José Roberto Guerrero Zurita
Diplomado Linux en Sistemas Embebidos

Compilación Cruzada

La compilación cruzada consiste en compilar código en una plataforma distinta a la que se ejecutará.

Para la siguiente tarea es necesario:
+ Código de Kernel de Linux
+ Código de SPL y U-Boot
+ Compilador cruzado
+ Variables de entorno
+ Compilar SPL y U-Boot
+ Configuración .config
+ Compilar Kernel y Device Tree Blob de la Hummingboard
+ Compilar módulos e instalar
+ Resultados

Código de Kernel de Linux

Para compilar para la plataforma Hummingboard de Solid Run primero me baje el repositorio del código fuente del kernel de Solid Run con el siguiente comando en terminal:
`git clone https://github.com/SolidRun/linux-fslc.git`

Código de SPL y U-Boot

Para compilar el Secondary Program Loader y el U-Boot primero me baje el repositorio del código fuente para la versión de Solid Run con el siguiente comando en terminal:
`git clone https://github.com/SolidRun/u-boot-imx6.git`

Compilador cruzado

Para hacer la compilación cruzada instalé el toolchain arm-linux-gnueabihf-4.8 por lo que se hizo un soft link para que `make` pudiera llamarlo como arm-linux-gnueabihf-gcc.

Se requieren de las siguientes herramientas para compilar u-boot y generar el zImage del kernel.

```
sudo apt-get install u-boot-tools
sudo apt-get install lzop
```

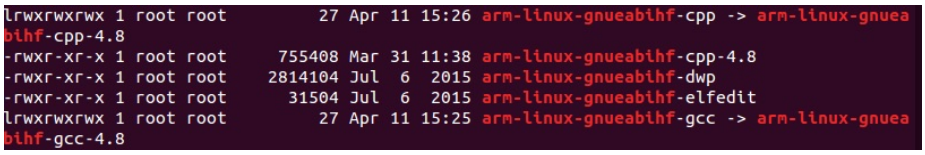

Figure 1: cross

Variables de Entorno

Para la compilación cruzada es necesario configurar las variables de entorno de la terminal usando los comandos export:

```
export ARCH=arm
export CROSS_COMPILE=/usr/bin/arm-linux-gnueabihf-
```

Compilar SPL y U-Boot

Una vez descargado el repositorio de U-Boot se tiene que compilar con la opción de `mx6_cubox-i_config` para que cree las configuraciones default para la HummingBoard.

```
cd u-boot-imx6
make mx6_cubox-i_config
make
```

Esto genera el archivo SPL (Secondary Program Launcher) y el u-boot.img. El siguiente paso fue crear la partición en la memoria SD. Se generó una partición ext4 con un offset de 2 MB para almacenar el SPL y el u-boot.img al inicio de la memoria. Las siguientes instrucciones copia el SPL a la primer dirección de la memoria y el u-image.img con un offset de 42KB.

```
sudo dd if=SPL of=/dev/sdb bs=1K seek=1
sudo dd if=u-image.img of=/dev/sdb bs=1K seek=42
```

Configuración .config

Para generar el .config por defecto de la Hummingboard se hacen los siguientes comandos en terminal:

```
make imx_v7_cbi_hb_defconfig
make ARCH=arm menuconfig
```

Este último comando genera un menu de Kconfig para cambiar las configuraciones de compilación del kernel. Para la compilación le hice cambios para agregar módulos para el VPU, IPU y VGA, al igual para la cámara por el puerto MIPI CSI2 y soporte para pantallas pequeñas TFT.

tft.jpg

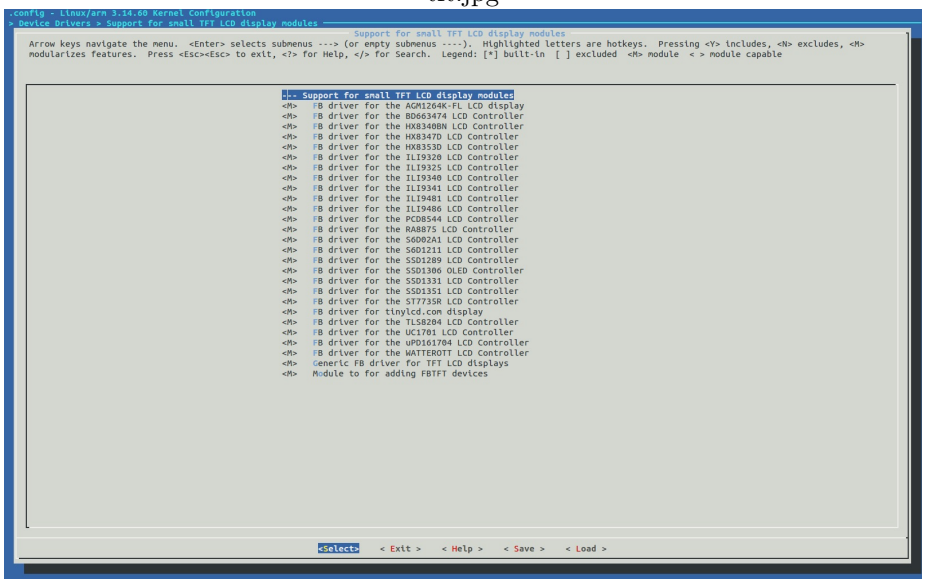

Figure 2: img1

Compilar Kernel y Device Tree Blob de la Hummingboard

La siguiente instrucción indica a make a compilar el kernel y guardarlo en el archivo comprimido zImage, al igual que los Device Tree Blobs para la HummingBoard. Los .dtb le indican al kernel de Linux el hardware que posee el SoC i.MX6 de Freescale.

```
make zImage imx6q-cubox-i.dtb imx6dl-cubox-i.dtb imx6dl-hummingboard.dtb imx6q-hummingboard.
```

Una vez que acaba de compilar el archivo zImage se encuentra en el ruta `/home/parallels/Downloads/linux-fslc/arch/arm/boot`. Los archivos .dtb se localizan en `/home/parallels/Downloads/linux-fslc/arch/arm/boot/dts`. Una vez compilado se deben instalar en la ruta raiz de la SD. En mi caso cree una carpeta llamada `~/bootpartition` y la monte en `/dev/sdb1`. Lo siguiente es copiar el zImage y los .dtb en `~/bootpartition/boot`.

camera.jpg

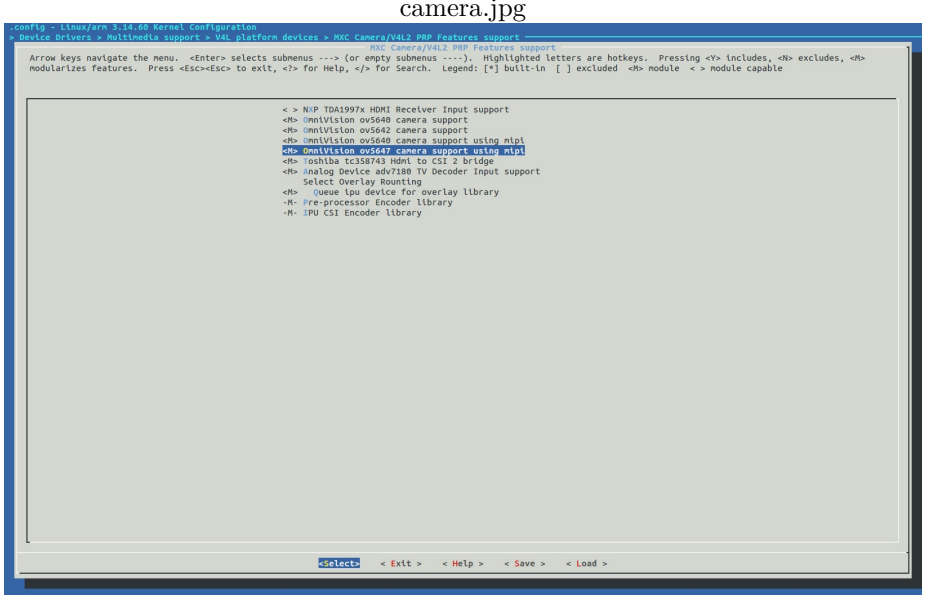

Figure 3: img2

platform drivers.jpg

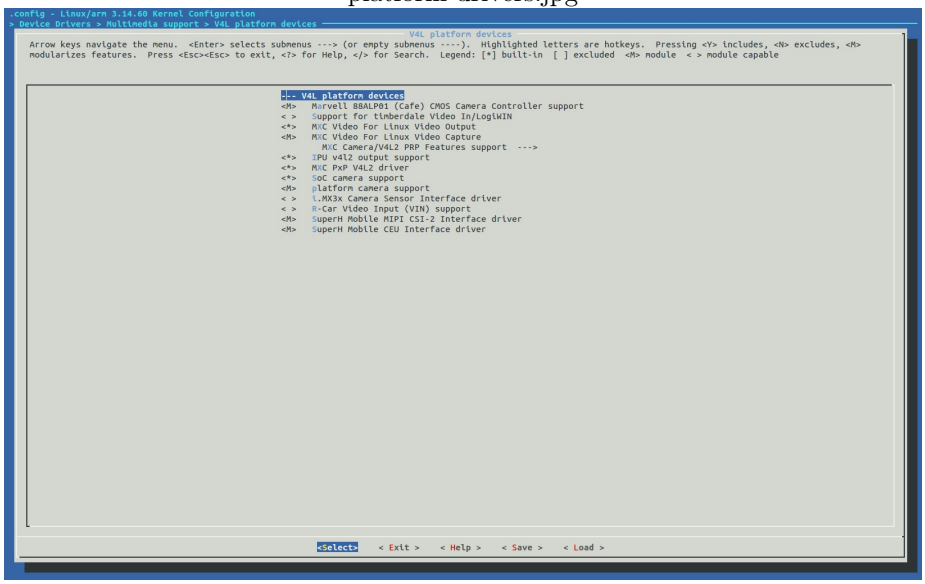

Figure 4: img3

cs2.jpg

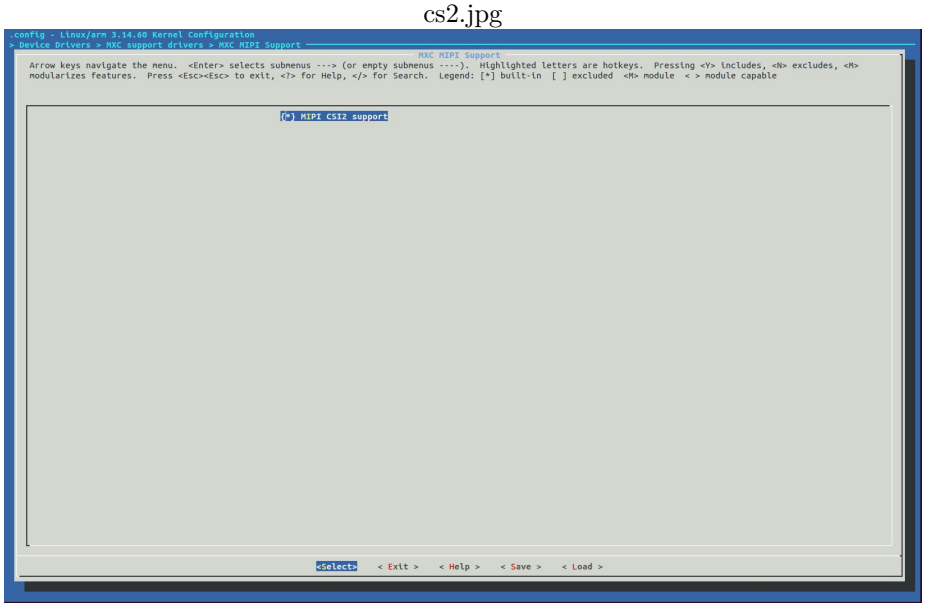

Figure 5: img4

Compilar módulos e instalar

El siguiente paso es compilar los módulos con la siguiente instrucción:

```
make modules
```

Una vez compilados los módulos se deben instalar en la partición de la SD.

```
make ARCH=arm modules_install INSTALL_MOD_PATH=(ruta donde este montada la sdb1 Ej. ~/bootpa
```

Por ultimo es necesario crear el archivo uENV.txt con el siguiente texto en la carpeta /boot para indicar a u-boot las variables de entorno.

```
mmcroot=/dev/mmcblk0p1 rootwait rw
mmcargs=setenv bootargs video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24,bpp=32 console=ttymxc0,
```

Resultados

Al prender la Hummingboard se inicia la pantalla de u-boot y se deja iniciar. Al final aparece lo siguiente en la pantalla:

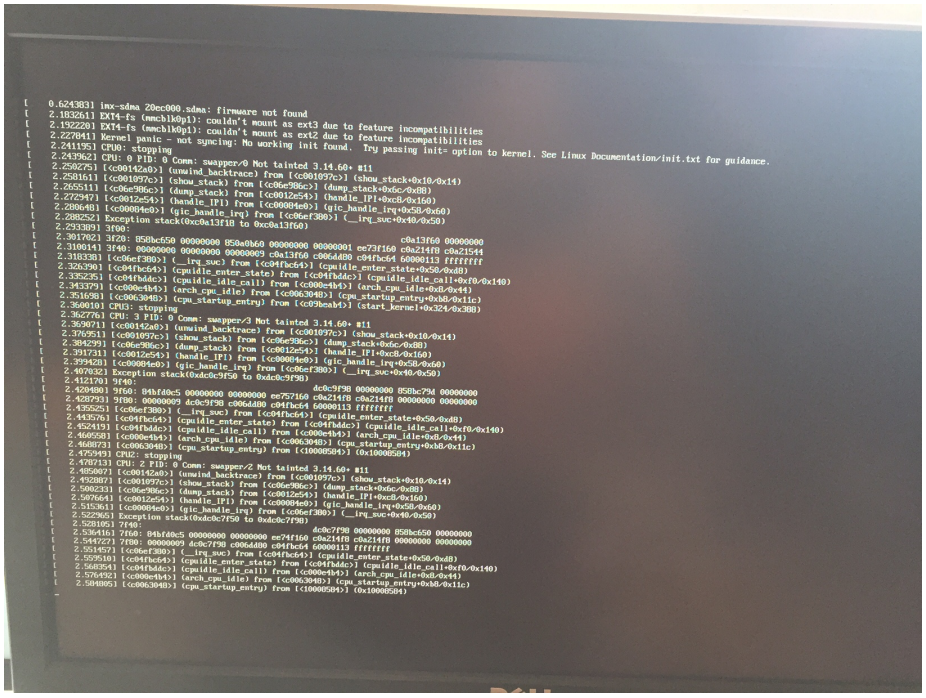

Figure 6: Resultado