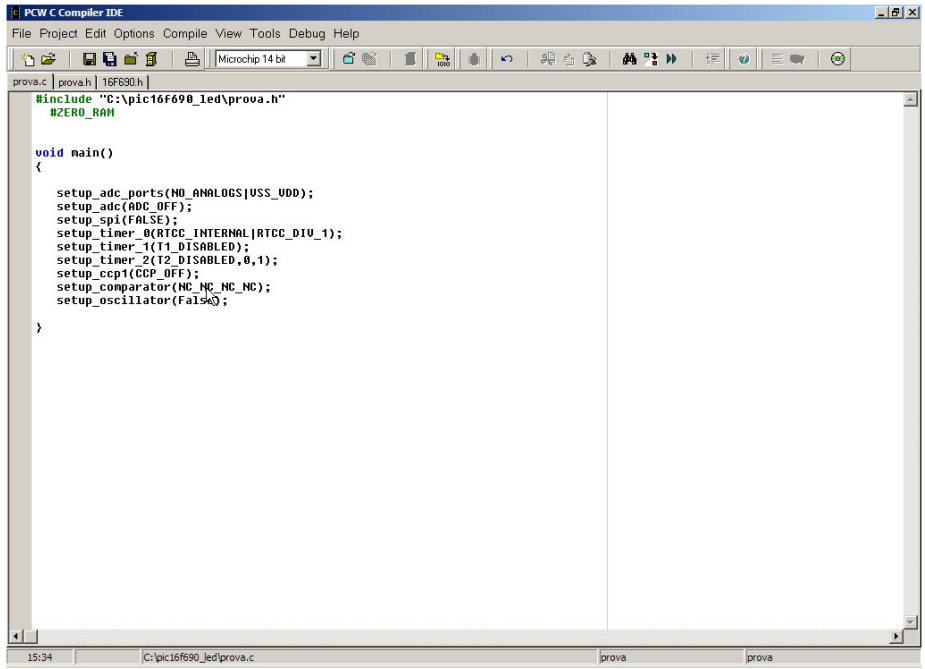
Una vez abierto 'prova.h', repetimos el proceso con '16f690.h', cuya inclusión se realiza desde prova.h.



En la figura inferior se puede observar la sección de comparadores del fichero '16f690.h':



21. En la siguiente figura podemos observar el código final después de los arreglos.

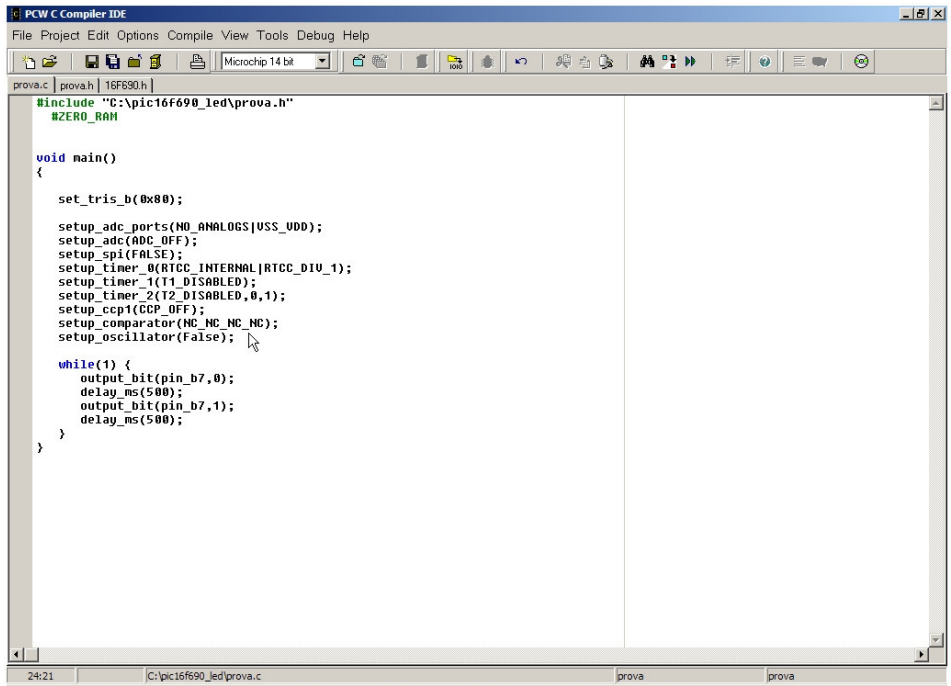


A destacar:

- La llamada a la función setup_comparator, con la constante que deshabilita los comparadores.
- La supresión de la llamada a la función que configura la tensión de referencia del A/D (setup_vref). Al no emplearla no será necesario configurar nada.

---

22. Escribimos el resto del código:

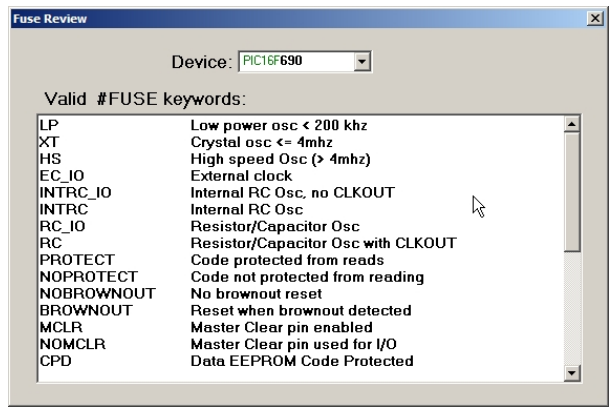


Como se puede ver, se trata de un bucle infinito (while(1)), en cuyo interior se activa y desactiva el bit 7 del puerto b (RB7), con retardo de 0,5 segundos. Así pues, si todo ha ido bien, veremos parpadear al led conectado a dicho pin.

---

23. A continuación vamos a ver qué constantes podemos asignar a la directiva #FUSES.

View > Valid Fuses.



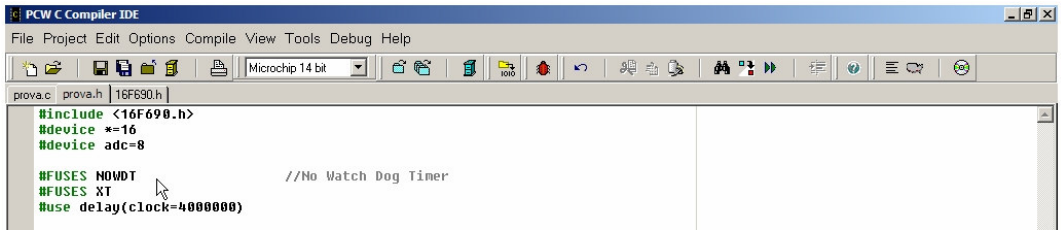
Añadiremos la directiva que indica que nuestro cristal funciona a una frecuencia inferior o igual a 4 MHz.

Como se puede observar, existen otras constantes interesantes. Por ejemplo, si quisiéramos proteger nuestro código de posibles lecturas, emplearíamos la constante PROTECT.

Deberemos tener en cuenta qué directivas empleamos en nuestro código porque deberemos configurar al programador universal en consonancia con los parámetros aquí escogidos.

---

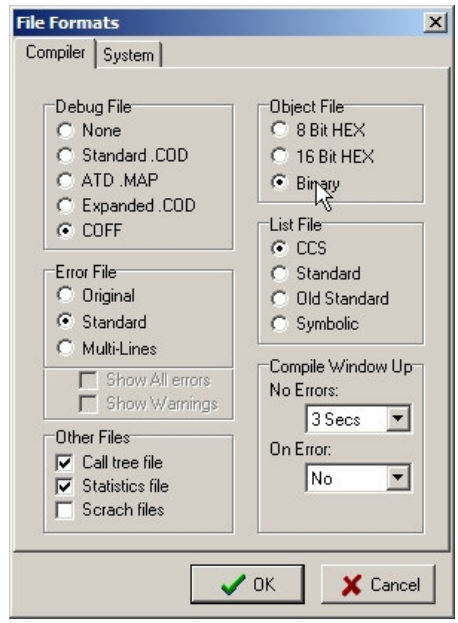
24. Modificamos prova.h añadiendo el parámetro XT a la directiva #FUSES:



---

25. Antes de compilar deberemos especificar el formato del archivo .hex que cargaremos en nuestro micro.

Accedemos al menú Options > File Formats y seleccionamos la opción 'Binary' del marco 'Object File'.

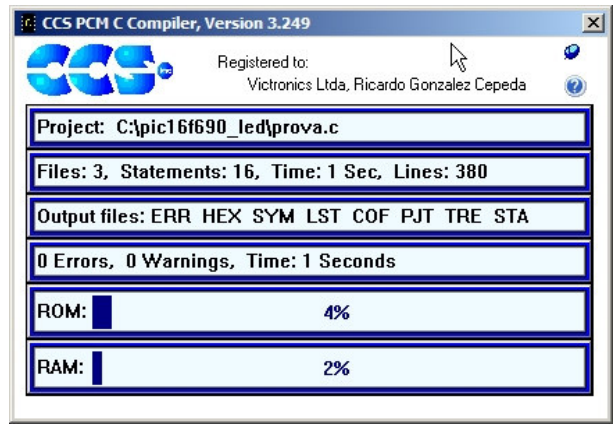


---

26. Recordemos salvar nuestro proyecto de vez en cuando. Una buena costumbre es hacerlo siempre antes de compilar cualquiera que sea el entorno de programación en el que trabajemos.

Ya podemos compilar nuestro proyecto mediante el menú Compile > Compile.

Si todo ha ido bien veremos la siguiente figura:

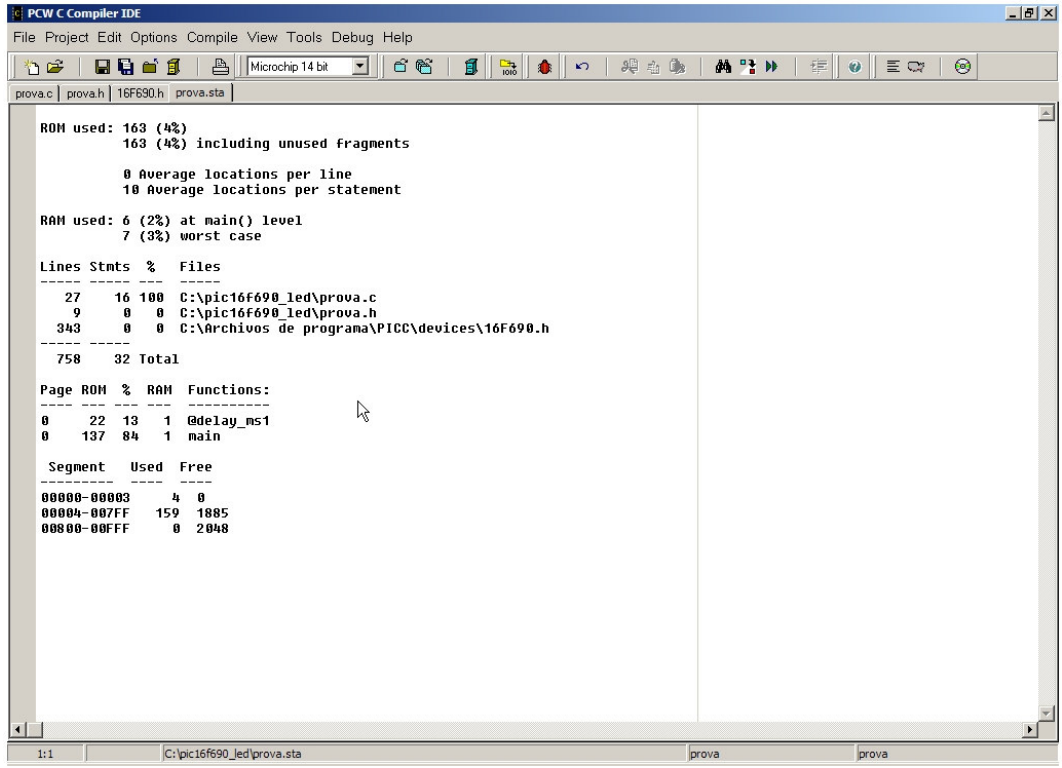


Obsérvese como una aplicación tan simple apenas consume memoria ROM (programa) y RAM (variables y registros de tiempo de ejecución).

Ya tenemos en el directorio de nuestro proyecto el fichero 'prova.hex' con el código máquina de nuestro micro, cuyo contenido volcaremos con el GALEP-4.

---

27. Antes de pasar al programador, puede ser útil saber que después de compilar podemos acceder al menú View > Statistics. Se nos mostrará un archivo recopilatorio de datos estadísticos útiles sobre el rendimiento de nuestro programa.



3.5. Programación del microcontrolador

1. Nos aseguraremos de que el módulo programador no tiene ningún circuito insertado.

---

2. Conectamos el cable de impresora al módulo programador y al PC.

---

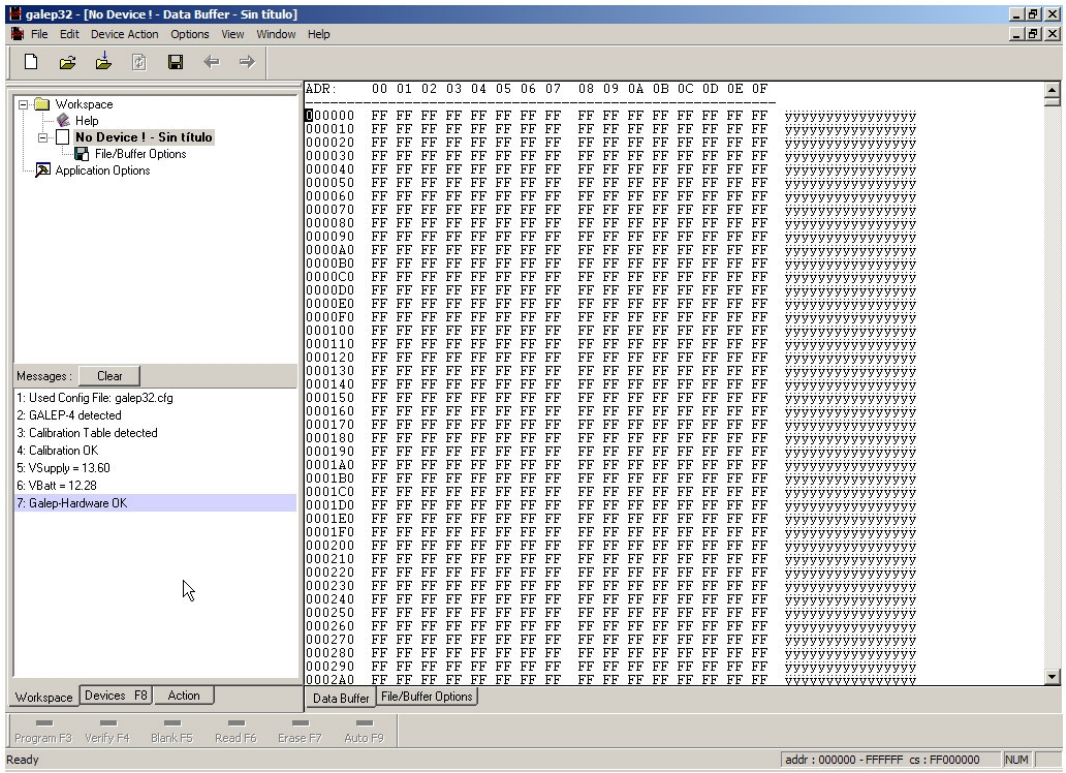
3. Conectamos la fuente de alimentación al módulo programador. Un LED rojo se encenderá (el de en medio).

---

4. Ejecutamos la aplicación GALEP 32 (la última versión instalada).

---

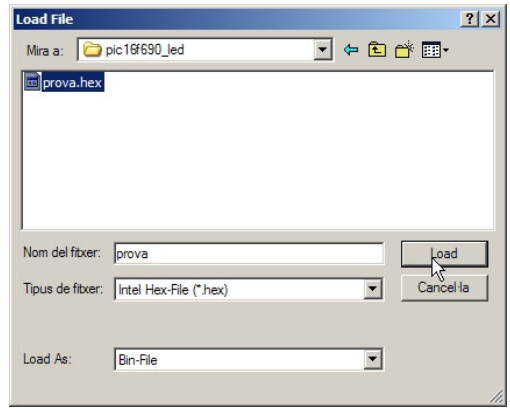
5. En la siguiente figura se puede observar el entorno de trabajo. Es importante observar que en la ventana de mensajes (abajo a la izquierda) se ha reconocido el módulo programador y que éste funciona correctamente. Al mismo tiempo, un LED de color verde se encenderá en el aparato (el inferior), indicando que está listo para realizar cualquier tarea.



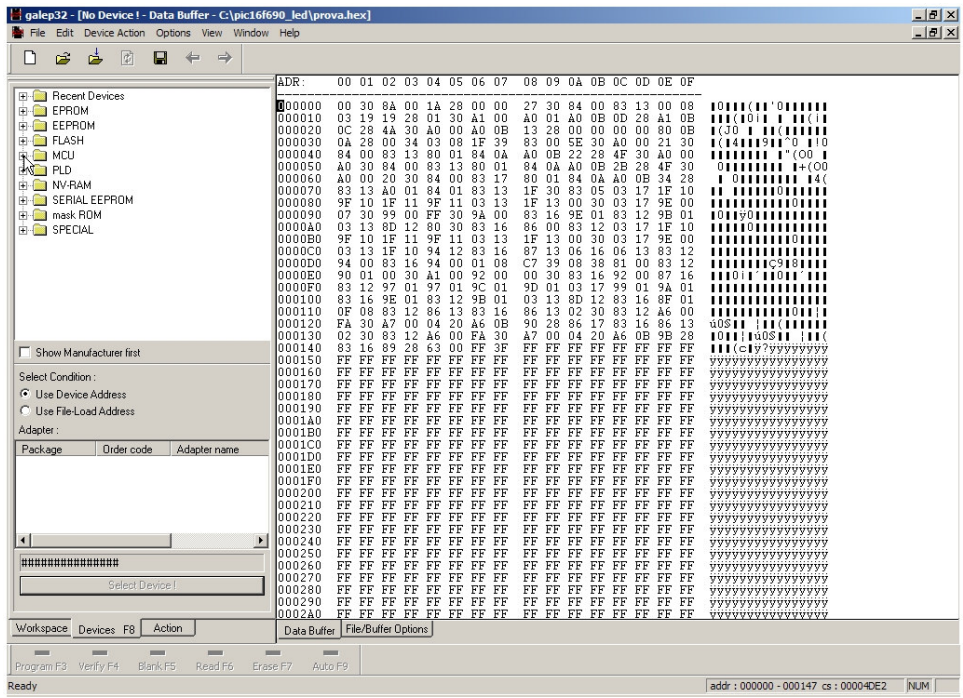
6. Vamos a cargar nuestro archivo .hex. Para ello:

Menú File > Load

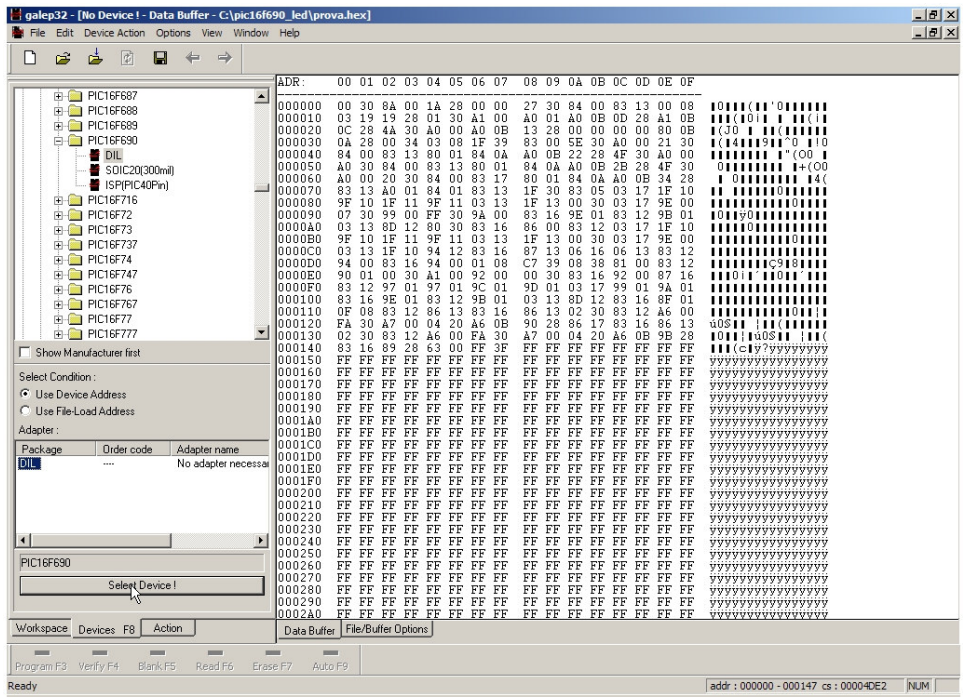
7. Seleccionamos el fichero 'prova.hex'. Deberemos indicar en 'Load As' el formato interno, esto es, 'Bin-File' (archivo en formato binario). Click en 'Load'.



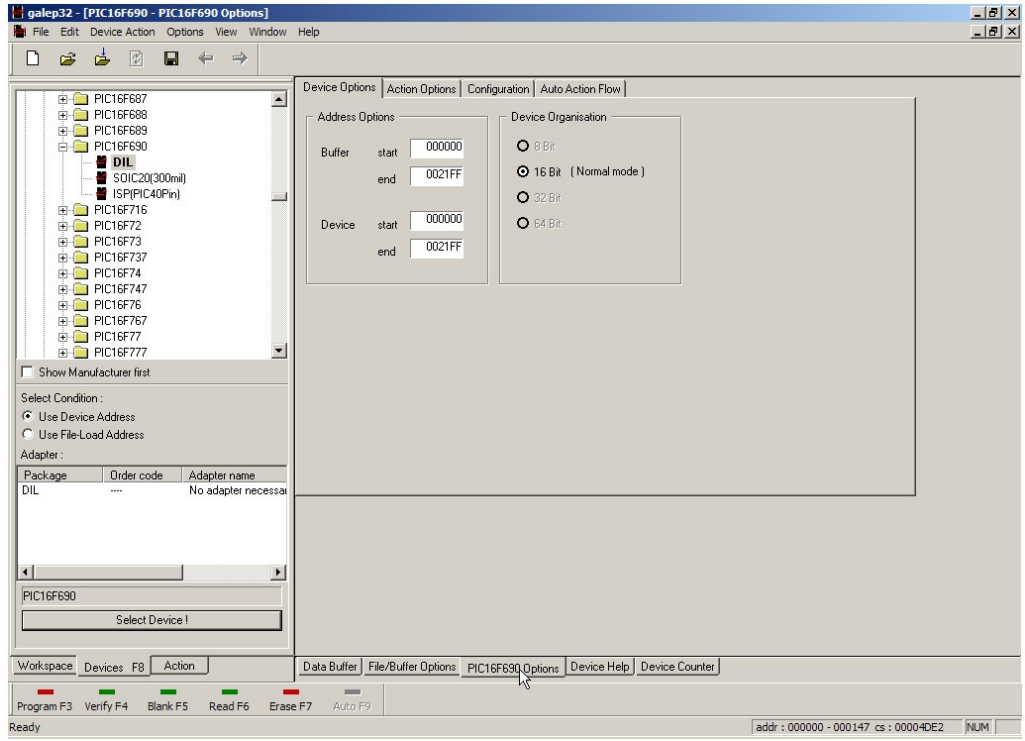
8. Click en la pestaña 'Devices'. Expandimos el árbol de dispositivos clicando en 'MCU'.



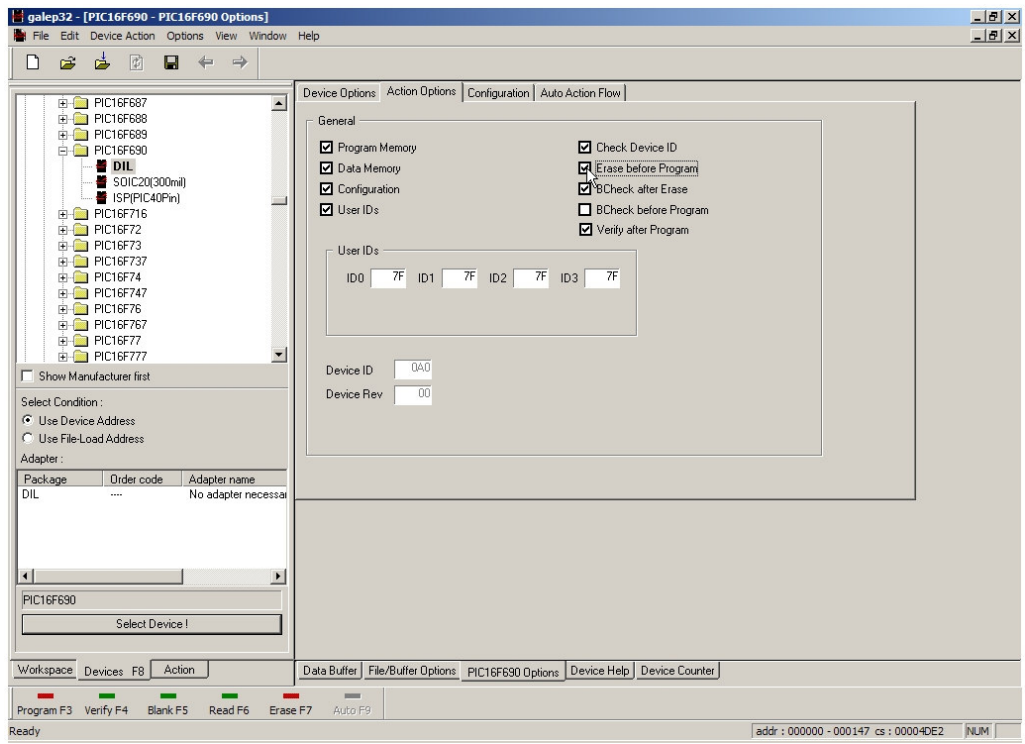
9. Buscamos la entrada 'PIC16F690'. Al clicar sobre el signo '+', se nos presentan los distintos encapsulados. Seleccionamos 'DIL' (Dual-In-Line), que es el que corresponde a nuestro micro. Finalmente clicamos en 'Select Device !'.



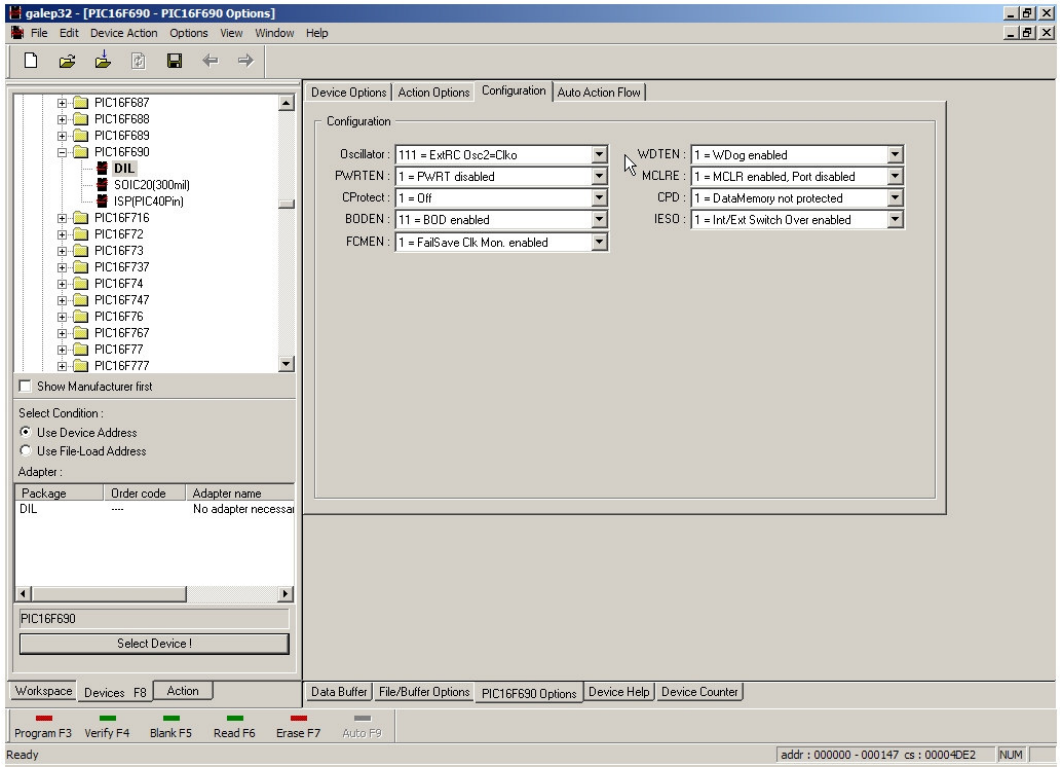
10. Click en la pestaña 'PIC16F690 Options', abajo a la derecha.



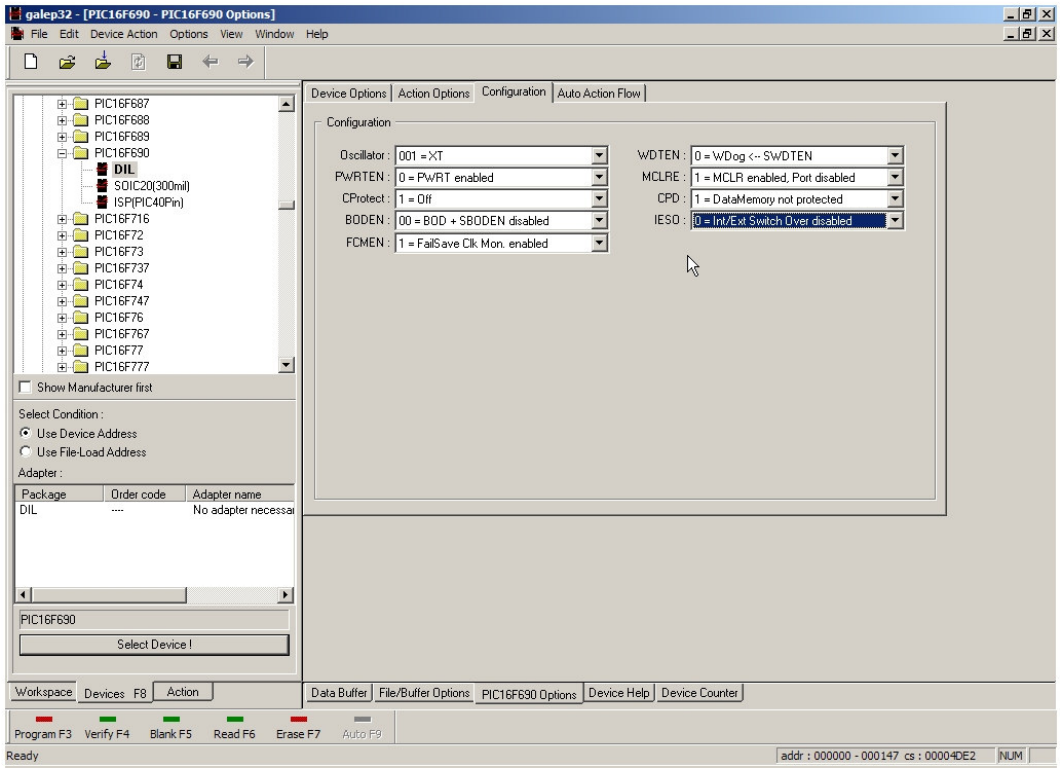
11. Click en la pestaña superior 'Action Options'. Seleccionamos la opción 'Erase before Program'.



12. Click en la pestaña ‘Configuration’. Las opciones por defecto son las siguientes:



No obstante, dejaremos la configuración tal y como sigue:



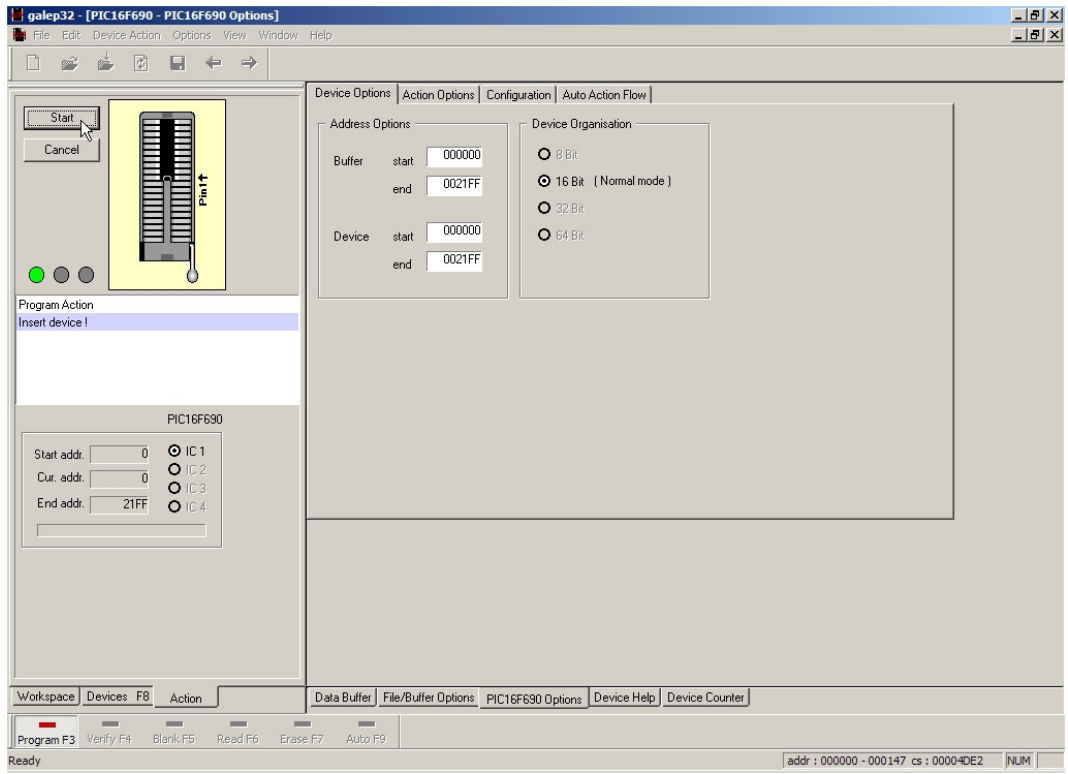
Es importante saber que las opciones seleccionadas en la figura anterior corresponden a la 'Config Word', esto es, a comportamientos específicos del micro, que en el compilador PCWH se suelen especificar mediante la directiva #FUSES.

Empezando por arriba a la izquierda, lo que hemos indicado es:

- Oscilador de tipo XT (corresponde al de 4 MHz).
- Power-Up Timer activado.
- Protección para la memoria de programa desactivada.
- Brown-Out Reset desactivado.
- FailSave Clock activado.
- WatchDog desactivado.
- Pin de reset actuando como Reset.
- Protección para memoria de datos desactivada.
- Internal/External Switch Over desactivado.

13.

- Click en la pestaña 'Actions' (parte inferior izquierda).
- Click en el botón 'Program'.

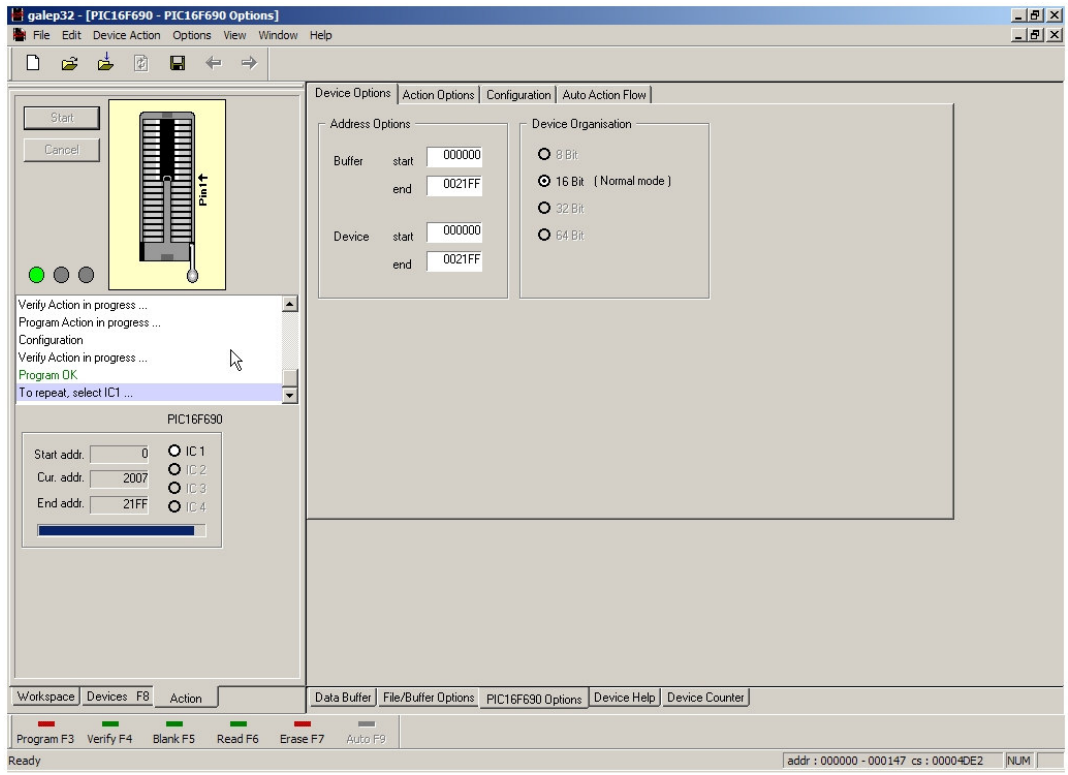


---

14. Ahora podemos insertar el microcontrolador: para ello, levantamos la palanquita del módulo, colocamos el chip y volvemos a bajar la palanca. Préstese especial atención a la figura de la página siguiente, donde se puede ver la disposición del circuito respecto al zócalo.

El pin número 1 debe quedar en la parte inferior derecha.

15. Click en 'Start'. El proceso de volcado empieza. Durante el transcurso del mismo el LED de color rojo permanece encendido. Finalizado el proceso el LED verde vuelve a encenderse. En el PC podremos observar la siguiente figura.

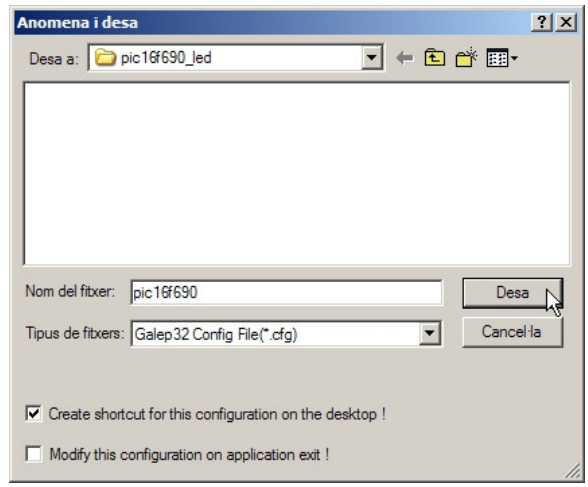


16. Ya podemos retirar el micro, levantando previamente la palanquita del zócalo.

---

17. Vamos a salvar la configuración del programador para el montaje realizado. Hacemos:

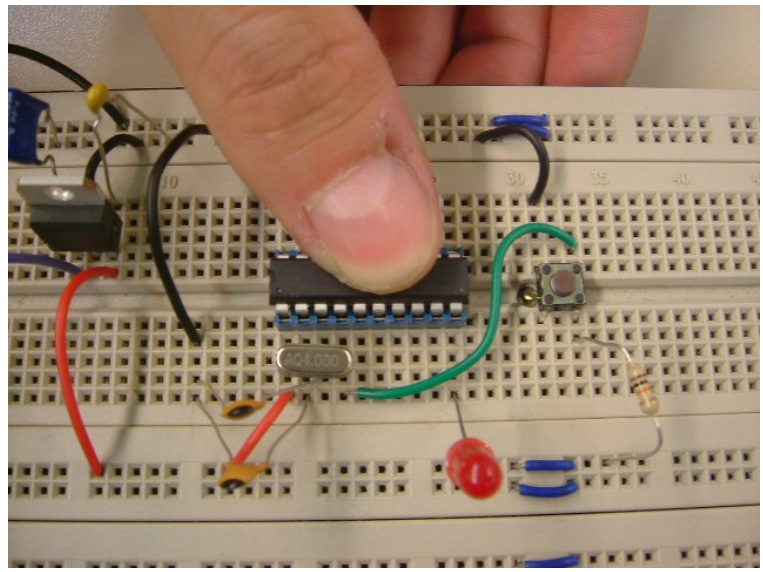
File > Save Config



En el cuadro de diálogo nombramos y ubicamos el fichero .cfg. Una opción que resulta interesante activar es 'Create shortcut for this configuration on the desktop!', gracias a la cual se nos creará un acceso directo en el escritorio para poder acceder al Galep 32 con la configuración y el fichero hexadecimal, del último montaje sobre el que hemos trabajado, ya cargados.

3.6. Comprobación del correcto funcionamiento

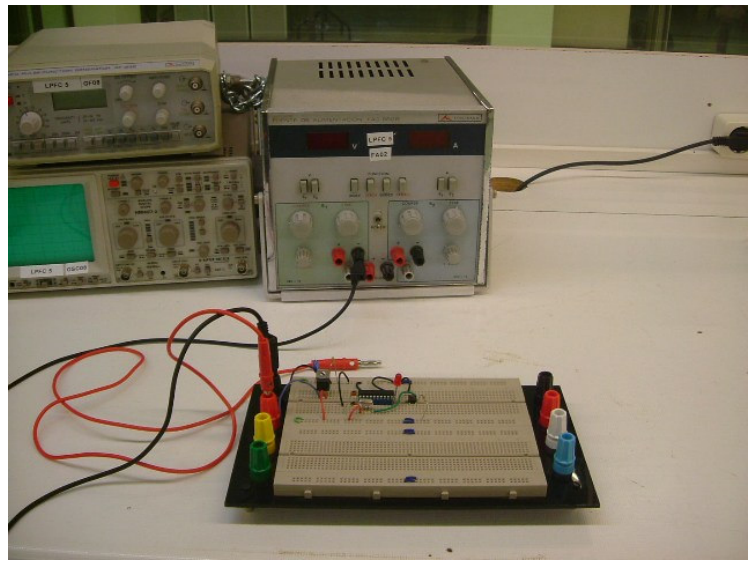
1. Insertamos el micro en la posición que le corresponde de la protoboard, asegurándonos de que queda bien fijado y las patitas hacen contacto con el sustrato metálico de la placa de pruebas.



---

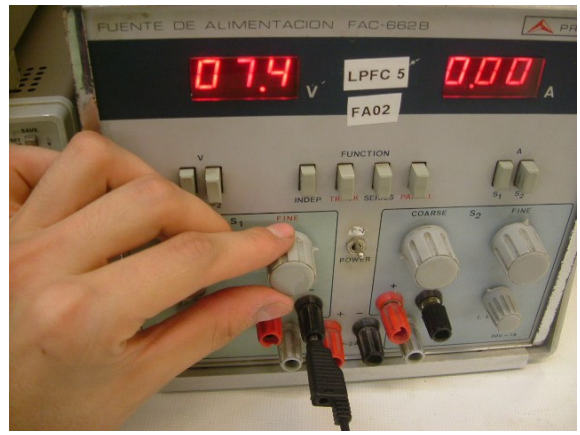
2. Conectamos los cables banana-banana a la protoboard y a la fuente de alimentación, a excepción del borne positivo que corresponde a la fuente. El porqué de proceder así radica en el pico de tensión y corriente que proporciona la fuente al encenderla. Podríamos dañar nuestro micro.

Más conveniente es conectar el positivo de la placa a la fuente una vez esté encendida.



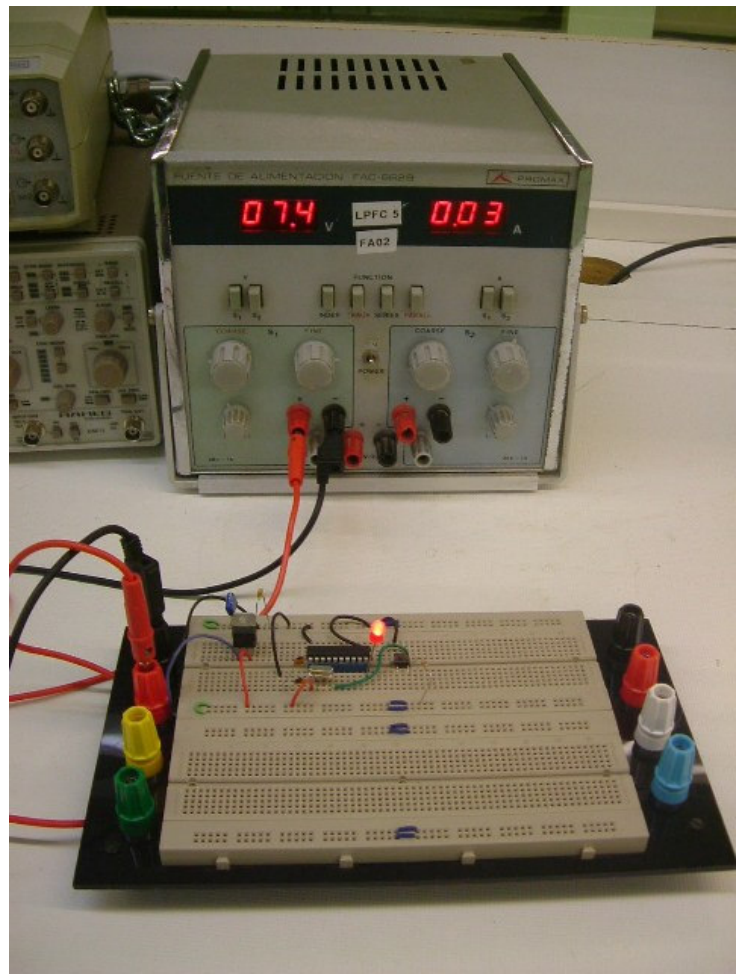
---

3. Encendemos la fuente de alimentación y seleccionamos una tensión superior a los 7 V (requisito del 7805) e inferior a 8 V.



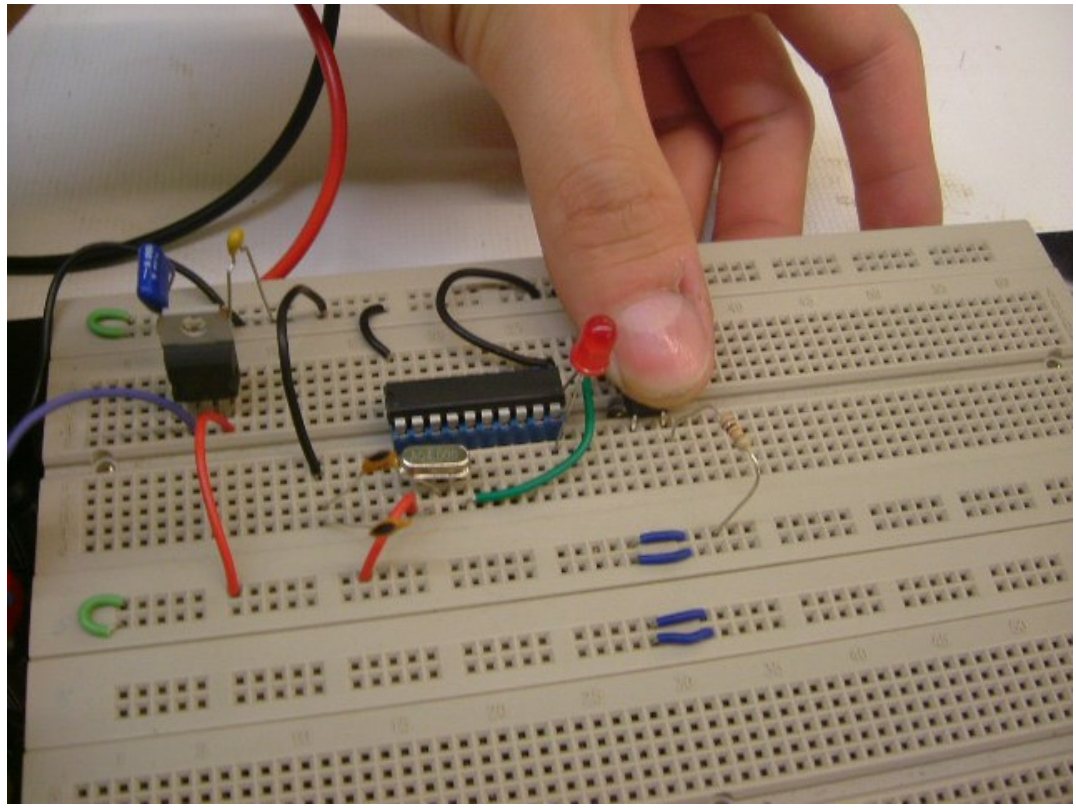
---

4. ¡Llegó el momento de la verdad! Conectamos la placa a la fuente. Si todo ha ido bien, nuestro LED se encenderá y apagará alternativamente cada medio segundo.



---

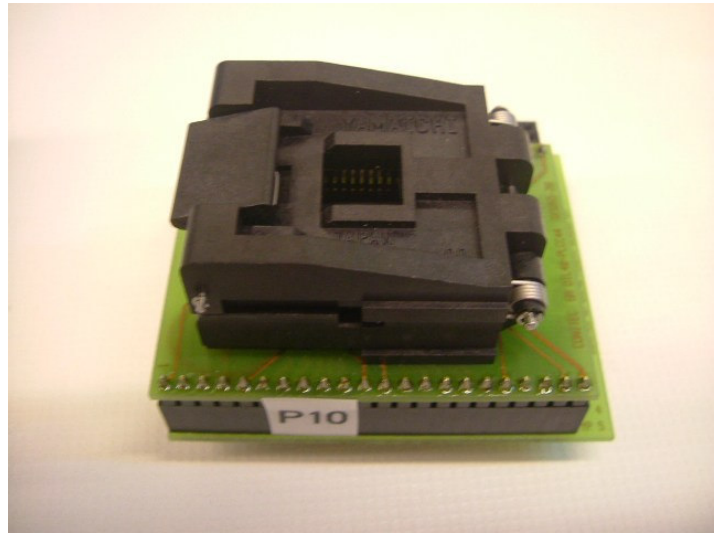
5. Finalmente comprobamos que el botón de reset realiza su función correctamente: al mantenerlo pulsado el circuito debería permanecer inactivo (el LED deja de encenderse). Al soltarlo vuelve a encenderse de forma intermitente.



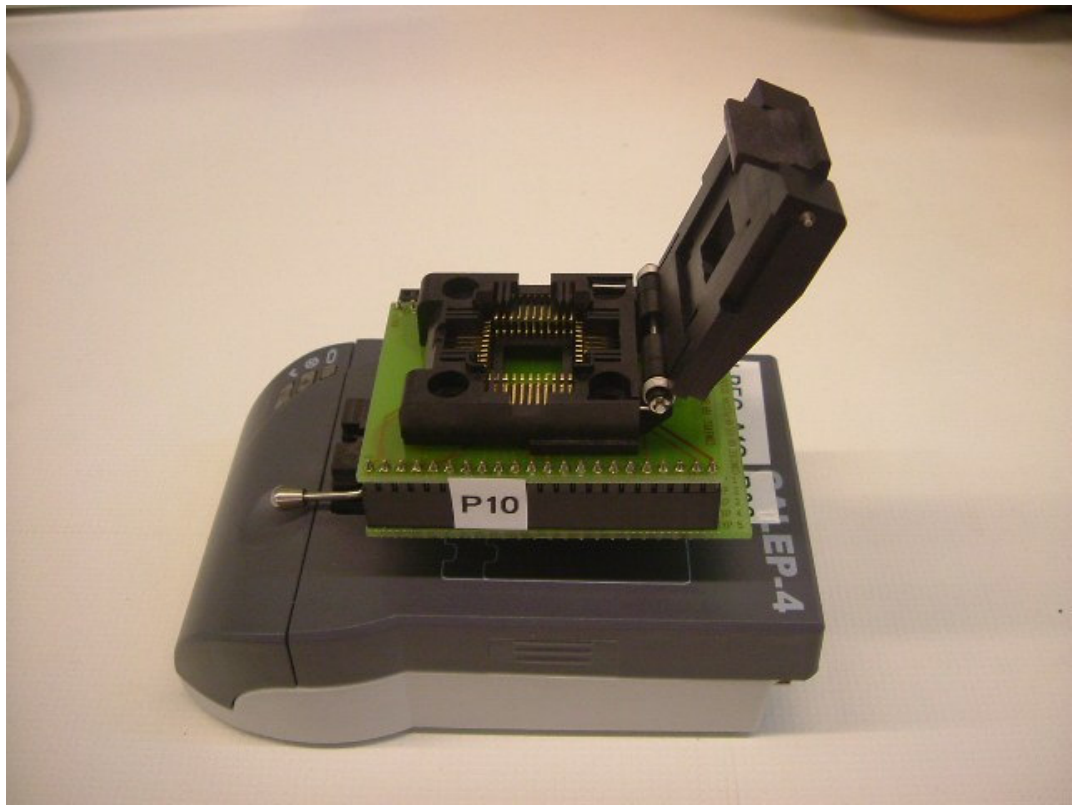
4. Apéndice: el adaptador PLCC 44 16 bit EPROM

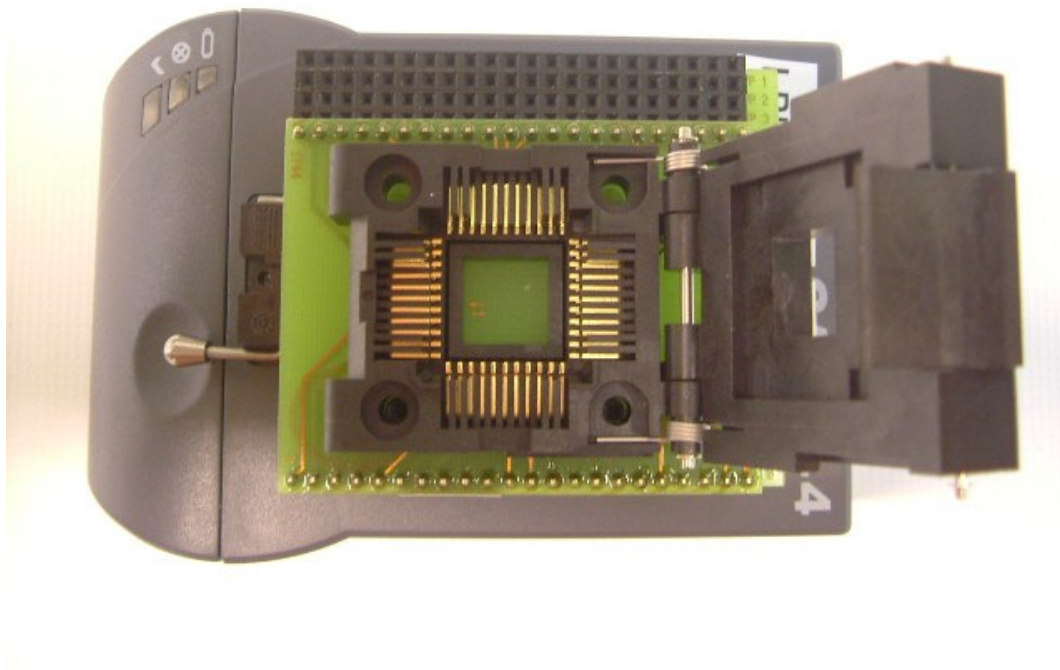
Muchos integrados vienen encapsulados en formato PLCC de 44 pines. El laboratorio de PFCs dispone de un adaptador DIL-40 <> PLCC-44 para el GALEP-4.

El aspecto del mismo es el siguiente:



Para insertarlo en el zócalo DIL del módulo programador levantaremos el bracito y colocaremos el adaptador como sigue:





Volvemos a bajar el brazo.

Cuando lo vayamos a guardar, protegeremos las patitas del adaptador con la almohadita que se proporciona: