

ISP (Programación en sistema) de los microcontroladores de NXP (Philips) 89LPC9XX utilizando “Flash Magic” y la UART (puerto serie) del microcontrolador

Patricio Coronado, SEGAINVEX-ELECTRONICA (Universidad Autónoma de Madrid) – marzo de 2013.

Para reprogramar la memoria flash de esta familia de microcontroladores, hay varios métodos. Uno, es el de hacer que el microcontrolador ejecute una rutina interna denominada bootloader, enviándole un “break condition” por el puertos serie (UART). Dicha rutina es capaz de borrar, y escribir la memoria flash de programa con un fichero recibido por el puerto serie. Para una información completa sobre este tema ver la nota de aplicación de Phillips:

AN10221 In-system programming (ISP) with the Philips P89LPC932 microcontroller.

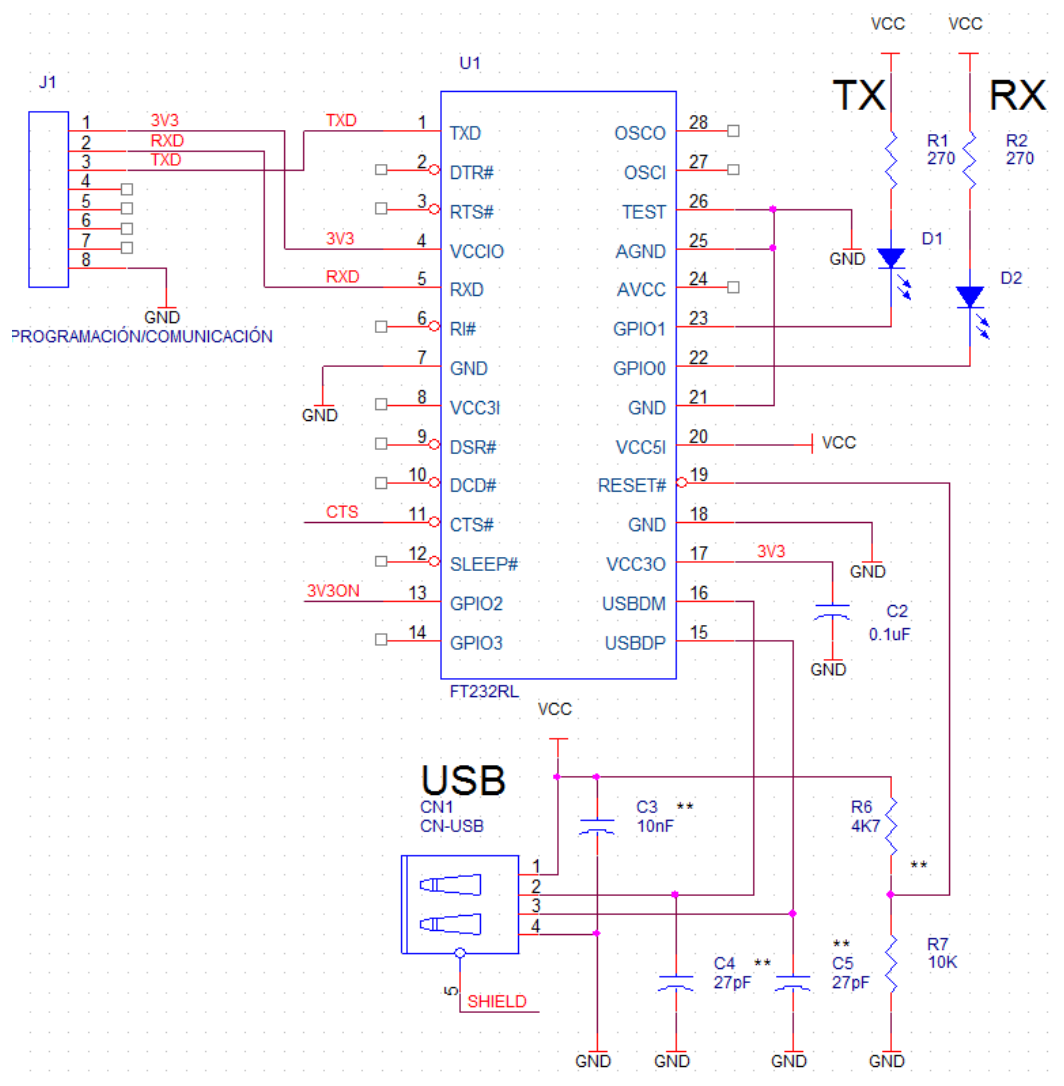


Figura 1. Circuito para comunicar el microcontrolador 89LPCXXX y el PC

1. Hardware para conectar el PC y el microcontrolador

Primero hay que comunicar el microcontrolador y PC entre sí. Para conectar la UART del microcontrolador a un puerto serie del PC necesitamos circuitería. Por ejemplo un max232, útil para adaptar niveles del microcontrolador al COM1 o 2 del PC. Si utilizamos un puerto USB del PC, podemos emplear el circuito integrado FT232RL o similar. En este caso el puerto serie (UART) del microcontrolador se conectará a un puerto COM virtual del PC.

En la figura 1 vemos un ejemplo con el circuito integrado FT232RL. La placa de circuito impreso del microcontrolador, tendrá un conector para conectar a este circuito. Como vemos en la figura2.

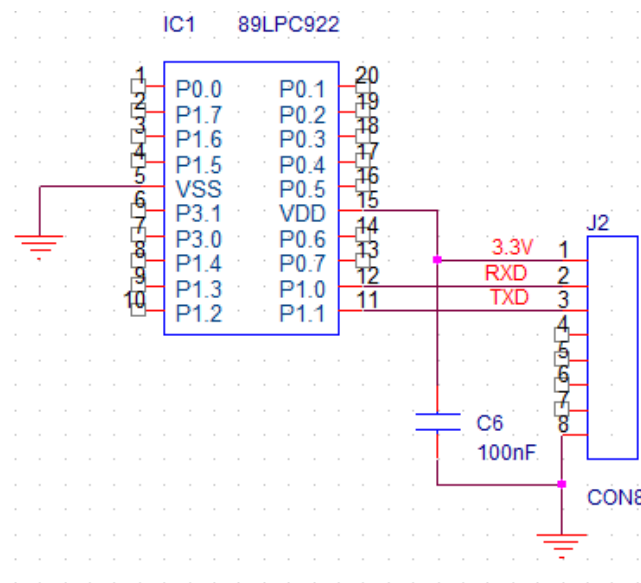


Figura 2. Conexión del microcontrolador

En este ejemplo concreto, vemos que el microcontrolador se alimenta con los 3,3V presentes en el pin 17 del integrado FT232. El pin 4 de dicho integrado también se conecta a los 3,3 voltios para que los niveles de TXD y RXD sean también de 3,3V. Aunque el microcontrolador soporte niveles de 5V en estos pines, a pesar de estar alimentado con 3,3V.

2. Software para programar la memoria flash del microcontrolador

Para programar la memoria flash del microcontrolador desde el PC, utilizaremos el programa de NXP (Philips) Flash Magic. Está disponible de forma gratuita en:

<http://www.flashmagictool.com/download.html>

Una vez bajado e instalado, lo ejecutamos. Vemos la ventana de la figura 3.

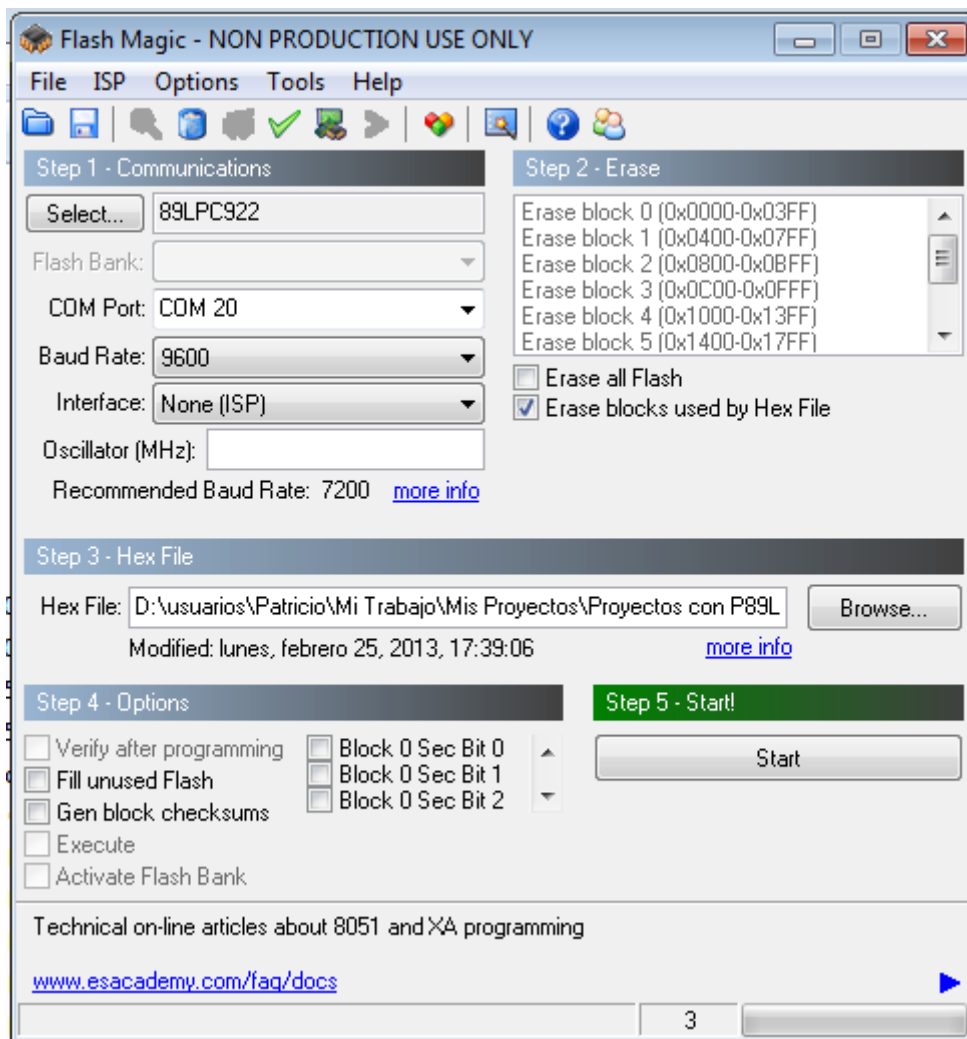


Figura 3

En la parte de "Step 1- Communiations" , con el botón de "Select.." seleccionamos el microcontrolador que estemos utilizando. En la lista desplegable "COM Port:" Seleccionamos el puerto virtual en el que se ha instalado el COM que comunica con el integrado FT232RL, o el puerto COM físico con el que estemos conectando al microcontrolador. El "Baud Rate" lo ponemos en principio a 9600 y si hay problemas de conexión lo bajamos. En "Interface" seleccionamos "None (ISP)".

En la parte de "Step 2 - Erase" Marcamos la casilla "Erase blocks used by Hex File"

En la parte de "Step 3 - Hex File" Buscamos con el botón "Brouse..." el fichero de nuestro programa con extensión .hex . También se puede acceder al fichero de programa a través del menú "File" y "Open..." .

El resto se deja como se ve en la figura 3.

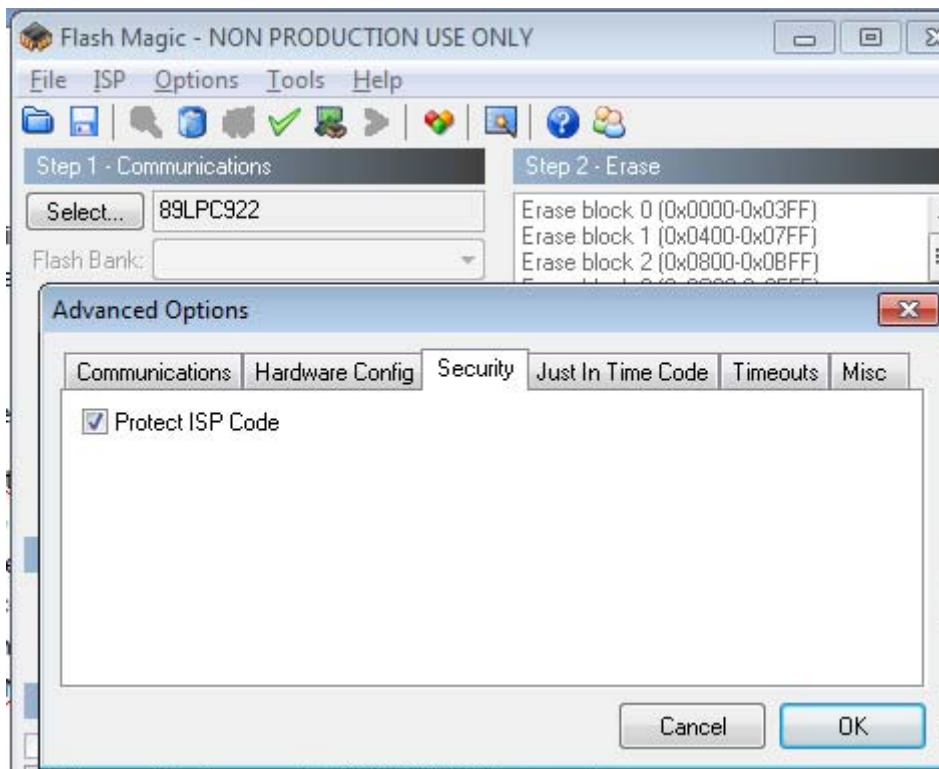


Figura 4

Una precaución básica, es evitar que el sector de la flash del microcontrolador donde reside la rutina de “bootloader” se borre o re programe. Para eso en “Options” “Advanced Options” marcamos la casilla “Protect ISP Code”. Figura 4.

3. Programando el microcontrolador por primera vez

El microcontrolador tiene un bit en una zona reservada de memoria, denominado “Status Bit”, que de fábrica está a 1. Cuando se conecta por primera vez el microcontrolador y debido a que el “Status Bit” está a 1, se ejecuta una rutina interna denominada “bootloader” que nos permite borrar y escribir la memoria de código, pudiendo así grabar el microcontrolador con el programa del usuario. Una vez programado el microcontrolador, el “Status Bit” se pone a 0. De modo que tras un reset, se ejecutará el código de usuario.

Si el microcontrolador es nuevo, podemos verificar que se está ejecutando el bootloader, con Flash Magic, pulsando “ISP” y “Read Security...”. Flash Magic lee el microcontrolador y nos muestra el estado de los bits de seguridad, como se ve en la figura 5. Aquí es donde Flash Magic nos puede presentar un pop-up informando que no ha podido conectar con el microcontrolador. Entonces hay que bajar los baudios hasta que consiga conectar. En el ejemplo conectaba mejor bajando el baud rate a 4800 bps.

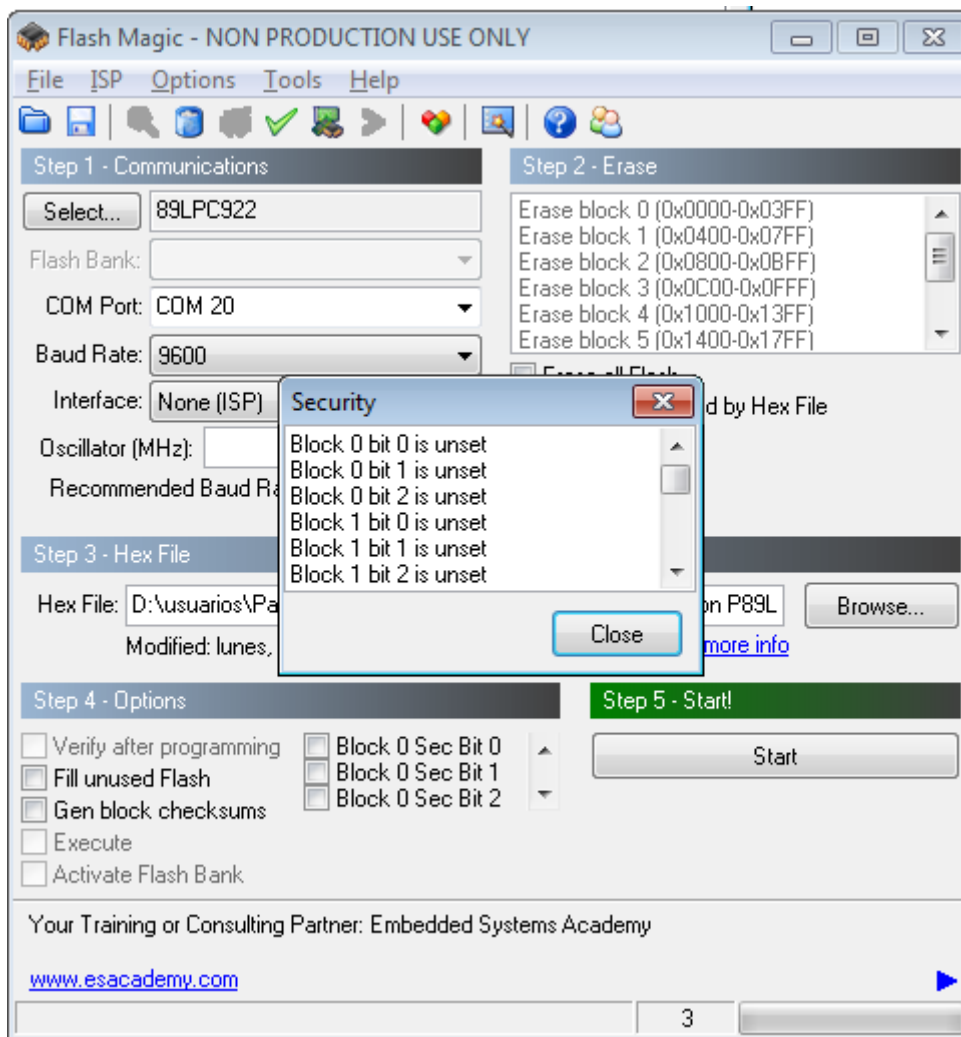


Figura 5

Si llegamos a este punto, podemos pulsar "Start" y Flash Magic programará el microcontrolador. Figura 6. En el caso del microcontrolador del ejemplo, al terminar de programar, Flash Magic, lanza un pop-up indicando que alguna operación no está soportada, pero si lo ha programado.

La próxima vez que se alimente el microcontrolador (o después de un reset), el "Status Bit" estará a 0 y ya no ejecutará el bootloader, sino que ejecutara nuestro programa.

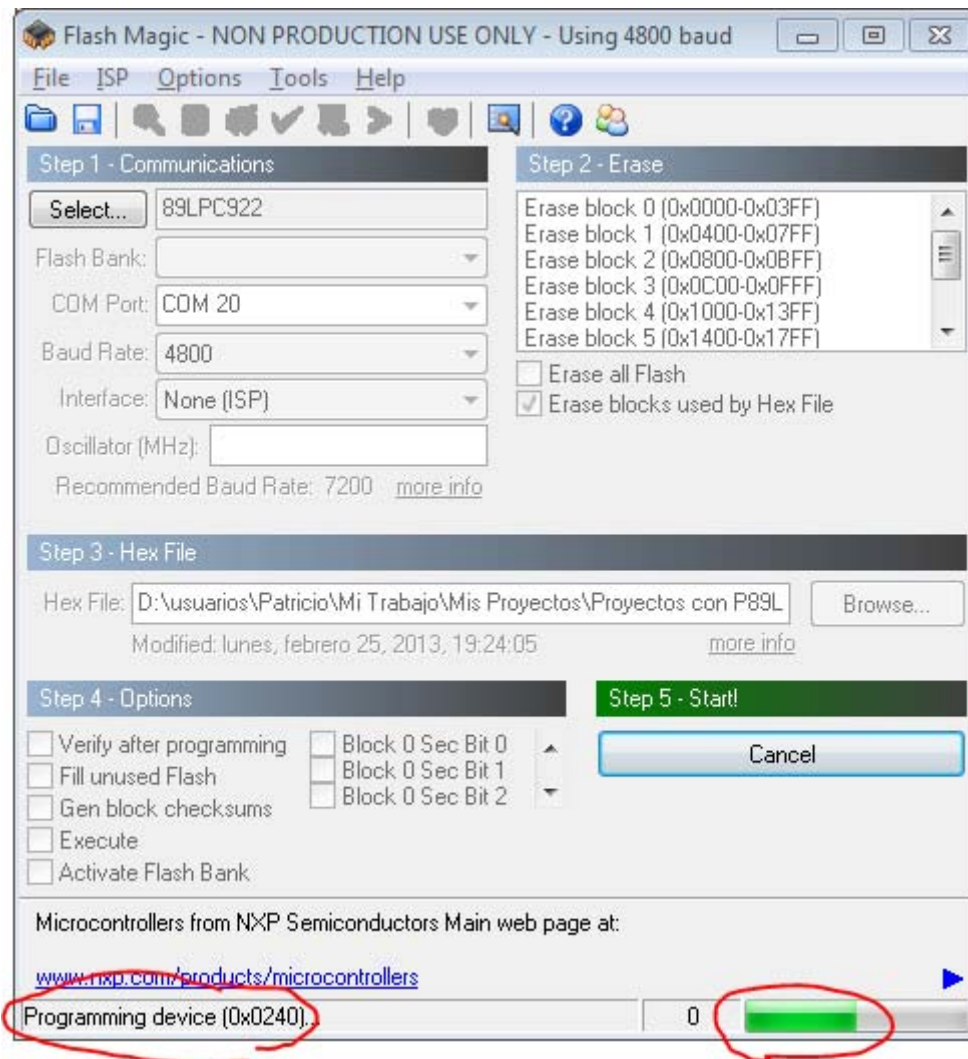


Figura 6

4. Reprogramando el microcontrolador

Si queremos reprogramar el microcontrolador, hay que hacer que este ejecute la rutina interna “bootloader”.

Uno de los métodos (el que usaremos aquí) para conseguir que el microcontrolador entre de nuevo en la rutina del bootloader, es el de enviarle un “Break Condition” a través del puerto serie del microcontrolador. Pero (y esto es muy importante) para que el microcontrolador reconozca un “Break Condition” tiene que tener habilitado y configurado el puerto serie. Trataré esto en el siguiente apartado.

Para enviar al microcontrolador un “Break Condition” por el puerto serie, en Flash Magic, pulsamos el menú “ISP” y “Start Bootloader...”. Figura 7.

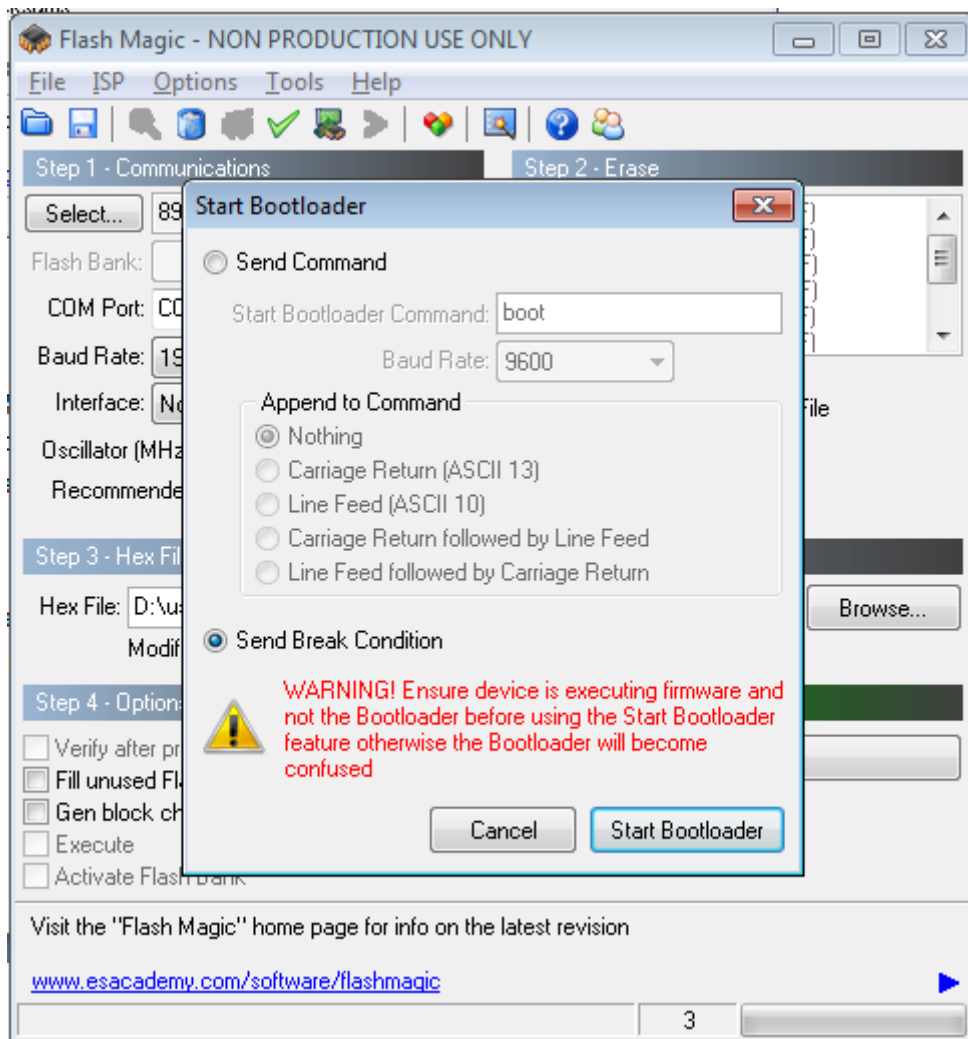


Figura 7

Pulsando el botón “Start Bottloader”, Flash Magic envía un pulso bajo de unos 270ms al pin de RXD del microcontrolador. Tarda unos segundos. Después de esto el microcontrolador estará ejecutando el bootloader. Podemos comprobarlo leyendo los bits de seguridad como hicimos en el apartado anterior. Ya podremos reprogramar el dispositivo, leyendo el fichero de programa pulsando en el menú “File” y “Open...”. Finalmente pulsamos “Start” .

5. Inicializar la UART (puerto serie) del microcontrolador

Como dije en el apartado anterior, para que todo esto funcione, nuestro programa de usuario que está corriendo en el microcontrolador debe haber configurado y habilitado el puerto serie. Un ejemplo de código para inicializar la UART es la función:

```
void uart_init(void) { // Configura la UART
```

```
//SM0 y SM1 configurar en modo 1
```

```

PCON &= ~0x40; // Pone el PCON.6=SMOD0 a 0 para acceder al SM0 y SM1 del SCON
SCON = 0x50; // (SM0=0 SM1=1 UART modo 1) & (habilita la recepción SCON.4=REN=1)
SSTAT = 0x00; // Registro de estatus de la UART (varias funciones)
AUXR1 |= 0x40; // habilita la recepción de un "break detect"
// configura el "baud rate generator" CCLK=7.373MHz y 9600 baudios
BRGCON = 0x00; // Acceso a BRGR0/1 habilitado
BRGR0 = 0xF0; // para conseguir 9600
BRGR1 = 0x02;
BRGCON = 0x03; // Acceso a BRGR0/1 inhabilitado
// TxD = push-pull, RxD = input configuración de pines de comunicación
P1M1 &= ~0x01;
P1M2 |= 0x01;
P1M1 |= 0x02;
P1M2 &= ~0x02;
} // uart_init

```

No es necesario configurar el pin de transmisión TXD, si solo se usa la UART para recibir una "break condition". Pero si hay que tener la precaución de que RXD sea una entrada.

En este caso el CLK del sistema es la RC interna del microcontrolador, de 7.373MHz. Para otra fuente de CLK de frecuencia distinta, hay que programar los registros BRGR0 y BRGR1 con valores distintos (consultar la "user guide del microcontrolador").

También hay que tener en cuenta que si la UART se pone en "power down" o el sistema entra en "power down", la UART queda deshabilitada.

Para ver en detalle cómo se configura el puerto serie revisar la guía de usuario del microcontrolador. Para generar código que configura el puerto serie (y otros recursos), ver la web:

http://www.codearchitect.org/nxp/v2/index.php?co_tree=

6. Como repone la rutina del bootloader si se ha borrado

La rutina del "bootloader" está en un sector de la memoria flash del microcontrolador. Si este sector se borra o se reescribe, perderemos la posibilidad de reprogramar el microcontrolador de la forma vista hasta ahora. Una solución a esto es reprogramarle el "bootloader" (que te puedes bajar de la web de NXP Semiconductors) con un programador paralelo.

Para una explicación complete de esto, ve la nota de aplicación AN10337 de NXP Semiconductors "Adding ISP firmware to an LPC900 software Project"

El código del bootloader está en:

http://www.nxp.com/documents/other/Adding_ISP_in_source_code.zip

En este fichero, además del código que necesitamos también está la nota de aplicación citada.

Al descomprimir podemos ver los ficheros de varias versiones de bootloaders (Figura 8).

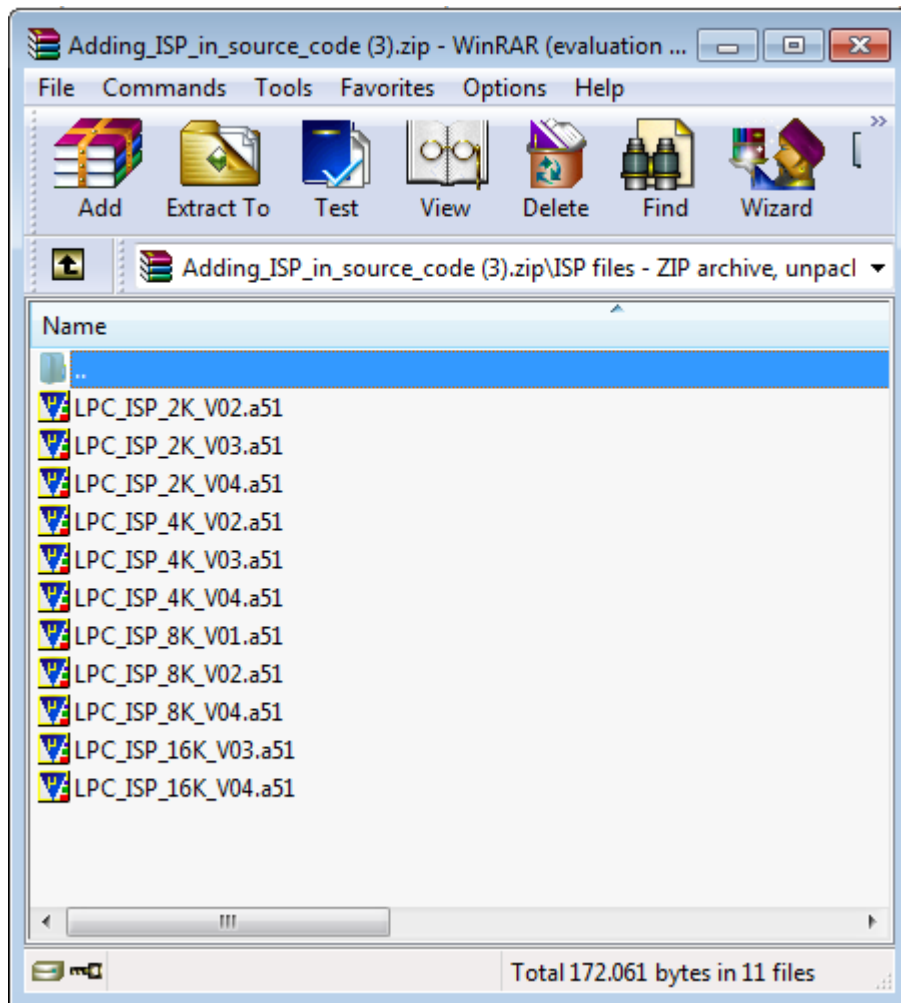


Figura 8

Para saber cual es la de nuestro microcontrolador en concreto, hay que recurrir a la nota de aplicación antes mencionada y buscar la tabla de la figura 9:

Table 1. ISP files used by LPC900 parts

Part number	ISP version	ISP sector location	Memory size	File name
P89LPC920	Version 2	Upper half of sector 1	2 kB	LPC_ISP_2K_V02.a51
P89LPC9201	Version 4	Upper half of sector 1	2 kB	LPC_ISP_2K_V04.a51
P89LPC921	Version 2/Version 4 ^[1]	Upper half of sector 3	4 kB	LPC_ISP_4K_V02.a51
P89LPC9211	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC922	Version 2/Version 4 ^[1]	Upper half of sector 7	8 kB	LPC_ISP_8K_V02.a51
P89LPC9221	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC922A1	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC924	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC9241	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC925	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC9251	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC930	Version 2	Upper half of sector 3	4 kB	LPC_ISP_4K_V02.a51
P89LPC9301	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC931	Version 2/Version 4 ^[1]	Upper half of sector 7	8 kB	LPC_ISP_8K_V02.a51
P89LPC931A1	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC932A1	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC9321	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC933	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC9331	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC934	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC9341	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC935	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC9351	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC936	Version 4	Upper quarter of sector 7	16 kB	LPC_ISP_16K_V04.a51
P89LPC938	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC9381	Version 4	Upper half of sector 3	4 kB	LPC_ISP_4K_V04.a51
P89LPC9401	Version 2	Upper half of sector 7	8 kB	LPC_ISP_8K_V02.a51
P89LPC9402	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC9408	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC952	Version 4	Upper half of sector 7	8 kB	LPC_ISP_8K_V04.a51
P89LPC954	Version 4	Upper half of sector 15	16 kB	LPC_ISP_16K_V04.a51

Figura 9

En el caso particular de este ejemplo el microcontrolador es el resaltado en amarillo.

Una vez que tenemos el código, tenemos que iniciar un proyecto en nuestro compilador e incluirlo. En mi caso para crear y compilar proyectos para microcontroladores, utilizo la versión 4 del uVision de Keil.

Abrimos el uVision, en “Project” seleccionamos “New uVision Project...”, le damos un nombre al proyecto y lo guardamos. Seguidamente se nos pide seleccionar el microcontrolador (Figura 10). Seleccionamos el nuestro

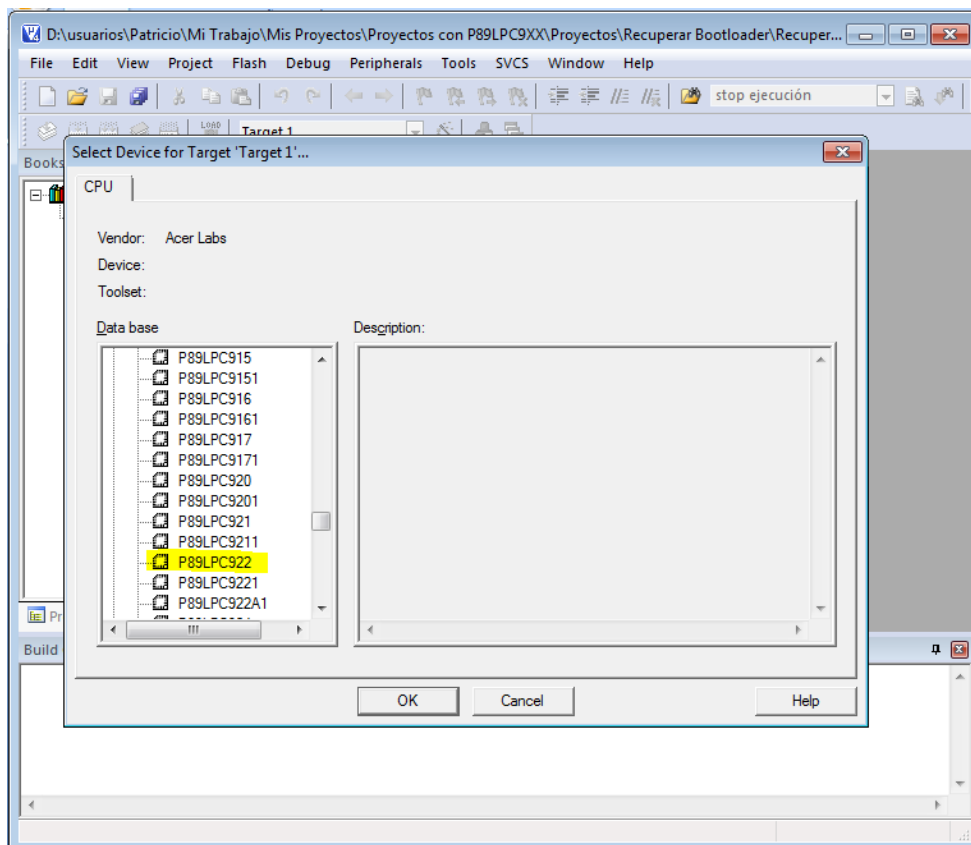


Figura 10

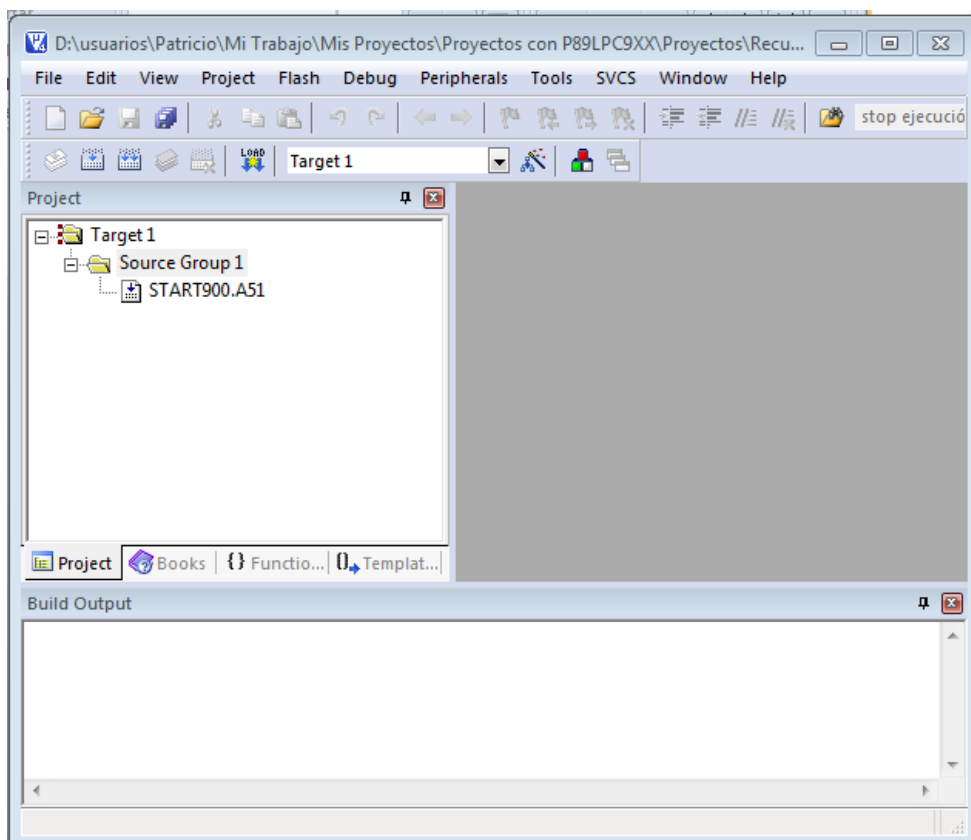


Figura 11

. Nos preguntará si queremos incluir el código Startup en nuestro proyecto. Le contestamos que sí. Pulsando en la pestaña “Project” abajo a la izquierda, vemos nuestro proyecto. Figura 11.

Ahora tenemos que copiar el código del bootloader de nuestro microcontrolador en la carpeta del proyecto que acabamos de crear. Luego, en el uVision picamos con el botón derecho del ratón sobre “Source Group 1” y seleccionamos “Add Files to Group Source Group 1...” y buscamos el fichero con extensión a51 donde está el código del bootloader. Para que se nos muestren todos los ficheros del directorio de nuestro proyecto, en “Tipo:” hay que seleccionar “All files”. Ahora ya vemos el fichero, lo seleccionamos, pulsamos “Add” y “Closed”. Ya está el fichero incluido en nuestro proyecto.

Seguidamente seleccionamos “File” y “New...” del menú. Así creamos un fichero nuevo, en el que escribimos una función “main” sin nada.

```
void main(){}
```

Salvamos el fichero con el nombre “main.c” y lo incluimos en el proyecto, como hicimos antes.

Aún quedan un par de pequeños (pero importantes detalles).

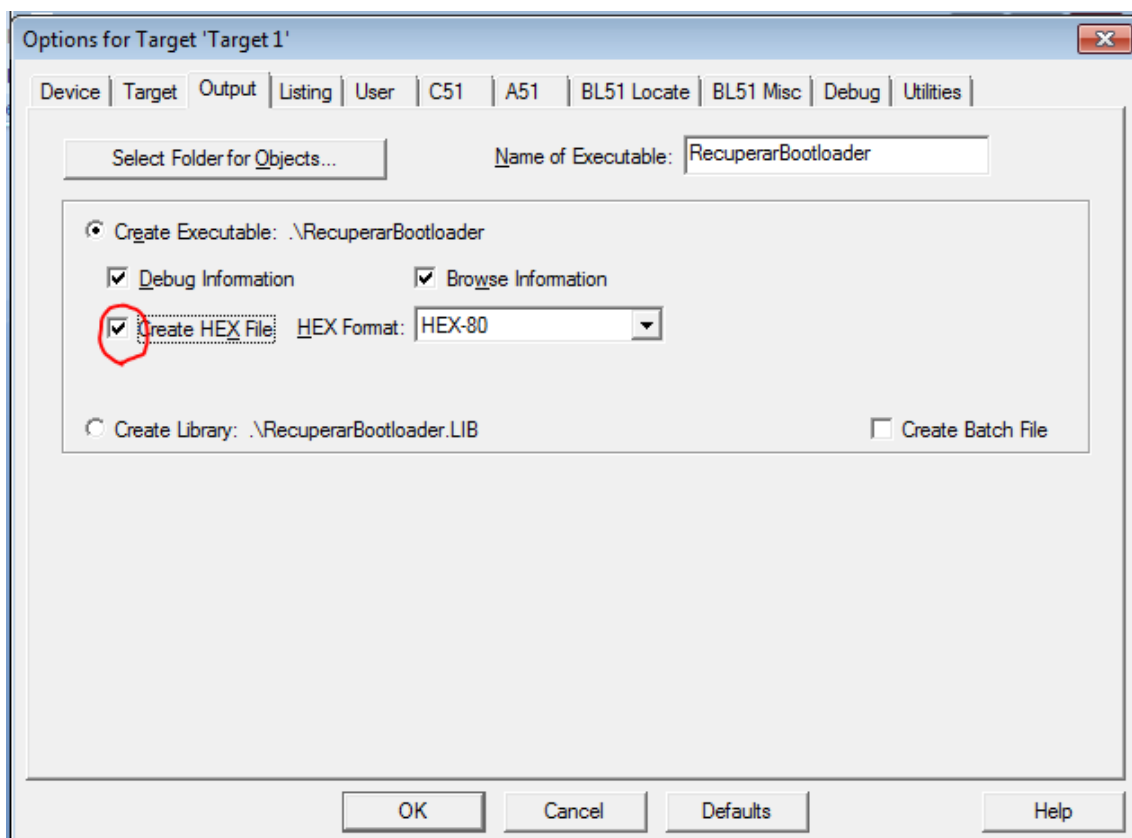


Figura 12

En el menú principal, en “Project” seleccionamos “Options for Target ‘Target1’...”. En el pop-up que aparece seleccionamos la pestaña “Output”. Marcamos “Create HEX File”. De este modo al compilar nos creará el fichero .HEX con el nombre que le demos en el text box “Name Executable”. En nuestro caso RecuperarBootloader.hex. Figura 12.

Cerramos y volvemos a la ventana principal de nuestro proyecto.

Ahora picamos dos veces seguido sobre START900.A51 y veremos este fichero. En las pestañas de abajo picamos en “Configuracion Wizard”. Y desplegamos las opciones de “Boot status”. Marcamos “Boot Status bit enable”. De esta manera al programar el microcontrolador, el “status bit” (ver apartado 3) se restablece y se ejecutará el “bootloader”. Figura 13.

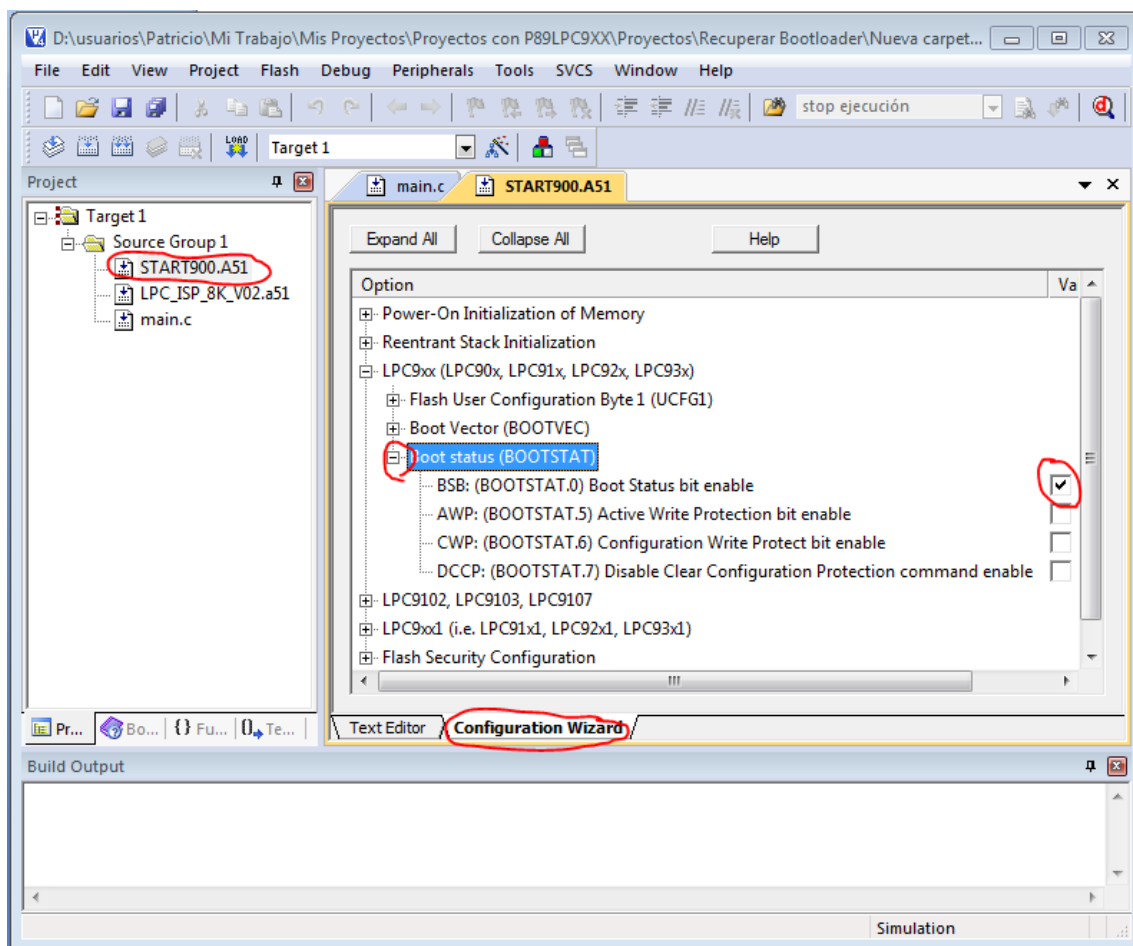


Figura 13

Ya podemos compilar el proyecto pulsando el icono como se ve en la figura 14

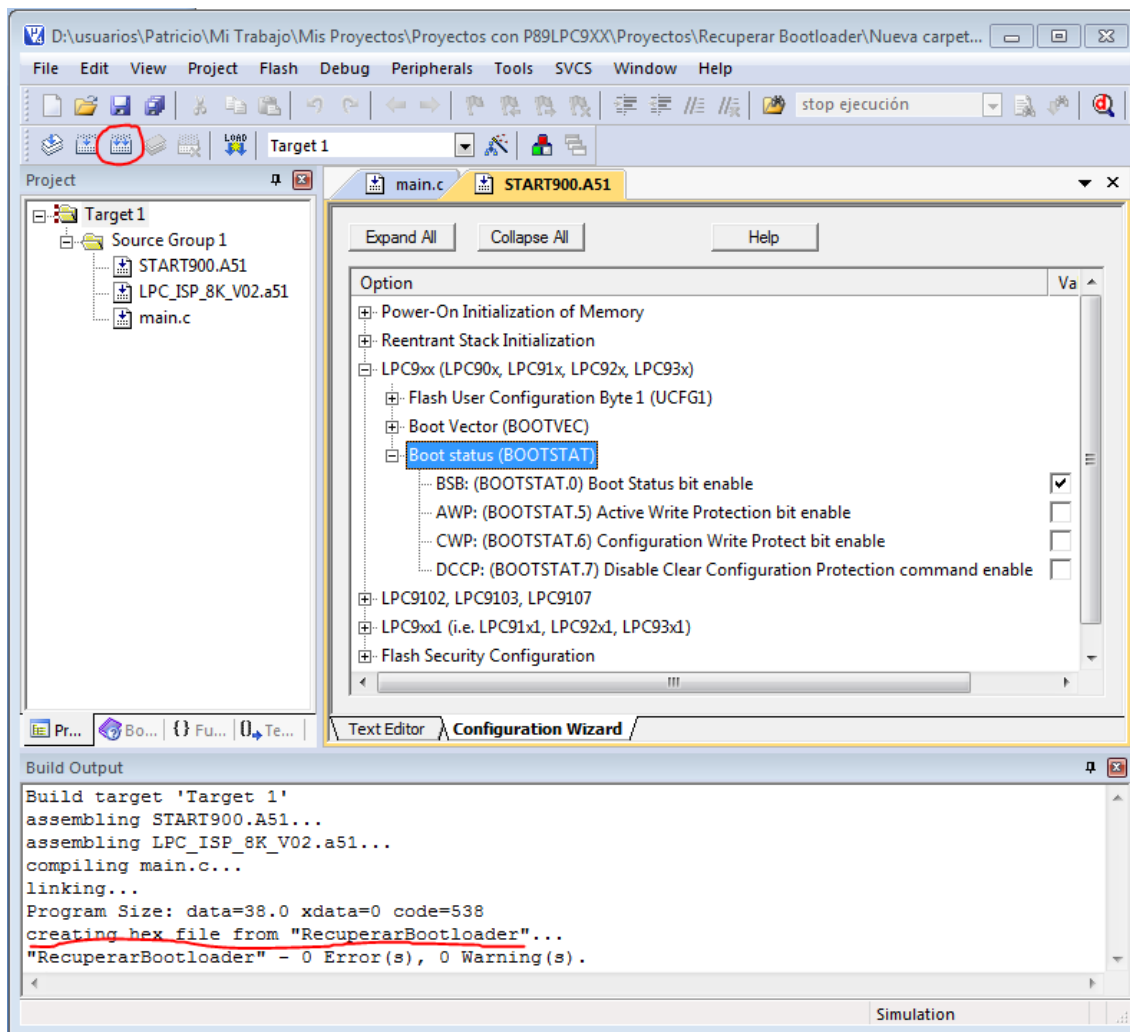


Figura 14

El proyecto se compila y se genera el fichero RecuperarBootloader.hex. Ahora hay que programar el microcontrolador con este fichero con un programador paralelo.

7. Programar el microcontrolador con un programador paralelo

En nuestro laboratorio tenemos un programador Dataman-40Pro.

Para programar el microcontrolador abrimos el software del programador y pulsando F3 se abre la ventana de "Load File". Buscamos nuestro fichero RecuperarBootloader.hex y lo abrimos.

Al picar el icono de programar se abre un pop-up de opciones, donde hay que asegurarse de que el sector del bootloader se va a programar. Figura 15.

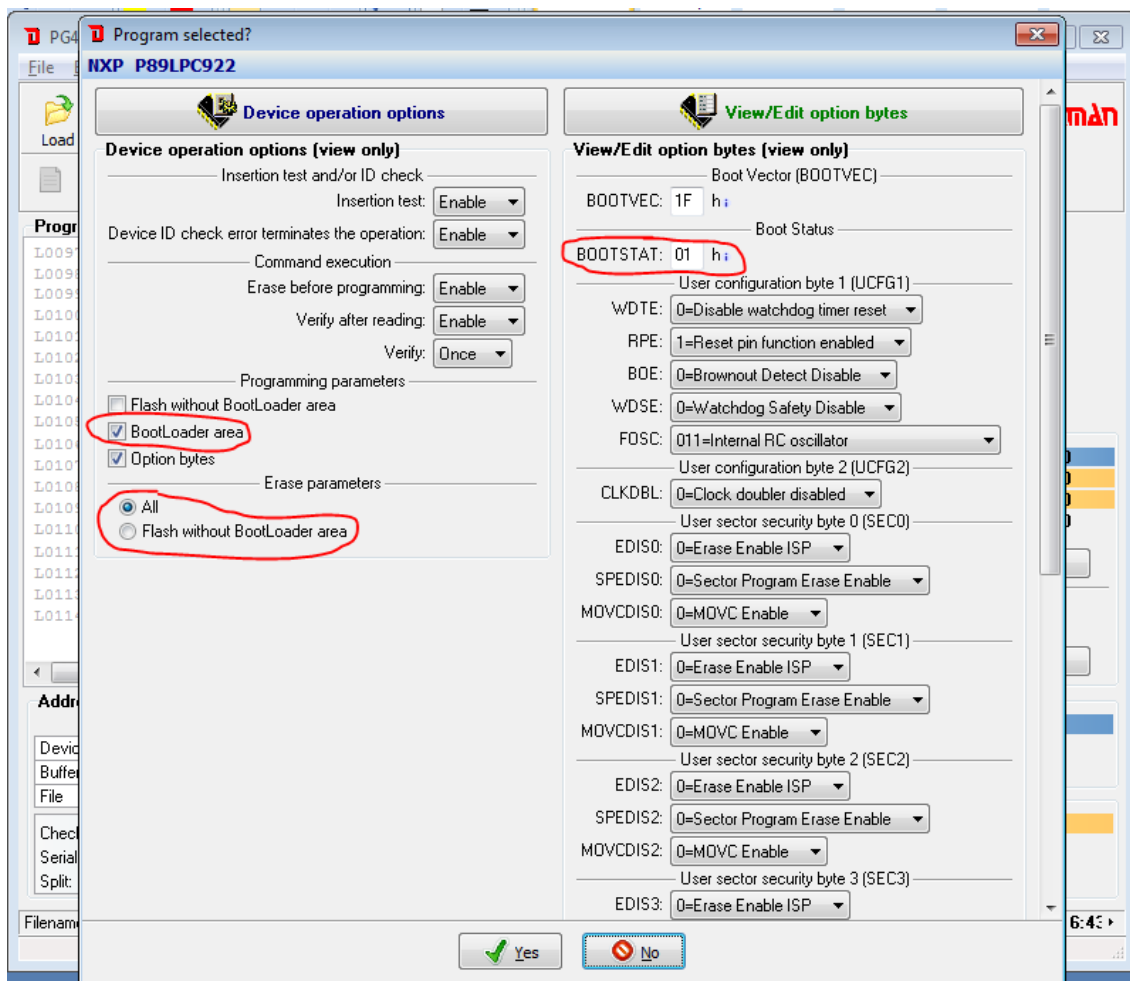


Figura 15

También vemos que el "status bit" se va a reponer.

Solo queda pulsar "Yes" y esperar un instante a que se cargue el programa en la flash del microcontrolador. Este estará como recién comprado. Para comprobarlo, ver el apartado 3.

También podríamos haber incluido el código del bootloader en el proyecto con el que estemos trabajando, compilarlo y programarlo todo junto.