

Guía de configuración del adaptador JLINK para carga y depuración

Versión 3.1

Sistemas Electrónicos Digitales

Universidad de Alcalá
Departamento de Electrónica

1 Introducción al documento

En esta guía se muestra el proceso de configuración del adaptador JTAG para la carga y depuración de programas *on-chip*. En concreto, el adaptador JTAG cuya instalación se mostrará es el J-LINK de Segger.

2 Índice de contenidos

1	INTRODUCCIÓN AL DOCUMENTO.....	3
2	ÍNDICE DE CONTENIDOS.....	3
3	INTRODUCCIÓN	5
4	INSTALACIÓN DEL CONTROLADOR DE USB DEL ADAPTADOR J-LINK.....	5
5	CONFIGURACIÓN DEL ENTORNO DE DESARROLLO μ VISION4 DE KEIL.....	6
6	PROGRAMA EJEMPLO	11

3 Introducción

En esta guía se abordará fundamentalmente la configuración del entorno de depuración Keil para posibilitar la utilización del adaptador J-Link de Segger para la depuración de programas. Esta configuración tiene dos partes, por un lado, la carga del controlador del dispositivo J-Link conectado al ordenador a través de un puerto USB, y por otro lado, la configuración del programa Keil para el empleo del adaptador J-link en la depuración de programas on-chip. En el último apartado de la guía se propone un programa ejemplo para que el usuario pueda comprobar que el adaptador ha sido instalado y configurado correctamente.

4 Instalación del controlador de USB del adaptador J-Link.

Antes de utilizar el adaptador J-Link de Segger se debe instalar el controlador del dispositivo para USB. Para ello basta con ejecutar el fichero *DPIInst* que se encuentra en la ruta: Keil/ARM/Segger/x86, tal y como se muestra en la figura 1. Este fichero contiene el paquete de instalación para Windows XP y Windows 2000. Para versiones anteriores de Windows debe emplearse el instalador contenido en la carpeta x64.

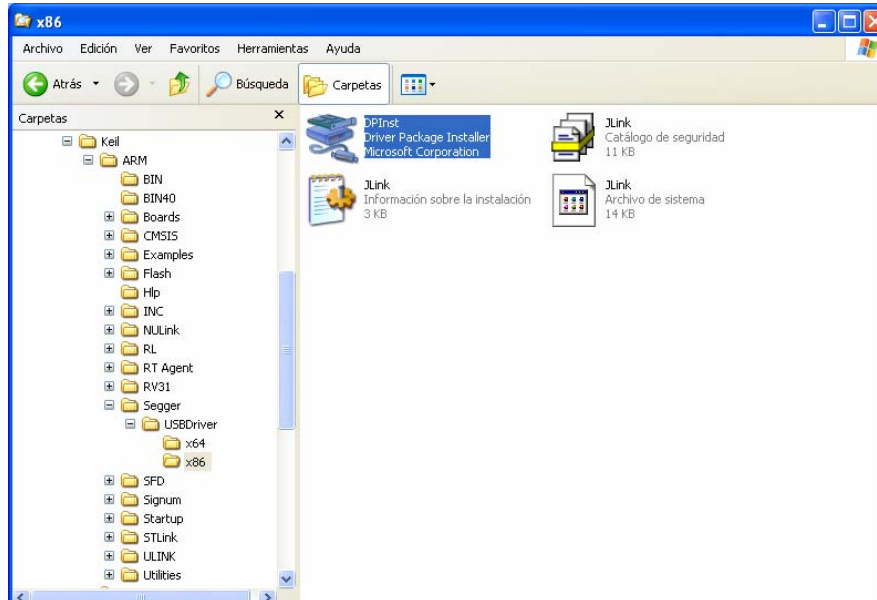


Figura 1. Ruta de acceso al instalador del controlador USB para el J-Link de Segger.

5 Configuración del entorno de desarrollo μ Vision4 de Keil.

Antes de iniciar el proceso de depuración *on-chip*, debemos tener el proyecto a depurar cargado en el entorno de desarrollo. El entorno de desarrollo μ Vision4 permite la depuración *on-chip* a través de adaptadores JTAG, para ello se debe activar la casilla “Use <Hardware de depuración>” en el apartado *Debug* del menú *Options for Target 1* del menú desplegable *Project*, tal y como muestra figura 2. El tipo de hardware que se debe seleccionar en nuestro caso es J-Link.

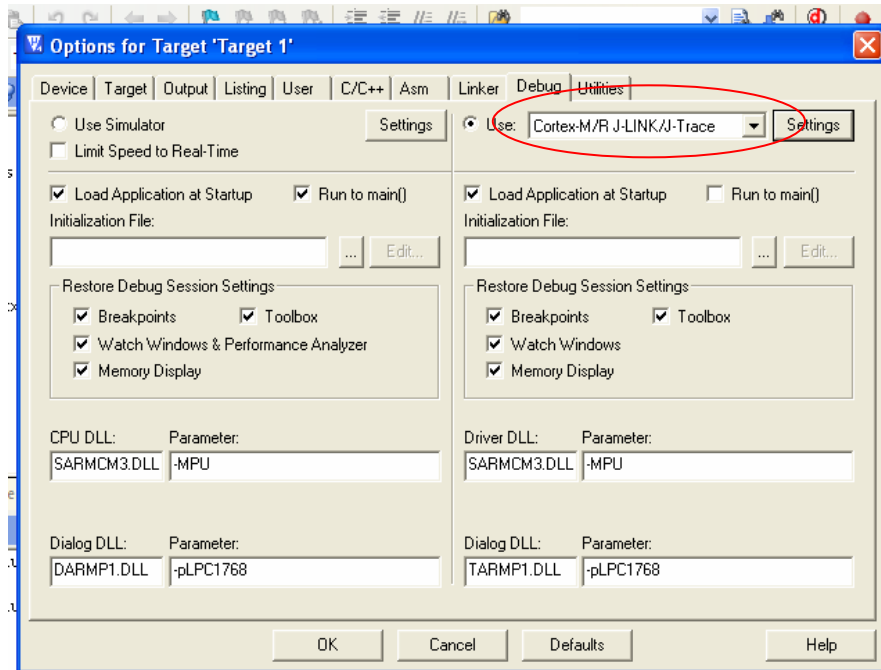


Figura 2. Menú en el que se puede seleccionar el tipo de depuración que se va a realizar:
Simulator o On-Chip

Aparte de las demás opciones que proporciona el cuadro de diálogo mostrado en la figura 1, es importante que no se modifiquen las librerías del simulador (CPU DLL y Dialog DLL) y del controlador del Hardware de depuración (Drive DLL y Dialog DLL).

La depuración *on-chip* permite el volcado de programas en la memoria Flash, para ello, se debe seleccionar el hardware para depuración que se va a emplear en el apartado **Flash --> Configure Flash Tools...-->Utilities**, según se muestra en la Figura 3.

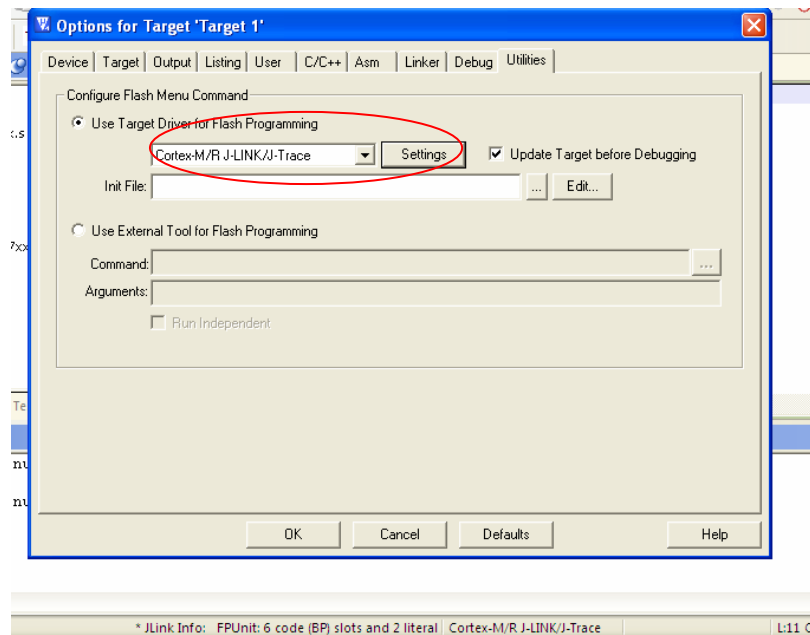


Figura 3. Menú que permite seleccionar el hardware con el que vamos a realizar el volcado en Flash de los programas a depurar.

Una vez seleccionado el Hardware de depuración, pulsamos *Settings* y en el siguiente cuadro de diálogo seleccionamos las características de la Flash de la tarjeta *Blueboard-H*, según se muestra en la Figura 4.

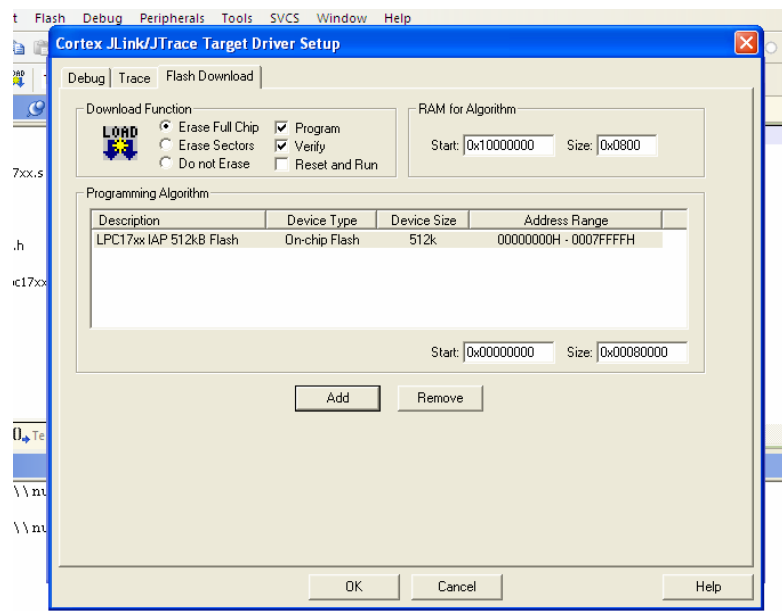


Figura 4. Especificaciones sobre la memoria flash de la tarjeta de desarrollo Blueboard LPC1768-H.

A esta misma ventana también se tiene acceso en la opción “settings” del cuadro de diálogo mostrado en la figura 2. En la pestaña “Debug” se debe seleccionar “interface USB”, mientras que el propio keil actualiza los datos referidos al J-Link, como son el número de serie (SN) y la versión de Hardware (HW). Si deseamos hacer el seguimiento en tiempo real de una variable debemos seleccionar port SW (serial wire) y pulsar sobre “Auto Clk” para que tome la frecuencia de reloj del dispositivo. Las opciones que deben estar seleccionadas se muestran en la figura 5.

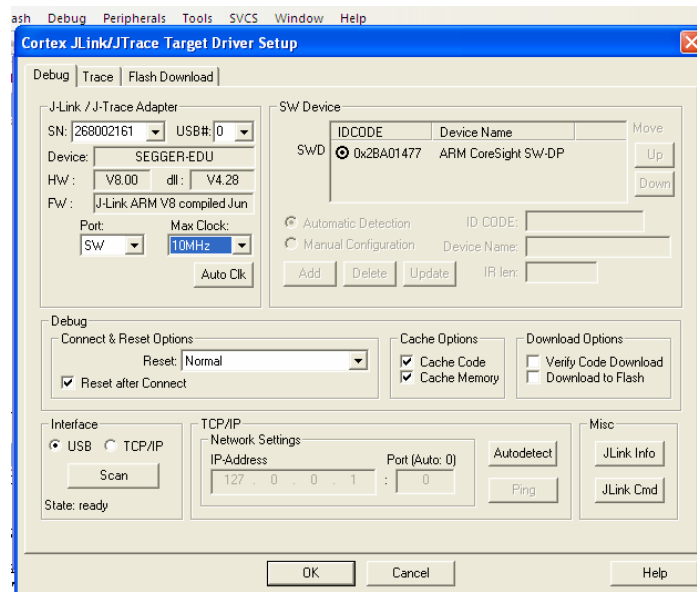


Figura 5. Especificaciones de depuración del J-link.

La pestaña Trace permite seleccionar las opciones de configuración de la toma de traza por el J-Link. Debemos tener activada las opciones que se muestran en la Figura 6. Entre ellas, las más importantes son:

- *PC sampling*, define la frecuencia de muestreo del ordenador sobre la tarjeta. El intervalo de muestreo debe ser lo suficientemente elevado como para permitir un refresco adecuado de los datos. La casilla “on data R/W Simple” debe estar activada para permitir que el PC realice un muestreo de las lecturas y escrituras de memoria.
- *Timestamps* permite reescalar los datos que se muestran en la pantalla del analizador lógico.

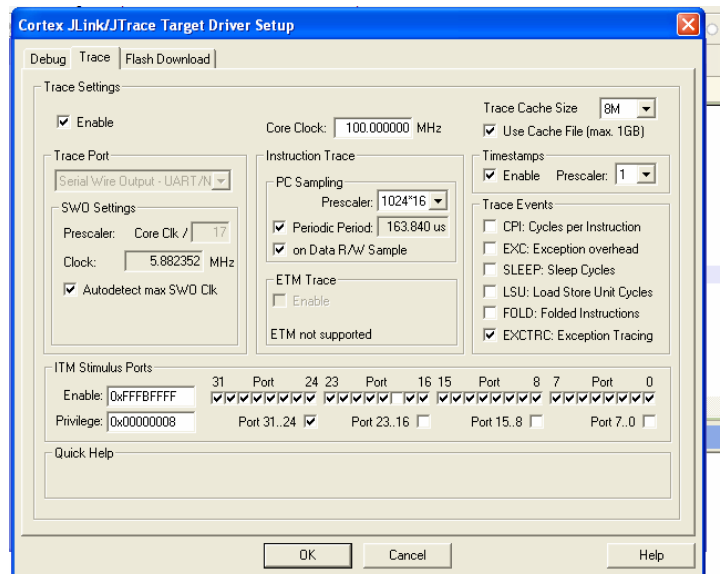


Figura 6. Opciones requeridas para el muestreo adecuado de variables en depuración.

Una vez configuradas las opciones anteriores, la depuración *on-chip* se realiza siguiendo los mismos pasos que en la opción de Simulación, iniciándola con la opción Start/Stop Debug Session

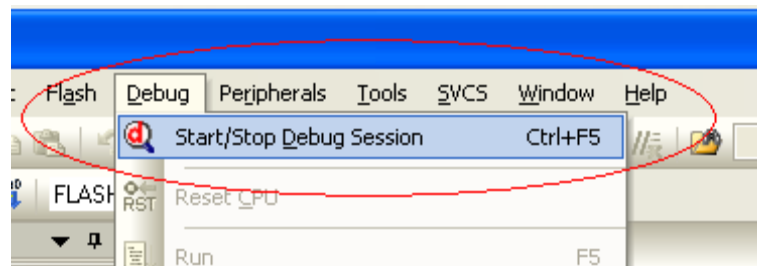


Figura 5. Opción de inicio de la depuración empleando JLINK.

El entorno μ Vision también permite la compilación de un mismo proyecto para ser depurado en simulación o cualquier otro medio, como el JLINK. Para ello se pueden generar distintos proyectos de manera paralela y que se modifican de manera simultánea. Para usar esta opción debemos crear un nuevo proyecto utilizando el gestor de componentes del proyecto. Tal y como aparece en la figuras 6a y 6b.

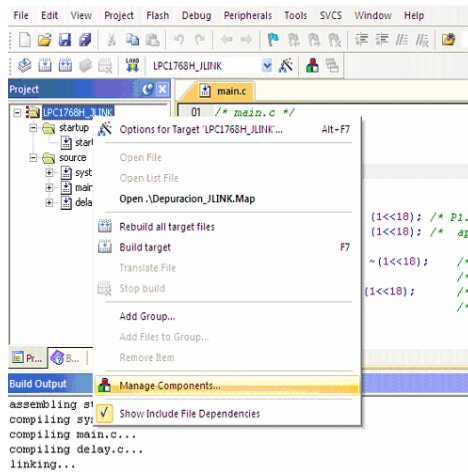


Figura 6a.

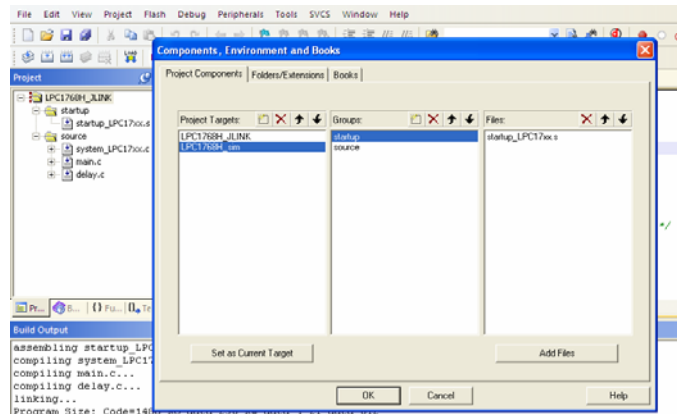


Figura 6b.

Una vez generadas las dos copias del proyecto, es posible cambiar de una copia a la otra usando el menú desplegable situado en la barra de comandos, como se recoge en la figura 7.

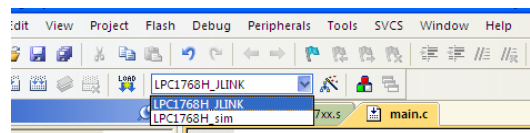


Figura 7. Menú desplegable que permite seleccionar el tipo de proyecto que se está depurando.

6 Programa ejemplo

El siguiente programa ejemplo produce el parpadeo del LED D7 de la tarjeta de desarrollo Blueboard LPC1768-H. Dicho LED se encuentra conectado al Pin P1.18, que es el que se cambia de estado a lo largo del programa. Para comprobar que se está depurando adecuadamente se debe ejecutar el programa paso a paso y comprobar que la ejecución de las líneas de encendido y apagado del LED producen el efecto deseado.

```
/* main.c */
#include "lpc17xx.h"
#include "delay.h"

int main (void)
{
    LPC_GPIO1->FIODIR |= (1<<18);      /* P1.18 definido como salida */
    LPC_GPIO1->FIOCLR |= (1<<18);      /* apago LED conectado a P1.18 */
    while(1)
    {
        LPC_GPIO1->FIOPIN &= ~(1<<18); /* enciendo LED conectado a P1.18 */
        delay (100000);                /* retardo */
        LPC_GPIO1->FIOPIN |= (1<<18);  /* apago LED conectado a P1.18 */
        delay (100000);                /* retardo */
    }
}
```

```
/* delay.c */

#include "delay.h"

void delay(int n)                /* funcion que produce el retardo*/
{
    int i;
    for(i=0;i<n;i++);
}
```

```
/* delay.h */

void delay(int n);
```