

	Módulos Digi + Windows CE	Comentario Técnico
		CoTC-001
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	Publicado: 00/00/0000
		Página 1 de 9

Revisión	Fecha	Comentario	Autor
0	19/10/2007		Ulises Bigliati
1	26/01/2010	Se agrega nota en página 5 sobre acortar tiempos de compilación.	Ulises Bigliati

## 1. Objetivo:

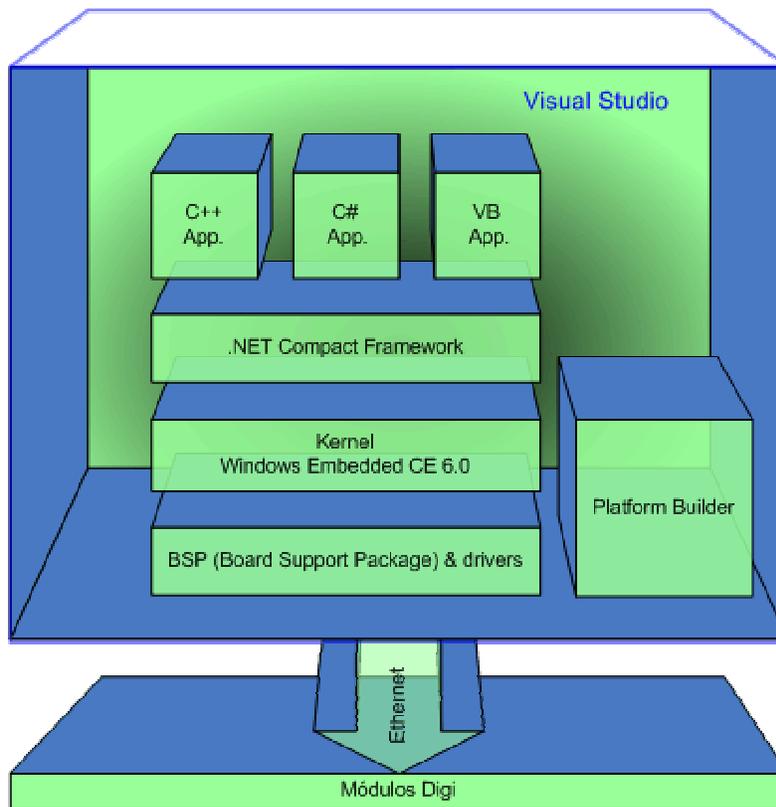
En esta ocasión describiremos los métodos de grabación y actualización de firmware en módulos Digi que ejecutan Windows CE.

Partiremos del supuesto caso de tener un proyecto de software listo para incorporar en forma estable a nuestro hardware.

Para esto se intentará proporcionar una aproximación al esquema de particiones que presenta la memoria flash y una visión del *boot loader* que utiliza Digi en sus módulos con el sistema operativo embebido Windows CE.

## 2. Introducción

El entorno de desarrollo de Visual Studio 2005 para Windows CE 6.0 embedded que correrá sobre módulos Digi basados en ARM9 se compone de una serie de partes que consideramos conveniente esquematizar. Una forma de verlo sería como se puede apreciar en la figura:



Podemos ver que el **Visual Studio 2005** integra los entornos de desarrollo para cualquiera de los tres lenguajes de programación que pueden apreciarse en la figura: **C++**, **C#** o **Visual Basic**.

Que estos lenguajes se apoyan en el **.NET Compact Framework** que proporciona a las aplicaciones una capa de abstracción que ofrece la suficiente portabilidad para que la plataforma hardware sea casi transparente al desarrollador. Tanto es así, que podremos correr nuestras aplicaciones indistintamente en la computadora de desarrollo o en el **módulo Digi**, ya que el Compact Framework, estará ejecutándose en ambas plataformas.

El **Board Support Package (BSP)** de Digi proporciona una capa de abstracción proporcionando drivers de dispositivos específicos para el hardware, permitiéndole al **Kernel de Windows CE 6.0**. el control del todo periférico presente en el módulo, lo cual se hace extensivo al usuario mediante gran variedad de librerías y API's.

	Módulos Digi + Windows CE	Comentario Técnico
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	CoTC-001
		Publicado: 00/00/0000
		Página 2 de 9

Finalmente a modo de plug-in, el **Platform Builder** se integra al Visual Studio en conjunto con el BSP, para permitir la configuración personalizada del kernel y la compilación de la imagen del sistema operativo Windows CE mas el sistema de archivos. Con esta utilidad podremos incluir nuestras propias aplicaciones, archivos y/o drivers en la imagen del kernel para luego grabar la memoria flash del módulo con ellas.

### 3 . Creación de una imagen del kernel de Windows CE con Platform Builder:

Esta operación generará una imagen binaria de Windows CE conteniendo en principio los componentes y drivers de dispositivos que hayamos definido.

A continuación, podremos agregar los archivos correspondientes a nuestra aplicación, como así también los accesos directos correspondientes que necesitemos.

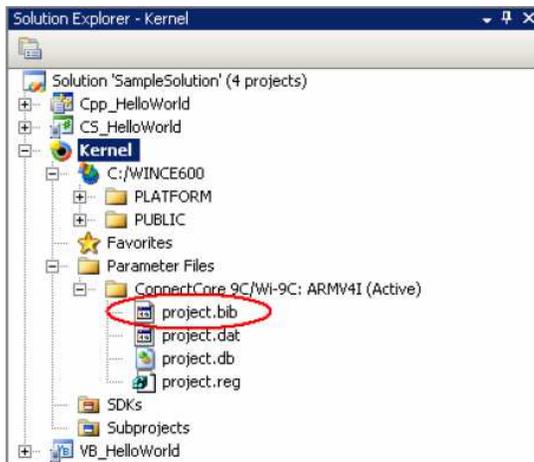
Para comenzar el proyecto de una nueva imagen del kernel debemos iniciar el Visual Studio 2005 y a continuación clicar en: File > New > Project, se abrirá la siguiente ventana:



No nos detendremos en los detalles consecutivos, ya que están bien explicados en el manual de usuario del kit de desarrollo. La intención de este trabajo es remarcar algunos puntos que quizá no son muy claros.

#### 3.1. Archivos de configuración del sistema

Al finalizar el Wizard del proyecto, podremos visualizar en la ventana de exploración de soluciones un listado como el siguiente:



Aquí nos importa remarcar lo siguiente: todo proyecto de Platform Builder tendrá una estructura similar a la que podemos ver a la izquierda. Lo que nos importa es la zona marcada en rojo, esto es el archivo **project.bib** y también el **project.dat**. En conjunto estos archivos son los que nos proporcionan la funcionalidad de incorporar a nuestra propia aplicación a la imagen del kernel de WinCE que correrá finalmente en el módulo. Físicamente estos archivos se encuentran en: A continuación se verá la estructura de estos archivos de texto plano que es muy simple, pero que es fundamental manipular, sobre todo cuando nuestro proyecto pase a producción, quizá no tanto en etapas de desarrollo.

	Módulos Digi + Windows CE	Comentario Técnico
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	CoTC-001
		Publicado: 00/00/0000
		Página 3 de 9

### 3.2. El archivo project.bib

La estructura del archivo project.bib puede verse a continuación, allí podemos ver que es necesario tabular tres datos por cada objeto que se quiera agregar a la imagen del kernel, a saber:

El nombre del archivo, la ubicación del mismo en la máquina de desarrollo y por último un código que indica el tipo de memoria en donde será grabado, esto por defecto será NK.

FILES	Name	Path	Memory Type
	vb_hello.exe	PathToTheFile\VB_HelloWorld.exe	NK
	cs_hello.exe	PathToTheFile\CS_HelloWorld.exe	NK
	cpp_hello.exe	PathToTheFile\Cpp_HelloWorld.exe	NK

**Todos los archivos cargados en la imagen del kernel que estamos construyendo serán almacenados en el directorio de Windows, es decir su ruta será “\Windows\”**

#### 3.2.1. Accesos directos:

Para crear un acceso directo, debemos crear un archivo de texto plano, y escribir en su interior la ruta de acceso del archivo de destino precedida por un número que indica su longitud mas el signo '#':

```
21#\windows\cs_hello.exe
```

Luego el archivo de texto debe ser guardado con extensión “.lnk”

```
Directory("\Windows\Startup") :-File("cs_hello.lnk", "\Windows\cs_hello.lnk")
```

Luego el acceso directo puede ser agregado al archivo project.bib como cualquier otro archivo:

cs_hello.exe	PathToTheFile\CS_HelloWorld.exe	NK
cs_hello.lnk	PathToTheFile\CS_HelloWorld.lnk	NK

### 3.3. El archivo project.dat

La imagen que estamos construyendo será grabada en la memoria flash del módulo, pero cuando este bootea, se crea una imagen de Windows en tiempo de ejecución que corren en RAM, esto es de forma similar a una PC de escritorio.

Por eso es que mediante el archivo project.dat es posible especificar la estructura de archivos de esta imagen de Windows de tiempo de ejecución.

En este archivo, pueden especificarse entonces diferentes locaciones para nuestros archivos que normalmente están grabados en el directorio \Windows.

Así, la siguiente línea escrita dentro del archivo project.dat:

```
Directory("\Windows\Startup") :-File("cs_hello.lnk", "\Windows\cs_hello.lnk")
```

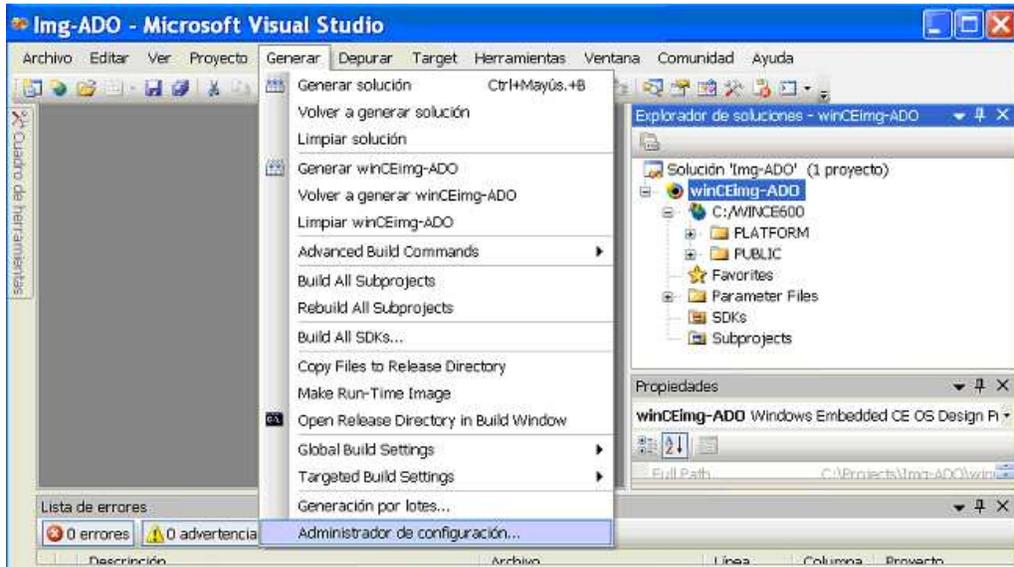
	Módulos Digi + Windows CE	Comentario Técnico
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	CoTC-001
		Publicado: 00/00/0000
		Página 4 de 9

Indica que, el acceso directo “cs\_hello.lnk”, que está ubicado dentro del directorio “\Windows” será ubicado en tiempo de ejecución dentro del directorio “\Windows\Startup”. Lo cual además agrega el efecto de que el sistema operativo intentará ejecutar inmediatamente la aplicación referenciada por el acceso directo al iniciarse.

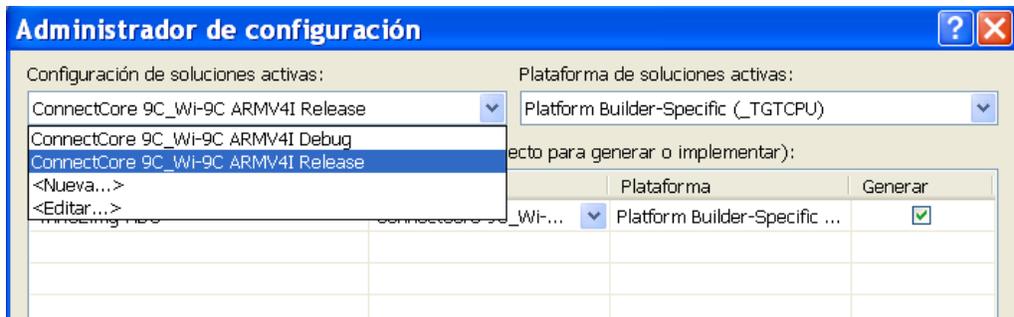
#### 4. Compilando una nueva imagen para Windows CE.

Una vez que terminamos de agregar los archivos de nuestra aplicación y de configurar la ubicación de los accesos directos, si esto hiciera falta. Debemos compilar la imagen.

En el menú Generar > Administrador de Configuración se debe seleccionar el tipo de imagen a generar

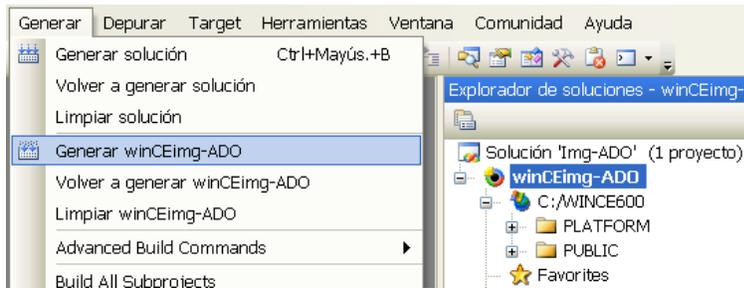


Esto es: Debug o Release, normalmente será Release, salvo que realmente fuera necesario debuggear el kernel.

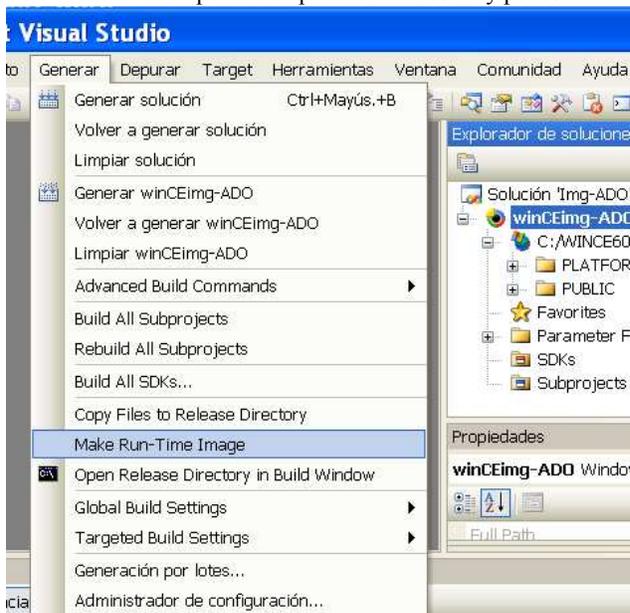


A continuación podemos realizar la operación de General o Compilar el proyecto y de ir a prepararnos un café, que a estas alturas se hará mas que necesario. La primera vez que se copia el proyecto demora entre 25 y 30 minutos, y a veces todavía mas, dependiendo de los componentes seleccionados y por supuesto del poder de la computadora.

	Módulos Digi + Windows CE	Comentario Técnico
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	CoTC-001
		Publicado: 00/00/0000
		Página 5 de 9



Pero, afortunadamente, una vez que ya ha sido realizada la primer compilación, en los sucesivos intentos, por ejemplo, si fuera necesario agregar un nuevo archivo al project.bib o un nuevo componente al catalogo de Windows. La recompilación se hará con el comando “Make Run-Time Image”, como puede verse en la imagen: Esto reduce el tiempo de compilación a unos muy pocos minutos.



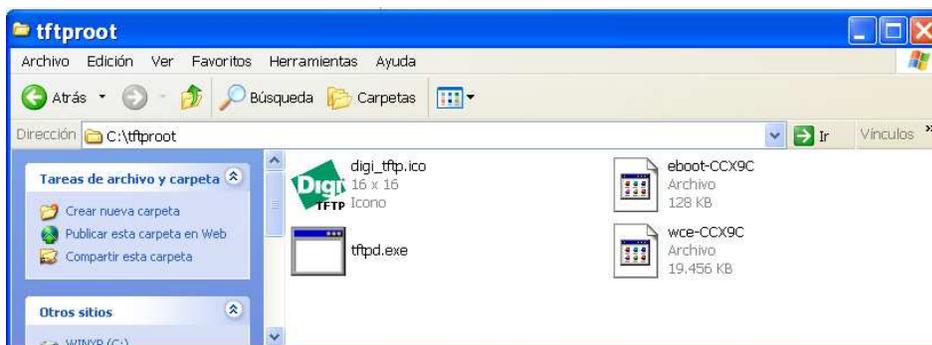
#### Nota importante:

Durante el desarrollo, cuando se realicen pequeños cambios sobre el sistema de archivos de WCE, como agregar nuevos archivos, o modificar el registro, en otras palabras, cuando se modifiquen los archivos: **project.bib**, **project.reg**. Será necesario, a fines de no sufrir una compilación completa del kernel, copiar esos archivos que hemos cambiado al “%FLATRELEASEDIR%”, es decir pasar los cambios desde los archivos originales del proyecto: **[PROJECTPATH]\Wince600\CCX9C\_ARMV4I\OAK\files** hacia el directorio de compilación: **[PROJECTPATH]\RelDir\ConnectCore\_9C\_Wi-9C\_ARMV4I\_Release** dentro de este, se encuentra el directorio **\postproc**. que es donde debemos sobrescribir los archivos (citados arriba en negrita) que hayamos modificado. A continuación podemos ejecutar el comando **“Make Run-Time Image”** para generar la nueva imagen conteniendo los cambios. Este procedimiento es tedioso, pero nos ahorra mucho, mucho tiempo...

Una vez concluido el proceso de compilación, la nueva imagen del kernel (que normalmente pesa unos 20MB) estará disponible para ser enviada al módulo. La ubicación por defecto de la imágenes compiladas producto de los proyectos del Platform Builder es “C:\tftproof”. Dicha ruta es la ubicación por defecto del servidor TFTP que es instalado durante la instalación del BSP de Digi.

Mas adelante daremos mas detalles del servicio TFTP, que justamente es una de los medios utilizados para grabar nuevas imágenes en el módulo. El nombre del archivo de imagen es, como puede verse **wce-CCX9C**.

Esto es **wce-Nombre\_de\_la\_plataforma**.



	Módulos Digi + Windows CE	Comentario Técnico
		CoTC-001
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	Publicado: 00/00/0000
		Página 6 de 9

## 5. Transferencia del sistema al módulo Digi

Tenemos la imagen lista para ser enviada a nuestro destino. Bien, hay varias formas de hacer esto:

- 1) Directamente desde el Platform Builder, pero a modo de debug, los procedimientos están detallados en el manual de usuario de WinCE del kit de desarrollo, no los comentaremos en este trabajo.
- 2) Mediante el servicio TFTP.
- 3) Mediante la interfaz de línea de comando del módulo que puede accederse vía Telnet.

A continuación se describen algunos conceptos que son necesarios para realizar las actualizaciones del sistema.

### 5.1. El boot loader

Genéricamente un boot loader es un pequeño componente de software encargado de cargar un sistema operativo -que normalmente se encuentra en un medio de almacenamiento masivo- en un área de memoria desde donde el microprocesador pueda ejecutarlo.

Entre las funciones de un boot loader para sistemas embebidos se encuentran:

- Inicializar el hardware, especialmente los controladores de memoria
- Proveer los parámetros de booteo para SO
- Cargar e iniciar el SO
- Permitir la lectura y escritura de sectores arbitrarios de memoria
- Proveer utilidades de uploading de nuevas imágenes binarias
- Copia de imágenes de memoria RAM a Flash.

El boot loader que está presente en el módulo Digi es el U-Boot, entre la documentación del Kit, se encuentra el U-Boot Reference Manual, allí se puede encontrar mayor información al respecto. Aquí nos enfocaremos sobre las funcionalidades básicas que nos permitirán iniciar y probar y actualizar nuestro sistema embebido.

Por defecto, U-Boot utiliza como consola de diálogos, el puerto serial A del módulo (38400,8,1,N) que accesible con niveles RS232 mediante un conector DB9 en la placa de desarrollo. Debemos conectar un cable serial estándar (pin a pin) desde el citado puerto al puerto serie de una computadora, e iniciar allí una terminal que nos permita acceder a la línea de comandos de U-Boot. Para esto alimentamos el módulo y podremos ver en la pantalla de nuestra terminal información relativa la actividad de U-Boot. Será necesario presionar un tecla cuando el diálogo de booteo así lo indique para detener el booteo automático y poder operar con los comandos del boot-loader.

#### 5.1.1. Comandos básicos

A continuación se muestran los comandos que básicamente utilizaremos, estos podemos agruparlos en dos clases:

##### 5.1.1.1 Los relativos al manejo de las variables de entorno de U-Boot:

```
printenv [variable]
    Muestra el valor de la variable de entorno solicitada, o de todas si esta no se
    especifica.
setenv <variable><value>
    Modifica el valor de la variable de entorno indicada, o la crea estableciendo el valor
    suministrado si es que todavía no existía.
saveenv
    Graba en forma permanente las variables de entorno.
```

	Módulos Digi + Windows CE	Comentario Técnico
		CoTC-001
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	Publicado: 00/00/0000
		Página 7 de 9

### 5.1.1.2. Los relativos a las acciones de booteo y actualización de firmware.

`dboot <os> <type> [partition]:`

Donde <os> es el sistema operativo utilizado en este caso será “wce” y <type> el modo de booteo, que será “tftp” o “usb”

`update <partition> <source> <file>`

Donde <partition> puede tomar los siguientes valores: “uboot”, “linux”, “rootfs”, “userfs”, “userfs”, “eboot”, “wce”.

Los valores para <source> serán “tftp” o “usb”

En lugar de <file> se puede especificar el nombre del archivo de imagen, o obviarlo si este está incluido entre las variables de entorno.

### 5.1.2. Las variables de entorno

Las variables que se utilizan en el proceso de booteo y que entre otras utilidades permiten la utilización de las funciones de actualización de firmware son las siguientes:

`bootcmd`

Define un string de comando que es ejecutado automáticamente cuando la cuenta regresiva inicial no se interrumpe. Se ejecuta solo cuando la variable **bootdelay** también está definida.

`Bootdelay`

Segundos de espera antes de iniciar el proceso de booteo automático de bootcmd.

`dhcp`

Si se setea en **on**, se intentará obtener una dirección IP por DHCP para la interfaz Ethernet.

`dhcp_wlan`

Idem para la interfaz WLAN.

`ipaddr`

Dirección IP estática para la interfaz Ethernet del módulo.

`ipaddr_wlan`

Idem para la interfaz WLAN.

`netmask`

Máscara de subred para Ethernet.

`netmask_wlan`

Máscara de subred para WLAN.

`serverip`

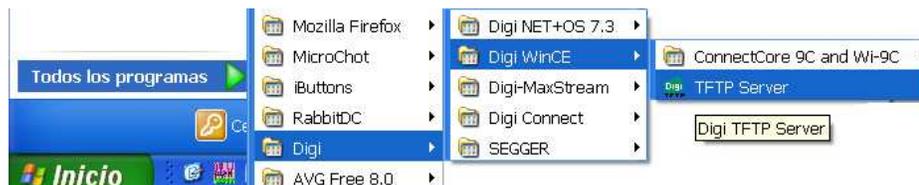
Dirección IP de la computadora de desarrollo (Necesario para la conexión de U-Boot por

tftp).

## 5.2 El servidor TFTP

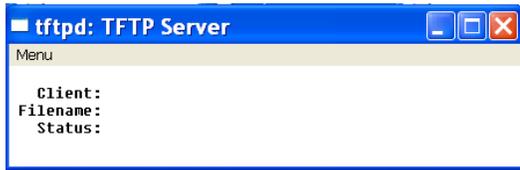
Para realizar una actualización de la imagen del sistema vía Ethernet o WiFi, desde la interfaz de comandos de U-Boot debemos tener funcionando el servidor TFTP para que el boot loader tome el archivo binario correspondiente a la imagen que queremos grabar o también desde la cual queremos bootear para testear su funcionamiento sin grabar el módulo.

Para ejecutarlo:



	Módulos Digi + Windows CE	Comentario Técnico
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	CoTC-001
		Publicado: 00/00/0000
		Página 8 de 9

Se abrirá una ventana como la siguiente que mostrará información relativa a la transferencia del archivo.



Recuerden que el archivo de imagen ya fue escrito en la ruta "C:\tftpboot" mediante la compilación hecha desde el Platform Builder.

### 5.3. Actualizando el sistema con U-Boot

Debemos asegurarnos de que la variable de entorno `serverip` esté configurada con la IP del host, normalmente la máquina de desarrollo.

Podemos chequear esto tipeando desde la consola "`# printenv serverip`" y si no está seteada o debemos modificarla, lo haremos con "`# setenv serverip <hostip>`" y a continuación "`saveenv`".

En estas condiciones tenemos la posibilidad de realizar las siguientes operaciones:

#### 5.3.1. Actualizar la imagen vía TFTP o USB

Para esto, tipeamos en la consola:

```
# update wce tftp, o bien:
# update wce usb
```

#### 5.3.2. Bootear con la nueva imagen en RAM

A fin de testear la imagen directamente en RAM sin grabar el módulo, leyendo la imagen desde USB o TFTP podemos utilizar el comando `dboot`.

Entonces tipeamos:

```
# dboot wce tftp, o bien:
# dboot wce usb
```

Los módulos ya vienen grabados con el nombre del archivo de imagen por defecto (`wce-CCX9C`) en una variable de entorno, por lo tanto no es necesario suministrar este dato al comando correspondiente.

En el caso de utilizar como medio la interfaz USB obviamente debimos previamente colocar en el módulo un pen-drive con el archivo de imagen.

### 5.4. Actualización mediante línea de comandos de Windows CE.

La otra alternativa para la actualización de la memoria flash es la conexión remota vía Telnet.

El BSP de Digi, provee la utilidad **update\_flash** accesible mediante la línea de comandos de Windows CE, que nos permite leer o escribir la partición de la memoria flash especificada, utilizando como destino o fuente el archivo indicado.

Cuando se trata de actualizaciones de flash, dicho archivo, debe ser transmitido previamente al módulo, es decir, tiene que estar presente en el file system de Windows CE, en la RAM del módulo.

Podemos hacer esto fácilmente mediante un cliente FTP, ya que en el módulo, por defecto se está ejecutando este servicio.

El menú del comando en cuestión, tras la conexión vía Telnet, luce de un forma similar a la que puede verse en la siguiente imagen:

	Módulos Digi + Windows CE	Comentario Técnico
	Métodos de grabación del firmware en módulos Digi con Windows CE 6.0	CoTC-001
		Publicado: 00/00/0000
		Página 9 de 9

```

C:\> telnet 192.168.42.30
Welcome to the Windows CE Telnet Service on <target>

Pocket CMD v 6.00
\> update_flash -h
Application to read, write and list
the flash partitions. Revision 1.0
Copyright(c) 2007 Digi International Inc.

Usage: update flash <options>

Where options are:
-r           Reads the specified partition
-w           Writes the specified partition
-p <partition> Selects partition (<partition> must be a number)
-f <filename> File name to write/read the partition data
-l           Lists the partition table
-h           Shows this help

Examples:
update flash -r -p 3 -f image1.bin
update flash -w -p 4 -f image3.bin
\>

```

Una vista típica de la tabla de particiones del módulo es la siguiente:

Partition number	Name	Flash start address	Flash end address	Length	Description
0	U-Boot	0x00000000	0x000c0000	768 KiB	Stores the U-Boot boot loader image
1	NVRAM	0x000c0000	0x00140000	512 KiB	Stores permanent configuration parameters like the MAC address of the network interfaces, the serial number of the module, environment variables of U-Boot, and so on
2	FPGA	0x00140000	0x00240000	1 MiB	Stores the FPGA firmware
3	EBoot	0x00240000	0x00340000	1 MiB	Stores the EBOOT program for connecting with Platform Builder
4	Registry	0x00340000	0x00440000	1 MiB	Stores the Windows CE Registry
5	Kernel	0x00440000	0x01840000	20 MiB	Stores the Windows CE kernel image

## 6. Conclusión

Durante etapas de desarrollo, cuando escribimos aplicaciones para Windows CE embedded, sobre módulos Digi, lo hacemos en el IDE de Visual Studio, y desde allí, podemos debugear o ejecutar directamente nuestra aplicación en el módulo vía Ethernet o WiFi, de esta manera los archivos del programa en desarrollo se alojarán en el file system del módulo, en su memoria RAM. También es posible portar nuestros archivos en un pen-drive y colocarlo en el módulo, y este será visto como un disco rígido.

Pero en etapas de producción, cuando necesitamos que el sistema desarrollado permanezca estable en el módulo, o incluso podríamos requerir la auto-ejecución del programa con el sistema operativo, y/o cualquier otra función o servicio, lo que debemos hacer es generar una nueva imagen del sistema completo, mediante el platform builder, que incluirá también a nuestra aplicación, junto con las modificaciones estructurales que pudieran hacer falta. Finalmente, la imagen debe ser transferida al módulo de alguna de las maneras descriptas.