

Guía instalación, uso y arreglos de Petalinux+SDSoC v2016.2

Raúl Ruiz Bueno

raulruizbueno@gmail.com

Mi intencion con esta guía es hacer algo parecido a un "Quick start" para montarle un Linux con Petalinux a una Plataforma definida en Vivado.

Recomiendo leer todo antes de ponerse a ello.

Previos:

-Tener una plataforma de Vivado hecha como se indica en el manual ug1146 Capítulo 2.

El proyecto de Vivado con el que aprendí esto:

[Zybo_hdmi_v2016.2](#)

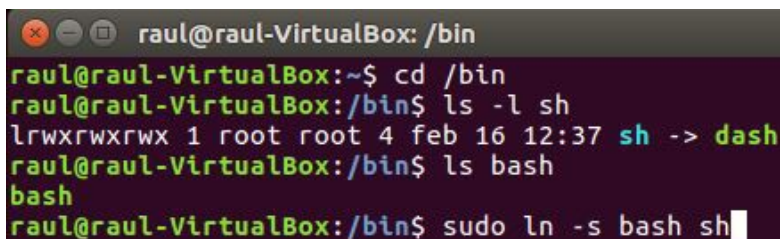
Comencemos:

Lo primero que tenemos que hacer es tener instalado **EXACTAMENTE LA MISMA** versión de Vivado suite y Petalinux, da igual si es v2015.4 , v2016.2 o la que sea, pero ambas instalaciones tienen que ser exactamente la misma versión. Mi recomendación es que sea por v2016.3 o superior, ya que un ingeniero de Xilinx me dijo que en esta versión se arreglaban algunas cosas. Este tutorial es para la 2016.2 así que realmente no se como de útil será para ti si escoges una versión diferente a esta (2016.2) Petalinux solo es instalable el Linux. En su día cuando yo lo hice estaba usando W10 para la suite de Vivado y Ubuntu 16 para la MV donde instalé Petalinux (recomiendo bastante usar Virtualbox, porque da menos la tabarra para el uso de carpetas compartidas, si vas a trabajar con windows no obvies esta facilidad)

Si usas una MV se generoso, sobre 60GB es lo suyo.

Antes de continuar, parece una chorrada, pero en la versión 2016.2 hay que tener el SO en inglés Americano. Porque si no, unas variables de entorno las pisa con cosas del idioma que tengamos.

Si no usas un Red Hat o Debian y lo haces en Ubuntu como hice yo, tambien tienes que hacer esto para que las ejecuciones de los scripts los haga con la consola bash y no con la consola dash



```
raul@raul-VirtualBox: /bin
raul@raul-VirtualBox:~$ cd /bin
raul@raul-VirtualBox:/bin$ ls -l sh
lrwxrwxrwx 1 root root 4 feb 16 12:37 sh -> dash
raul@raul-VirtualBox:/bin$ ls bash
bash
raul@raul-VirtualBox:/bin$ sudo ln -s bash sh
```

Te va a dar un error esto, antes de ejecutar la ultima orden tienes que borrar sh: sudo rm sh y vuelves a ejecutar la última línea

Una vez instalada la Suite de Vivado, instalamos Petalinux

Lo mejor es que sigas el manual ug1144 v2016.2:

- Página 10,11: donde te dicen que paquetes necesarios instalar segun tu SO(este paso es muy importante que instales todo, si no se llama como en el pdf busca como, porque si no, no te va a funcionar nada)
- Página 12: donde te dice cada comando a ejecutar y en que directorio instalar.

Llegados a este punto si todo lo tienes instalado ejecuta este comando (página 15):

```
source <path-to-installed-PetaLinux>/settings.sh
```

El path seguramente sea /opt/pkg/petalinux-v2016.2-final/

Que te pone en el entorno de Petalinux. **Esto lo tenemos que hacer**

SIEMPRE que queramos trabajar con Petalinux, si cerramos la terminal o apagamos el SO se va.

Bien, a estas alturas debes de tener una carpeta compartida entre Windows y Linux si estas trabajando como hice yo. Si lo estas haciendo todo el Linux pasa de esto.

Nos metemos en tarea.

Es recomendable que el proyecto de Vivado y el proyecto de Petalinux cuelguen del mismo directorio y que los paths no sean demasiado largos, para que los scripts no se disloquen. **Trabajar sobre una carpeta de dropbox es un error, porque muchas veces bloquea ficheros que esta sincronizando.**

Ahora es el momento de crear el directorio de trabajo, entramos dentro y copiamos nuestro proyecto de Vivado aquí.

Activamos el entorno de Petalinux si no lo tenemos activado:

```
source <path-to-installed-PetaLinux>/settings.sh
```

Creamos nuestro proyecto Petalinux (info ug1144 página 23):

```
petalinux-create --type project --template zynq --name Nombreproy
```

Entramos dentro del proyecto que acabamos de crear.

Le indicamos a Petalinux donde está la ruta de nuestro fichero de descripcion de hardware de la plataforma vivado (suele estar en la carpeta .sdk(donde este el .hdf)) con la siguiente orden, cuando salte el menú hay que poner las cosas como se indica en la página 25 del ug1144:

```
petalinux-config --get-hw-description=Path hasta .sdk
```

Ejemplo:

```
petalinux-config --get-hw-description=/home/raul/Desktop/zybo_hdmi_in/vivado/zybo_hdmi_in.sdk
```

Compilamos:

```
petalinux-build
```

Creamos el uImage:

```
petalinux-package --image -c kernel --format uImage
```

Cuando esto este completado vamos a la carpeta del proyecto Petalinux:
/images/linux aquí estarán todos los ficheros necesarios.

En esta versión cuando arrancamos Petalinux para la zybo el u-boot nos pide el fichero image.ub. Tenemos que configurarlo bien para que tenga la libreria libstdc++6 de lo contrario no nos dejará ejecutar la aplicación.

Para ello vamos a la página 25 del manual ug1146 y hacemos los pasos del 1 al 6.

Si todo se ha compilado bien, todos los ficheros que hemos sacado en la carpeta /images/linux necesarios para el arranque en linux, los pegamos en la carpeta /boot de nuestro proyecto de Vivado. Creamos nuevo proyecto en SDSoC usando este de vivado como plataforma (si no sabes como ve a los enlaces, ahí redirecciono a un video del proceso), compilamos todo y cuando copiemos los ficheros a la SD, tenemos que copiar el fichero image.ub que esta en la carpeta images/linux del proyecto Petalinux, si no el u-boot lo va a pedir.

Usuario y contraseña de Petalinux: root

El path para ejecutar las aplicaciones ya en Petalinux es:
/run/media/mmcblk0p1/

¡Felicidades ya tienes tu linux específico para tu plataforma de Vivado!

Anexo, errores comunes y cosas de interes:

- Si petalinux-build te da error, revisa los paquetes necesarios y que tengas el SO en inglés Americano.
- Si petalinux-package --image -c kernel --format uImage te da el siguiente error, tienes que hacer lo de la primera imagen de este documento que ya ha sido explicado el por qué:

```

raul@raul-VirtualBox:~/Desktop/petatemp$ petalinux-package --image -c kernel --format u
Image
/bin/sh: 1: arithmetic expression: expecting primary: ""
/bin/sh: 1: arithmetic expression: expecting primary: ""
[INFO ] package rootfs.cpio to /home/raul/Desktop/petatemp/images/linux
Generating filesystem description file for gen_init_cpio...
Creating /init symlink if required...ln: failed to create symbolic link '/home/raul/Des
ktop/petatemp/build/linux/rootfs/targetroot/init': File exists
Done.
directories...files...symlinks...device nodes...done.
Generating CPIO archive /home/raul/Desktop/petatemp/images/linux/rootfs.cpio ...done.
/bin/sh: 1: [: arm: unexpected operator
Image Name:
Created:      Mon Mar  6 17:04:22 2017
Image Type:   ARM Linux RAMDisk Image (gzip compressed)
Data Size:    2515571 Bytes = 2456.61 kB = 2.40 MB
Load Address: 00000000
Entry Point:  00000000
[INFO ] Update and install vmlinux image
[INFO ] vmlinux linux/kernel
[INFO ] install linux/kernel
[INFO ] package uImage
/bin/sh: 3: [: arm: unexpected operator
/bin/sh: 9: [: arm: unexpected operator
/bin/sh: 12: [: arm: unexpected operator
/bin/sh: 17: arithmetic expression: expecting primary: ""
/bin/sh: 18: [: arm: unexpected operator
[INFO ] install linux/kernel
raul@raul-VirtualBox:~/Desktop/petatemp$

```

- Si te da este error, pegar image.ub en la SD:

```

Hit any key to stop autoboot:  0
Device: sdhci@e0100000
Manufacturer ID: 41
OEM: 3432
Name: SD8GB
Tran Speed: 50000000
Rd Block Len: 512
SD version 3.0
High Capacity: Yes
Capacity: 7.2 GiB
Bus Width: 4-bit
Erase Group Size: 512 Bytes
reading image.ub
** Unable to read file image.ub **
U-Boot-PetaLinux>

```

- Si al ejecutar la aplicación te dice que no existe fichero o directorio, es que te falta la biblioteca libstdc++6 y tienes que compilarla para el fichero image.ub de la forma que se indica en el ug1146 página 25

Enlaces de interes:

[ug1146-2016.2](#)

[ug1144-2016.2](#)

[Flujo de trabajo para SDSoc- Crear plataforma desde 0](#)