Práctica II

U-boot

Implementación de Sistemas Operativos II

Maestría en Sistemas Embebidos

Año 2021

Autor

Mg. Ing. Gonzalo Sanchez

Tabla de contenido

Registro de cambios

Revisión	Cambios realizados	Fecha
1.0	Creación del documento	24/07/2021
1.1	Cambio de título de carátula Escrita sección “U-boot como bootloader”	27/07/2021
1.2	Agregada sección Configuración de la tarjeta MMC/SD	06/09/2021
1.3	Agregada sección Configuración de la comunicación serie con la placa. Agregada sección Iniciar Uboot desde la memoria SD	07/09/2021

U-boot como bootloader

Dado que el cargador de arranque es la primera pieza de Software ejecutada por una plataforma Hardware, el procedimiento de instalación del cargador de arranque es específico para dicha plataforma.

En general existen dos casos:

- El procesador no ofrece ninguna facilidad para la instalación del cargador de arranque, en cuyo caso se debe utilizar JTAG para inicializar el almacenamiento flash y escribir código del cargador de arranque en la misma. Es necesario un conocimiento detallado del Hardware para poder realizar estas operaciones.
- El procesador ofrece un monitor, implementado en ROM, a través del cual acceder a la memoria es más simple.

La placa Beaglebone Black, utiliza el SoC AM3359 y entra en la segunda categoría. Por defecto, el monitor integrado en la ROM del AM3359 lee la tarjeta MMC1 (la memoria integrada eMMC) seguido de la MMC0 (MicroSD), la UART0 y finalmente el USB0. Si el pulsador **S2** es presionado durante el encendido de la placa, la ROM iniciará desde la interfaz SPI0, luego la MMC0 (MicroSD), la USB0 y finalmente la UART0.

Obtención de fuentes, configuración y compilación

En esta práctica dentro del directorio **ISO_II** se debe crear un directorio llamado **bootloader**. Luego de crearlo, se debe ingresar al mismo.

```
$ mkdir -p -v $HOME/ISO_II/bootloader

$ cd $HOME/ISO_II/bootloader
```

Una vez dentro del directorio de trabajo, es necesario clonar las fuentes de **Uboot**.

```
$ git clone git://git.denx.de/u-boot.git
```

Habiendo clonado el repositorio, se debe ingresar al directorio y luego hacer un checkout a la última versión estable. Esto se hace de la misma forma que la práctica anterior, ya

que sigue el mismo esquema de versionados que CrossTool-NG. La última versión estable es la 2021.07, por lo que para poder compilar la misma se debe efectuar un checkout al tag **v2021.07**.

```
$ cd u-boot

$ git checkout v2021.07
```

Es recomendable en cualquier caso, ya sea esta herramienta u otra, leer el archivo README y por sobre todo cualquier instrucción que tenga que ver con la compilación para el hardware utilizado. Esto usualmente supone un ahorro de tiempo en el caso de que exista alguna particularidad con el hardware en el que se quiere utilizar cualquier herramienta.

Para poder compilar los archivos fuentes, se debe primero definir dos variables de entorno, las cuales son **CROSS_COMPILE** y **ARCH**. Es muy importante que el lector entienda que *estas variables de entorno solo son válidas para este proyecto, dado que están definidas en su makefile*. Si bien, algunos proyectos la comparten, como ser el kernel de linux, pueden no existir en otros o bien tener nombres o significados diferentes.

Estas variables de entorno se definen antes de invocar a make, y lo recomendable es definirlas dentro del bash que se está utilizando mediante el comando *export*.

```
$ export CROSS_COMPILE=arm-linux-

$ export ARCH=arm
```

Es necesario aclarar que el prefijo del compilador debe contener el guión medio final dado que lo único que se le adosa es el *gcc* al final para invocar al compilador. Asimismo, el PATH en donde está almacenado este cross compilador debe ser conocido para el sistema, caso contrario el comando make finalizará con un error por no encontrar el compilador.

Antes de compilar el proyecto, debe ser configurado para el SoC en cuestión. De la misma manera que crosstools-ng contiene *samples*, las cuales son configuraciones

mínimas para cada hardware, u-boot posee configuraciones mínimas por defecto para algunos hardwares. En el caso de la BeagleBone Black, el archivo es **am335x_evm_defconfig**.

```
$ make am335x_evm_defconfig
```

Para poder navegar en esta configuración y ver detalles con la posibilidad de modificarla, es posible ejecutar *make menuconfig*. Se invita al lector a investigar las configuraciones que se hacen sobre la compilación y modificar alguna. Un ejemplo sencillo es el tiempo de timeout antes de que se ejecute el boot automatizado.

Finalmente, para compilar el proyecto se ejecuta simplemente *make*. En el directorio de trabajo se generan dos archivos que son necesarios, un archivo **MLO** y **u-boot.img.** Al archivo *MLO* se lo conoce como **First Stage Bootloader**. Al archivo *u-boot.img* se lo conoce como **Second Stage Bootloader**.

Puesta a punto de hardware

Configuración de la tarjeta MMC/SD

Para poder correr U-Boot recientemente compilado, se debe preparar una memoria microSD de manera especial para poder utilizarla como unidad de booteo.

NOTA: *Algunas memorias con capacidad menor a 2GB pueden fallar en el proceso de booteo.*

El primer paso será borrar todas las particiones existentes de la memoria microSD haciendo uso del comando **fdisk**. En el comando siguiente debe ser reemplazado el dispositivo de salida sdXY por el que corresponda, donde X representa una letra otorgada por el sistema e Y el número de la partición correspondiente. Puede determinar este dispositivo mediante el comando **dmesg | tail** o ingresando al directorio **/dev**, una vez conectada la memoria a la estación de trabajo. También puede utilizarse una herramienta gráfica como **gparted**. Antes de borrar cualquier partición deberá desmontarlas.

```
$ umount /dev/sdXY

$ sudo fdisk /dev/sdXY

   Command (m for help): d
```

La opción **d** dentro del comando fdisk borra la partición seleccionada.

Luego de haber borrado todas las particiones, se procede a crear una nueva con un formato específico para ser utilizada como arranque en la SBC BeagleBone Black.

```
Command (m for help): n

Partition type:

     p   primary (0 primary, 0 extended, 4 free)

     e   extended

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-15130623, default 2048): 2048

Last sector, +sectors or +size{K,M,G} (2048-15130623, default
15130623): +64M


Crea una nueva partición 1 de tipo 'Linux' y de tamaño 64 MiB.
```

Dentro del comando fdisk la opción **n** crea una nueva partición, la cual debe ser marcada como primaria, y se le asigna el número de partición 1. Luego el tamaño se especifica como +64M, no es necesario un tamaño mayor.

Hecho esto, se debe indicar que esta partición es del tipo de arranque, para lo cual se utiliza la opción **a** dentro del comando fdisk.

```
Orden (m para obtener ayuda): a

Se ha seleccionado la partición 1

El indicador de iniciable de la partición 1 ahora está activado.
```

Para finalizar con este proceso, es necesario cambiar el tipo de partición a *WIN95 FAT32 (LBA)* para lo que se utiliza la opción **t** y luego el tipo se indica con **c**.

```
Command (m for help): t Selected partition 1 Hex code (type L to list codes): c

Se ha cambiado el tipo de la partición 'Linux' a 'W95 FAT32 (LBA)'.
```

La partición ahora tiene las propiedades necesarias para poder bootear. Para que la partición creada según las opciones seleccionadas se escriba efectivamente en la memoria SD es necesario ejecutar el comando **w**.

```
Command (m for help): w
```

Una vez hecho esto Se procede a formatear esta partición como tipo FAT y se marca como etiqueta BOOT mediante el comando **mkfs**.

```
$ sudo mkfs.vfat -n "BOOT" /dev/sdXY
```

Es necesario que el medio sea desmontado y se vuelva a montar para poder utilizarlo, lo que se puede lograr rápidamente extrayendo el medio y volviendo a conectarlo. Una vez vuelto a montar, es posible copiar los archivos **MLO** y **u-boot.bin** los que utiliza la SBC para poder iniciar.

Configuración de la comunicación serie con la placa

Para poder visualizar la salida de consola del bootloader en la workstation, se debe conectar la salida serie de la SBC, y el estándar de-facto es utilizar un adaptador USB-UART TTL.

Como puede verse en la figura 1, la forma de conectar el adaptador USB-UART TTL a la placa Beaglebone Black es por medio de tres pines expuestos. El cable negro en la figura corresponde a GND del adaptador y se conecta al pin 1 (GND) del conector J1. El cable TXD del adaptador al pin 4 del conector J1 y el cable RXD al pin 5 del mismo conector. Cuando se conecte el adaptador a la estación de desarrollo, deberá aparecer un nuevo puerto serie **/dev/ttyUSB0** el cual será utilizado para la comunicación.

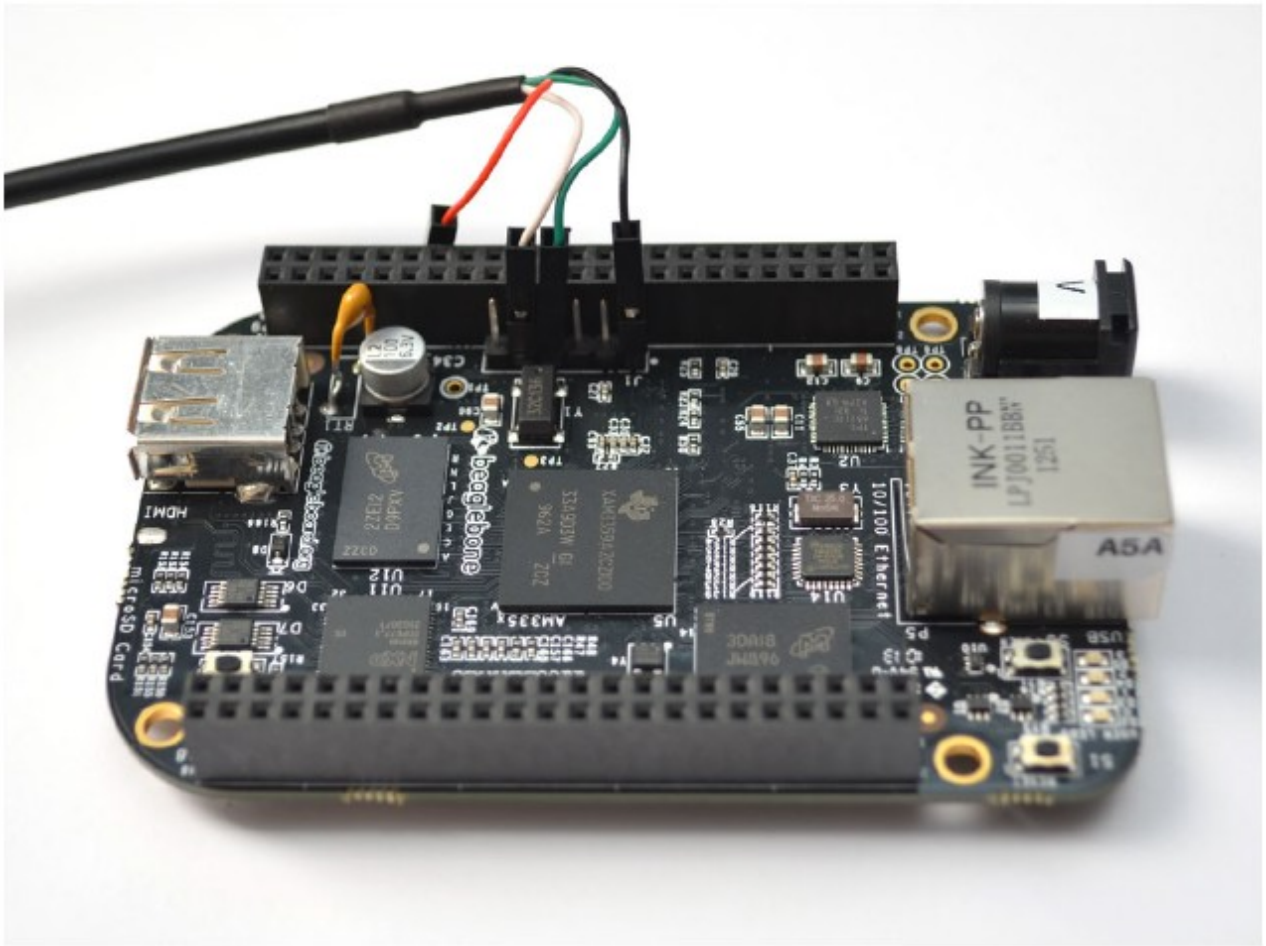

Figura 1: Conexión de adaptador USB-UART TTL a la SBC BeagleBone Black

Es posible visualizar si el dispositivo fue reconocido por la estación de desarrollo observando la salida del comando **dmesg | tail**.

Una vez reconocido el dispositivo y enumerado por la estación de trabajo, se puede utilizar cualquier programa destinado a la comunicación serie, como por ejemplo *puTTy*, *gtkterm*, *picocom*, entre otros.

NOTA: Es altamente probable que no sea posible abrir el puerto la primera vez que quiera utilizarse, y esto responde a la falta de permisos del usuario para efectuar esa acción. Es necesario verificar que el usuario actual esté incluido en el grupo *DIALOUT*. En el caso de no estarlo, es necesario ejecutar el siguiente comando y luego reiniciar la sesión del usuario actual.

```
$ sudo adduser $USER dialout
```

Iniciar Uboot desde la memoria SD

Antes de proceder a hacer uso de la imagen de u-boot recientemente creada, es necesario recordar el proceso de inicio y boot de la BeagleBone Black, el cual se muestra en la figura 2 mediante un diagrama de flujo.

El botón **BOOT** está en la SBC indicado como **S2**. Este botón switch está posicionado en el extremo donde se sitúan los conectores USB, microUSB y el alojamiento de la memoria microSD. La mencionada posición se muestra claramente en la figura 3.

Según el diagrama de flujo, la secuencia de booteo puede modificarse según el estado del botón S2 al momento de energizar la SBC, resultando en los siguientes casos:

Secuencia	Botón S2 sin presionar (boot normal)	Botón S2 presionado (boot alternativo)
#1	eMMC	SPI0
#2	UART0	MMC (sd card, partición 1 FAT)
#3	USB0	USB0
#4	--	UART0

Esta secuencia, muestra que para poder iniciar la imagen de u-boot cargada en la memoria microSD, se debe energizar la SBC con el botón S2 presionado. Lo usual es que la placa incluya una versión de u-boot precargada en la memoria eMMC, lo que genera confusiones en más de una oportunidad. Para determinar qué secuencia se ha utilizado (y por lo tanto desde que medio se hizo el boot) se observa la versión de u-boot que se imprime en pantalla cuando el mismo inicia. Lo usual es que la versión cargada en eMMC sea más antigua que la creada y cargada en la memoria SD.

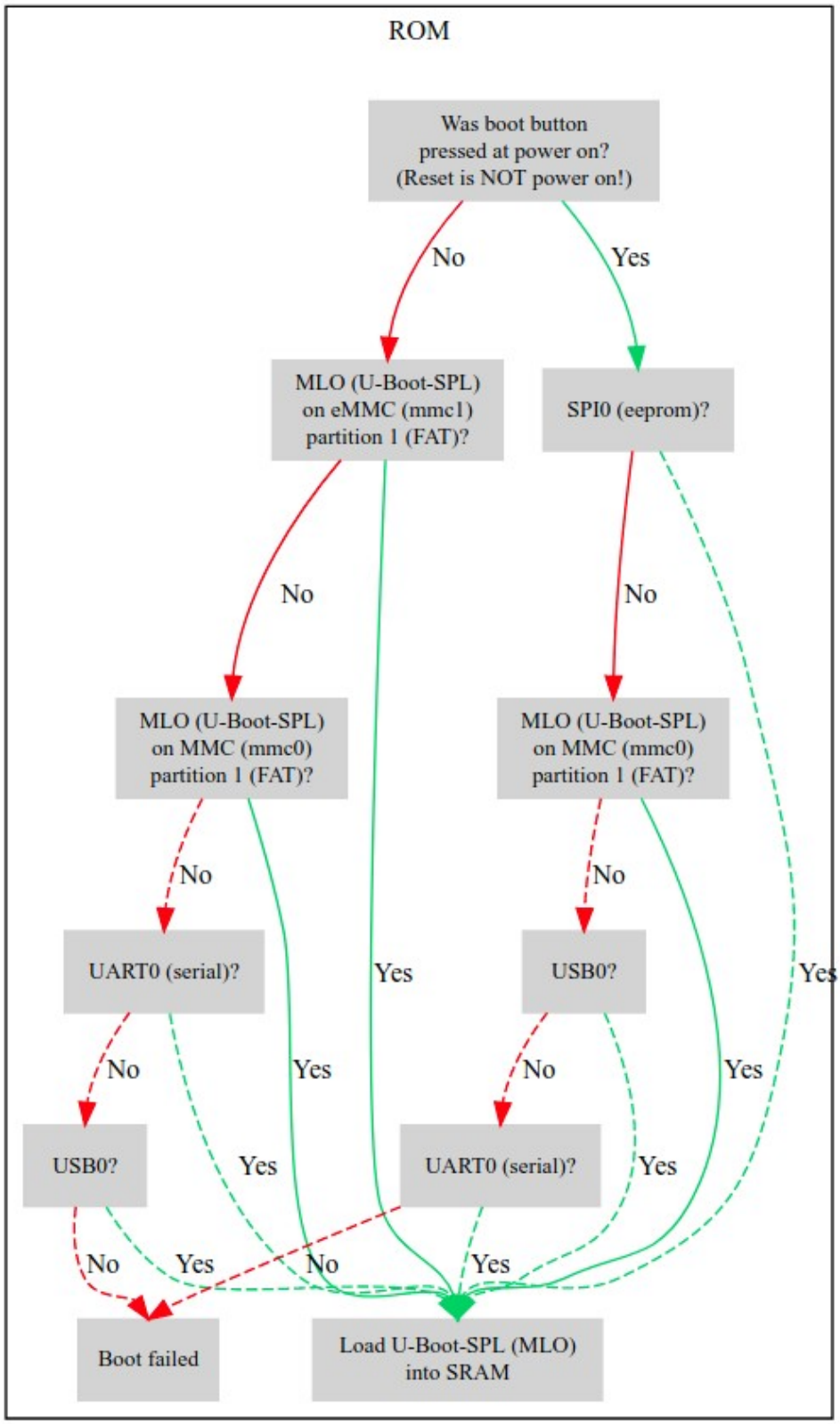

Figura 2: Diagrama de flujo correspondiente al booteo de BeagleBone Black

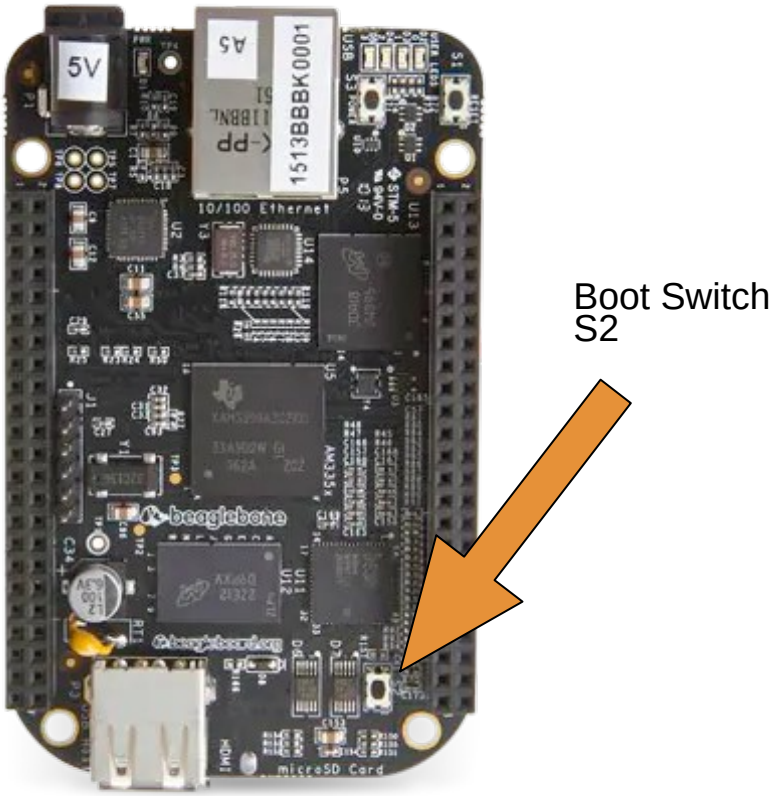

Figura 3: Posición del botón BOOT en BeagleBone Black