IMPLEMENTACIÓN DE UNA RED ETHERNET PARA LA COMUNICACIÓN ENTRE SISTEMAS BASADOS EN FPGA

Leidy Johana Beltrán Usme

Ingeniería de telecomunicaciones

Director: Luis Fernando Castaño Londoño

INSTITUTO TECNOLÓGICO METROPOLITANO

Septiembre 2016

RESUMEN

En este documento se describe la implementación de una topología de red tipo estrella con sistemas de desarrollo basados en dispositivos FPGA. La implementación se realiza empleando un sistema de desarrollo Zedboard, el cual contiene una FPGA de la familia Zynq-7000 de Xilinx. Esta tarjeta posee conexiones para diferentes tipos de periféricos de entrada y salida. El objetivo del trabajo presentado es establecer la comunicación entre los sistemas de desarrollo por medio de una red Ethernet. Para la configuración de la conexión se efectúa la instalación en la Zedboard de una distribución del sistema operativo Linux embebido conocida como Petalinux, usando las aplicaciones y los controladores requeridos. El sistema operativo se carga en la RAM mediante una memoria SD y es el encargado de detectar periféricos y asignar los controladores correspondientes. Las tarjetas funcionan como clientes DHCP a través del uso de un servidor DHCP implementado en un ***router***. Esto garantiza la asignación de una dirección IP como identificador para cada FPGA. Finalmente, por medio de un ***switch*** se realiza la comunicación entre todas las FPGAs. Este producto es desarrollado en el laboratorio de Sistemas de Control y Robótica del ITM. La realización de este producto sirve como insumo para trabajos de grado y proyectos de investigación que requieran el uso de sistemas basados en FPGA en red, para la implementación de diferentes tipos de algoritmos. Para el desarrollo de esta práctica se utilizaron recursos disponibles en el laboratorio Microelectrónica y Nanotecnología del ITM (sede fraternidad).

Palabras clave: Internet de las cosas, Ethernet, FPGA, Sistema operativo embebido, Zedboard, Petalinux, Topología, Servicios de red, ***Switch***.

RECONOCIMIENTOS

Siento una enorme gratificación por los docentes del Instituto Tecnológico Metropolitano (ITM) por compartir su conocimiento conmigo y aportar en mi formación como profesional. El ITM me abrió sus puertas lo cual permitió contar con el acompañamiento, el espacio y las herramientas necesarias para adquirir el aprendizaje obtenido durante mi carrera. Ser exitoso como profesional va de la mano con sentir entusiasmo por la carrera elegida, esto conlleva a poner el máximo esfuerzo y desempeñar las labores con el mayor agrado. Ser profesional se trata de amar lo que se hace. De esta manera la carrera se convierte en una vocación y no en una obligación. Mediante muchas de las vivencias durante mi formación, logré adquirir el criterio sobre ser profesional de *cartón* y ser un profesional por pasión y convicción.

A mis amistades más cercanas: Marcela Carmona, Tatiana Arboleda y Alexander siento hacia ellos un agradecimiento muy especial por acompañarme durante este período de mi vida. A Sebastián Gallego quien me ayudó a aterrizar muchos de los planes que contemplo en mi proyecto de vida, el cual me acompañó durante todo el ciclo complementario de mi carrera y en particular en este proyecto.

A mi asesor por darme la oportunidad de trabajar en su propuesta, por la paciencia y por compartir conmigo sus conocimientos, los cuales fueron fundamentales para llevar a cabo este proyecto.

A mi familia por el apoyo y compañía constante. Cada escalón en que avanzo en mi vida son logros dedicados a ellos.

ACRÓNIMOS

IP Protocolo de Internet

PL Lógica Programable

PS Sistema de procesamiento

IOT Internet de las cosas

DNS Sistema de Nombres de Dominio

FTP Protocolo de Transferencia de Archivos

UTP Par Trenzado Sin Blindaje

LAN Red de Área Local

OSI Modelo de Interconexión de Sistemas Abiertos

SDK Kit de Desarrollo de Software

BSP Paquetes de Soporte de Placa

EIA Alianza de Industrias Electrónicas

TIA Asociación de la Industria de Telecomunicaciones

SSH Intérprete de Órdenes Seguro

UDP Protocolo de Datagrama de Usuario

DHCP Protocolo de Configuración Dinámica de ***Host***

HTTP Protocolo de Transferencia de Hipertexto

IEEE Instituto de Ingeniería Eléctrica y Electrónica

ANSI Instituto Nacional Estadounidense de Estándares

GPIO Chip Entrada/Salida de Propósito General

UART Transmisor-Receptor Asíncrono Universal

FPGA Arreglos de Compuertas Programables en Campo

ICMP Protocolo de Mensajes de control y Error de Internet

IJMNCT Revista Internacional de Red Móvil de Comunicaciones y Telemática

TABLA DE CONTENIDO

1. INTRODUCCIÓN

1.1 Generalidades

En la actualidad el internet o red de redes tiene presencia en la mayor parte de actividades habituales. La inclusión de la tecnología en el diario vivir ha evolucionado a grandes pasos. Por ejemplo, anteriormente las personas se conectaban a internet mediante los computadores, luego con los teléfonos inteligentes y ahora con los televisores, los relojes entre otros. Esto es porque ha llegado la era del internet de las cosas (IoT).

El internet de las cosas se trata de un entorno de conexión, donde las personas y aparatos (cosas en un lenguaje coloquial) interactúan entre sí con un entorno virtual de datos en el mismo espacio y tiempo. Pensar en que este hecho sea realidad sitúa a la humanidad en un entorno futurístico. Sin embargo, esto será posible gracias a la información suministrada por una infinidad de sensores que se alojarán en todo rincón donde habite el ser humano (Clúster ICT - Audiovisual de Madrid, 2013). Dichos sensores estarán incorporados a todos los dispositivos usados frecuentemente por las personas. A la vida del ser humano han llegado los sistemas embebidos como respuesta a la necesidad de tener artefactos conectados e integrados a la cotidianidad. Dispositivos como el horno microondas, los carros, los ascensores e instrumentos biomédicos funcionan a razón de pequeñas computadoras o microsistemas que no contienen una pantalla o un disco duro. No obstante, este control es ejecutado por los sistemas embebidos los cuales son responsables de darle funcionalidad a un aparato inactivo. Es tan amplio el campo de aplicación de estos, que provee al ser humano beneficios como el confort, salud, seguridad entre otros. La figura 1 muestra un resumen del IoT en la vida de las personas.

Figura 1. Resumen del internet de las cosas. (Cisgroupla, 2015).

De manera resumida, el objeto del internet de las cosas es la integración de sensores y dispositivos con los artefactos de uso cotidiano que se hallan conectados a internet mediante redes fijas o inalámbricas (Clúster ICT - Audiovisual de Madrid, 2013). El uso masivo de internet representa una ventaja para el IoT.

En la actualidad, donde la sociedad se encamina hacia el internet de las cosas, las aplicaciones creadas gracias a los sistemas embebidos apuntan hacia la conexión. Este hecho será posible por medio de la tecnología Ethernet.

Lo anteriormente expuesto, antecede la necesidad de explorar la funcionalidad y utilidad que ofrecen las redes Ethernet a los sistemas embebidos cuando se trata de conexión. La limitada información que se tiene en la universidad respecto al uso de las tarjetas de desarrollo Zedboard y particularmente el hecho de que no se ha explorado la importancia de establecer redes entre dichas tarjetas, ha originado el desarrollo expuesto. Desde el

área de telecomunicaciones, se ha efectuado el análisis e implementación de una topología de red tipo estrella con sistemas embebidos. Mediante la red de FPGAs se pretende explorar, identificar y detallar la funcionalidad y las ventajas que ofrecen la conexión en red a través de tecnología Ethernet. El uso del SoC programable de la familia Zynq-7000 de Xilinx basado en FPGA ofrece la ventaja de trabajar con un dispositivo en el cual su hardware puede ser adecuado a determinada necesidad.

La solución propuesta servirá como insumo en el laboratorio de Sistemas de Control y Robótica en el desarrollo de trabajos de grado y proyectos de investigación que requieran el uso de sistemas basados en FPGA conectados en red y para la implementación de diferentes tipos de algoritmos.

1.2 Objetivo general

Analizar el funcionamiento de sistemas de desarrollo basados en FPGA conectados en red a través de tecnología Ethernet con el fin de explorar, identificar y detallar las características y ventajas que confiere este tipo de conexión gracias al uso de servicios y protocolos de red.

1.3 Objetivos específicos

- Instalar Petalinux en la Zedboard con el propósito de facilitar la interacción y el control de los diferentes periféricos de la tarjeta de desarrollo.

- Implementar una topología de red tipo estrella con la finalidad de establecer conexión entre tarjetas de desarrollo Zedboard, ***switchs***, ***routers*** y ***hosts*** a través de Ethernet.

- Identificar las diferentes tecnologías necesarias para establecer una comunicación adecuada entre sistemas de desarrollo basados en FPGA.

- Instalar y verificar los servicios, protocolos y aplicaciones de red DHCP, SSH, FTP y HTTP a través del intercambio de información entre los diferentes dispositivos activos de la red y de esta manera evaluar la utilización de estos servicios en comunicaciones entre tarjetas basadas en FPGA.

1.4 Organización del trabajo

En la sección 1, se encuentra generalidades del proyecto como la conveniencia de establecer conexiones en red entre tarjetas basadas en FPGA. Dentro de este marco es presentado el problema, las preguntas que se pretenden responder y la explicación del beneficio que genera el producto realizado.

En la sección 2, se explica el estado del arte del uso de los sistemas de desarrollo basados en FPGA mediante tecnología Ethernet y la importancia de obtener conectividad en el marco del internet de las cosas.

En la sección 3, se muestra una metodología descriptiva que evidencia las fases de desarrollo del establecimiento de la topología de red.

En la sección 4, se detalla el proceso de validación de la conexión Ethernet tomando como referencia el Modelo OSI. También se prueba el GPIO-demo que trae consigo el Petalinux.

En la sección 5, se presenta las conclusiones de acuerdo a las metas trazadas al inicio del proyecto. Además, se detallan algunos aspectos a mejorar hallados durante la ejecución del proyecto. También se mencionan posibles trabajos futuros con base al desarrollo expuesto.

2. MARCO TEÓRICO

2.1 Revisión del estado del arte

2.1.1 Plataforma de comunicación Ethernet para dispositivos sintetizados en FPGA de Xilinx

Artículo con autoría de Rodrigo Neri de Souza, Daiana Nascimiento Muniz y André Vaz da Silva Fidalgo. Fue expuesto en la conferencia internacional sobre la computadora como herramienta (EUROCON) en el año 2011. El escrito citado expone la implementación de una plataforma de comunicación para dispositivos sintetizados dentro de un sistema de tarjetas FPGA de Xilinx. La plataforma de comunicación se muestra como una solución eficiente y adaptable para los dispositivos mencionados (Neri de Souza, Nascimento Muniz, & Vaz da Silva Fidalgo, 2011).

Especialmente, la plataforma provee una comunicación de datos a través de Ethernet y el medio para controlar dispositivos lógicos programables. El diseño de la plataforma se fundamenta en un sistema embebido tradicional tanto software y hardware, sintetizado en FPGA. El hardware se basa en un procesador *microblaze* y se implementa utilizando XPS. La plataforma es modulable, es decir es compatible con muchos dispositivos por ejemplo los sintetizados en FPGA. Actualmente, la plataforma se emplea para proporcionar acceso remoto y es compatible con la IEEE 1149.1 (Neri de Souza, Nascimento Muniz, & Vaz da Silva Fidalgo, 2011).

Los resultados del proyecto han sido positivos. Se pretendía identificar si era posible instalar Petalinux en una FPGA. Además, el desarrollo se enfoca en la capa de servicios de TCP/IP, en el GPIO y la puesta en operación de los traductores. Uno de los aportes que ofrece esta solución es el mantenimiento remoto de los dispositivos electrónicos. Se

verificaron las capas de servicios del modelo TCP/IP con el propósito de validar la aceptación de los mismos a través del puerto Ethernet de las tarjetas. Igualmente, se ha implementado un traductor que se encarga de comunicar el Petalinux con la tarjeta, cabe aclarar que el ***kernel*** de Petalinux también cumple con esta función (Neri de Souza, Nascimento Muniz, & Vaz da Silva Fidalgo, 2011).

Para trabajos futuros, se recomienda diseñar un bloque que incluya todos los componentes y servicios necesarios para establecer comunicación entre tarjetas a mediante una red. Por ejemplo, incorporar la imagen ISO del Petalinux y la configuración adecuada para el GPIO, de esta manera se logra simplificar mucho más la configuración e implementación de este tipo de comunicación (Neri de Souza, Nascimento Muniz, & Vaz da Silva Fidalgo, 2011).

2.1.2 Una simple implementación de una pila Ethernet en lenguaje VHDL para habilitar la reconfiguración lógica en una FPGA

Memoria expuesta en la conferencia International sobre computación reconfigurable y FPGAs, llevada a cabo en Cancún en el año 2011. Marcus R. Perret e Izzat Darwazeh exponen sobre la implementación del hardware de una pila de red la cual permite el uso de Ethernet para enviar datos, paquetes de control y reconfigurar la lógica de interconectado (R Perrett & Darwazeh, 2011).

Se muestra el diseño y la verificación de un método para encapsular los protocolos Ethernet dentro de la pila de red. En resumen, la implementación fue probada configurando el hardware de las FPGA a través de la red. Por ejemplo, una *tarjeta A* transfiere su configuración de hardware a una *tarjeta B* por medio de una comunicación Ethernet (R Perrett & Darwazeh, 2011).

Las tarjetas ya tienen la interfaz de red, pero en el desarrollo se crean bloques que puedan ser invocados al programar en lenguaje VHDL. De esta manera se pone a disposición toda la configuración que ya se tiene diseñada para la comunicación de datos en una red Ethernet. Inclusive, con esta técnica no se requiere la instalación de un sistema operativo embebido sino que propiamente se emplea un bloque que contenga el algoritmo para establecer la comunicación (R Perrett & Darwazeh, 2011).

2.1.3 Implementación del estándar Gigabit Ethernet usando FPGA

Artículo publicado en el 2012 por IJMNCT Vol.2, No.4. El ***paper*** fue escrito por V.R. Gad, R. S. Gad y G.M. Naik pertenecientes al Departamento de Electrónica de la Universidad de GOA (India). Este artículo presenta los resultados de la implementación del estándar Gigabit Ethernet en un dispositivo FPGA. El diseño utiliza una FPGA Stratix-II GX de Altera y soporta la tasa de transferencia de datos de 10/100/1000Mbps. El uso de FPGA ofrece la ventaja de modificar la funcionalidad de la plataforma para desarrollar diferentes tareas. El rendimiento de este diseño es evaluado para Gigabit Ethernet empleando como medio la fibra óptica y el cobre.

También da un repaso de las tecnologías Ethernet de 10/100/1000Mbps y describe su implementación sobre la FPGA Stratix-II GX de Altera. El software Quartus II se usa para sintetizar y crear el archivo *.soft (*de programación). El diseño es descargado al chip de la FPGA a través del puerto JTAG. Se analiza el rendimiento de este diseño y la tasa de línea se ubica en 76.19% mínimamente para paquetes de 64 Bytes y se aproxima al 100% para tramas de 9600 Bytes. El ***throughput*** es el más pequeño para tramas de 64Bytes y se acerca a 1 Gbps para tramas de 9600 Bytes. Para una toma particular de ancho el ***throughput*** y la tasa de línea permanecen casi del mismo tamaño para todos los estándares de Gigabit Ethernet. También, a medida que el tamaño de la trama aumenta, el tiempo de transmisión total empieza a disminuir y permanece constante para tramas 1024 Bytes. Este diseño se considera robusto. Los autores indican que también

desarrollaron una plataforma experimental para introducir errores en la red, pero para trabajos futuros se planea incluir en dicha plataforma la detección de errores y el análisis de las correcciones (Gad, Gad, & Naik, 2012).

2.2 Importancia de la conectividad enmarcado en el internet de las cosas

En el IoT el término *conexión* cobra importancia, ya que busca que todos los objetos de uso cotidiano estén conectados a internet. Cuando el concepto fue mencionado por primera vez, pensar en que fuera realidad era casi una utopía. Sucesos actuales como el desarrollo de los dispositivos habilitados por IP, la masificación del uso de banda ancha, la llegada del protocolo IPV6 y el número de aplicaciones conectadas a la interconexión de objetos son factores que suscitan el IoT (Alonso Arévalo, 2016). El IoT conecta en red las personas, procesos, datos y cosas con el fin de transformar la información en forma de, experiencias, nuevas capacidades y oportunidades generando grandes beneficios desde el ámbito social y económico (Alonso Arévalo, 2016).

Los dispositivos electrónicos han avanzado a velocidades extremas en las últimas décadas. Las máquinas convencionales ejecutan y procesan comandos, el futuro de las máquinas se proyecta en que aparte de las funciones mencionadas, estas puedan predecir y prever lo que los seres humanos desean (Alonso Arévalo, 2016).

Los sistemas embebidos son un pilar fuerte en el ámbito del IoT. La lógica embebida proporciona el control remoto, la monitorización y la oportunidad de inspeccionar y analizar fuentes de datos. La necesidad de conectar estos aparatos a Internet se ha vuelto prioridad y al día de hoy los fabricantes de circuitos integrados y componentes electrónicos tienen una amplia gama de productos que cumplen dicha finalidad.

Las evidencias anteriores respaldan el objeto del presente desarrollo. Si bien es cierto, se emplearon tarjetas y configuraciones diferentes pero los resultados muestran una certeza

de funcionamiento de la red Ethernet. De acuerdo a la visión de IoT, los desarrollos relacionados con la conexión de sistemas embebidos a través de Ethernet tienen vigencia por la proliferación de dispositivos electrónicos a permanecer conectados. La innovación permite emplear estos dispositivos que se caracterizan por ser reconfigurables (respecto al hardware) lo cual posibilita adecuar una infinidad de aplicaciones a su estructura.

3. METODOLOGÍA

Durante la ejecución del proyecto se ha empleado un método teórico-práctico con el fin de cumplir los objetivos establecidos. En la figura 2 se muestra un resumen de las diferentes etapas necesarias para la implementación de la topología de red.

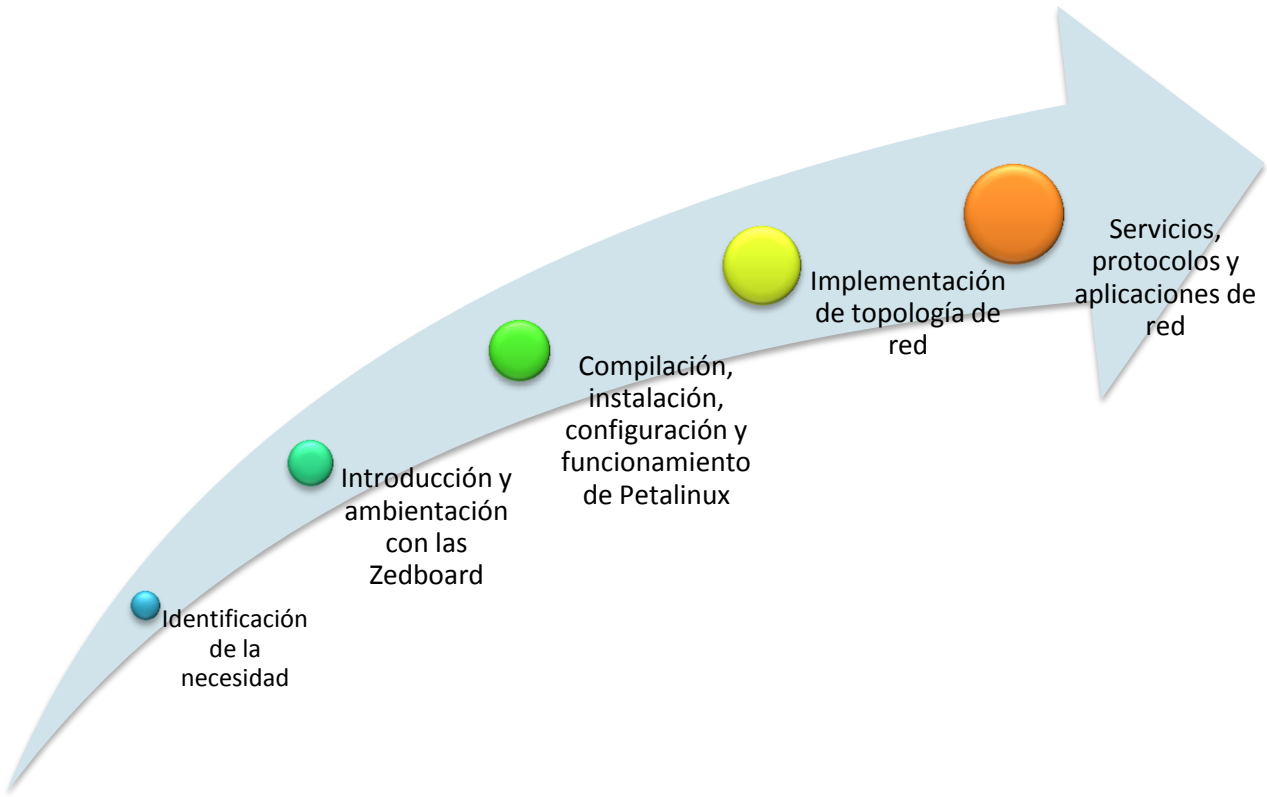


Figura 2. Fases de desarrollo del producto.

3.1 Identificación de la necesidad

La limitada información que se tiene en la universidad respecto al uso de las tarjetas Zedboard y particularmente el hecho de que no ha sido explorada la importancia de establecer redes entre las mismas, ha originado el presente desarrollo. Desde el área de telecomunicaciones, se ha efectuado el análisis e implementación de una topología de red tipo estrella con sistemas embebidos.

Este desarrollo se fundamenta en la intención de explorar, identificar y detallar las características y ventajas que ofrecen las tarjetas Zedboard conectadas en red mediante tecnología Ethernet.

La posibilidad de trabajar en el laboratorio de Sistemas de Control y Robótica del ITM y la existencia del SoC programable de la familia Zynq-7000 de Xilinx basado en FPGA representa una ventaja considerable para la realización del producto.

3.2 Introducción y ambientación a las Zedboard

Los conceptos básicos sobre las tarjetas de desarrollo Zedboard se han tomado de la información del fabricante. El fabricante en su sitio ***web*** contiene bien documentado los detalles técnicos, las guías de usuario y las herramientas compatibles con las tarjetas (Xilinx INC, s.f.).

La tarjeta Zedboard es un kit de desarrollo completo para los diseñadores interesados en la exploración de FPGA de la familia Zynq-7000 de Xilinx. La placa contiene todas las interfaces necesarias y las funciones de apoyo para permitir una amplia gama de aplicaciones (Xilinx INC, s.f.). En la figura 3 se muestran algunos detalles técnicos de la tarjeta (Xilinx INC, s.f.).

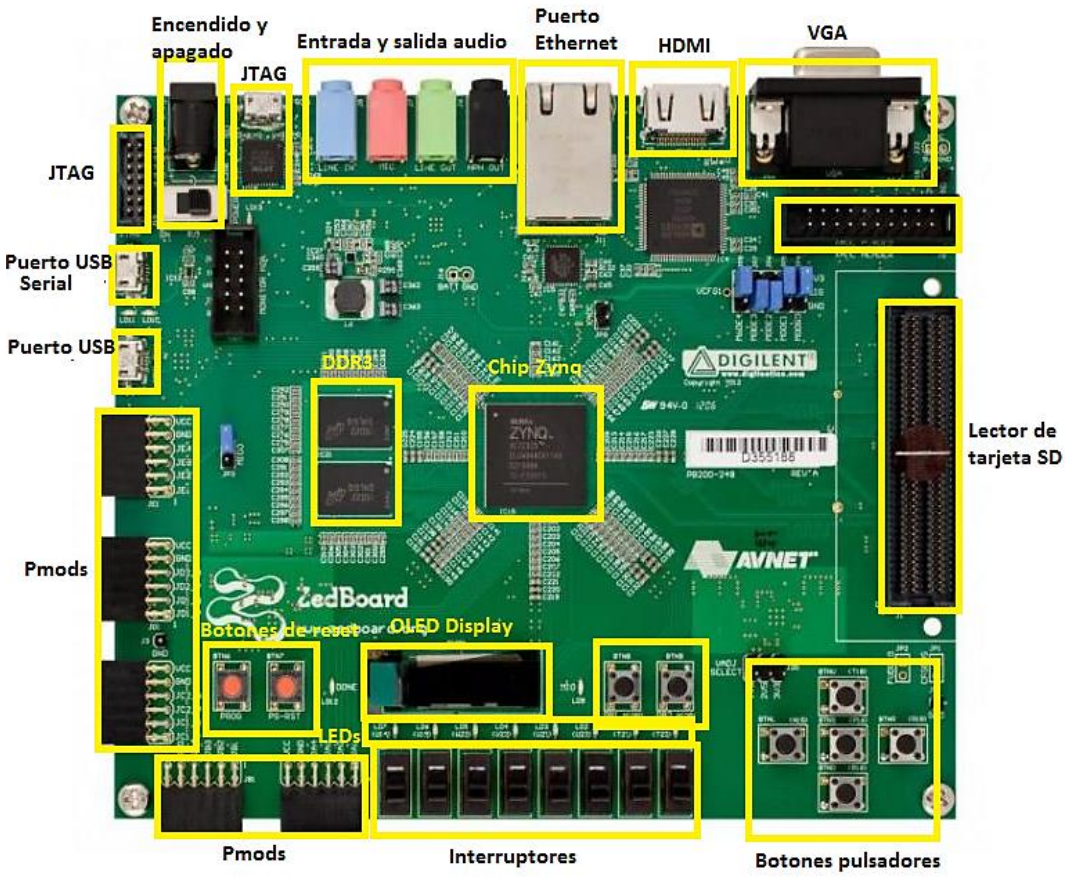


Figura 3. Placa de desarrollo Zedboard y sus componentes. Adaptada de (Xilinx INC, s.f.).

Procesador

- Zynq™-7000 AP SoC XC7Z020-CLG484-1

Memoria

- Memoria RAM 512 MB DDR3 (módulos de 128 Mb x 2).
- 256 MB de memoria QSPI Flash.

Interfaces Entrada/Salida

- USB-JTAG.
- Interfaz de red de 10/100/1G Ethernet.
- USB OTG 2.0.

- Lector de tarjetas SD.
- USB 2.0 FS USB-UART.
- 5 Conectores PMOD.
- 1 Conector LPC FMC.
- 1 AMS ***Header***.
- 2 Botones de ***reset*** (1 para PS, 1 para PL).
- 7 Botones pulsadores (2 PS, 5PL).
- 8 Interruptores conectados únicamente al PL.
- 9 ***Leds*** para el usuario (1 PS, 8 PL).
- 1 DONE ***led***.

Relojes en placa:

- 33.33 MHz para el PS.
- 100 MHz dirigido al PL.

Vídeo/Audio:

- Salida HDMI.
- Salida VGA de 12-bit.
- 128x32 OLED ***display***.
- Entrada/Salida de audio, salida de cascos y entrada de micrófono.

Otros:

- ***Switch*** de encendido y apagado
- Regulador 12V @ 5V AC/DC

3.3 Compilación, instalación, configuración y funcionamiento de Linux embebido (Petalinux)

En las tarjetas de desarrollo Zedboard activas de la red se ha instalado una distribución embebida del sistema operativo Linux.

El sistema operativo propicia la interacción y el aprovechamiento de los componentes de las Zedboard. Ejemplo de ello es la utilización de los servicios y protocolos de red que vienen incorporados en el sistema operativo. El uso del sistema operativo ha evitado la complejidad de emplear un lenguaje de bajo nivel. Esta herramienta favorece a una persona con conocimientos en ***networking*** y sin la experiencia en hardware configurar una red de datos empleando las tarjetas de desarrollo.

Petalinux es una versión desarrollada por Petalogix y empleada como sistema operativo embebido. Xilinx como fabricante de la Zedboard evidencia la alta compatibilidad que tiene con los chips Zynq. Petalinux es un sistema operativo embebido estable, probado y documentado. Como lo indica Xilinx, proporciona las herramientas necesarias para crear e implementar soluciones con Linux embebido en sistemas de procesamiento de Xilinx (Xilinx INC, s.f.). Básicamente, Petalinux permite un desarrollo *llave en mano* entre tarjeta y sistema operativo. A continuación, se detallan los pasos más relevantes en esta etapa:

3.3.1 Preparación de la estación de trabajo (anfitrión)

En una estación de trabajo que contenga Kubuntu 14.04 como sistema operativo, se crea la imagen de Petalinux. Desde el sitio ***web*** del fabricante se descarga la versión de Petalinux 2015.4 y Avnet Diligent Zedboard (BSP) 2015.4 (Xilinx INC, s.f.). En el directorio */opt* normalmente se alojan paquetes de aplicaciones de terceros o suplementarios. La ubicación mencionada ha sido destinada para colocar el BSP, las herramientas de configuración y la imagen de Petalinux generada para la Zedboard entre otros. Antes de iniciar la creación de la imagen, se debe tener la última versión de todos los paquetes de

Kubuntu. También es necesario instalar herramientas y bibliotecas requeridas para el correcto funcionamiento de Petalinux como *tofrodos, iproute, tftpd-hpa, gawk y gcc git-core y algunas herramientas de red como libncurses5-dev-dev zlib1g libssl-dev flexión bisontes lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc ++ 6 libselinux1* (Embedded Centric, s.f.).

Igualmente es requisito instalar la aplicación *gtkterm* que corresponde a la terminal para acceder a la Zedboard cuando ya se tenga la imagen del sistema operativo lista.

3.3.2 Correr el instalador de Petalinux

Durante este procedimiento se corre el instalador del Petalinux con el fin de extraer el código fuente del ***kernel*** de Petalinux y toda la cadena de herramientas (Embedded Centric, s.f.).

3.3.3 Creación de nuevo proyecto de Petalinux

La creación de un nuevo proyecto de Petalinux se ejecuta con el propósito de instalar el BSP de la Zedboard. El BSP de la Zedboard es un diseño de referencia que sirve como base para crear nuevos proyectos. Inclusive, el BSP posibilita al diseñador centrarse en sus nuevas aplicaciones ahorrando tareas de configuración de bajo nivel. En el directorio donde se almacenan los ficheros generados durante la instalación, se debe hallar los siguiente (Embedded Centric, s.f.):

- Build: contiene el ***Kernel***, árbol de dispositivos y archivos raíz del sistema.
- Components: contiene aplicaciones, librerías, módulos del ***kernel***, ***drivers*** de dispositivos Xilinx y códigos genéricos.
- Hardware: contiene Vivado y proyectos SDK para la base del diseño.
- Hw-description: contiene metadatos para el hardware.
- Pre-built: contiene compiladores, ejecutables y flujo de datos.

- Subsystem: contiene archivos de configuración para los diferentes componentes del sistema.

3.3.4 Configuración de los archivos raíz del sistema

Esta acción se realiza con el fin de configurar los componentes del sistema de archivo tales como herramientas de utilidad, librerías, aplicaciones entre otros. Además de adicionar la aplicación *GPIO-demo* que viene por defecto con el Petalinux. GPIO-demo ha sido ejecutada con el fin de probar la funcionalidad de los ***leds***, pulsadores y ***switchs*** contenidos en la placa de desarrollo (Embedded Centric, s.f.).

3.3.5 Construcción de imagen de Petalinux

En esta etapa ya se tiene el núcleo de Linux gracias a la instalación del BSP en pasos anteriores. De esta manera la contrucción de la imagen se puede iniciar solo con escribir el siguiente comando: *petalinux-build*.

Este punto es crucial para la instalación de Petalinux. Al terminar este proceso, no deberían presentarse errores, ni evidenciar ausencia de paquetes o servicios. Se puede considerar como posibles causas de error la inadecuada actualización de la versión de paquetes y librerías de Kubuntu durante la preparación de la estación de trabajo, la incompatbilidad entre las diferentes versiones de Kubuntu, vivado, Petalinux y Avnet Diligent Zedboard o sencillamente la omisión de algunos comandos esenciales para el correcto funcionamiento del sistema.

3.3.6 Generación de paquete *BOOT.BIN*

Para iniciar Petalinux en las Zedboard se conocen diferentes métodos, en este caso ha sido iniciado desde una memoria SD. El *BOOT.BIN* contiene los siguientes archivos (Embedded Centric, s.f.):

- Download.bit: corresponde al archivo de flujo de datos para la parte PL de la tarjeta.
- Zynq_fsbl.elf: corresponde a la primera etapa del gestor de arranque LPSA para la parte del PS de la tarjeta.
- U-BOOT.ELF: corresponde al gestor de arranque universal, que se encarga de cargar la imagen del núcleo de Linux.

El BOOT.BIN se genera mediante la configuración de la herramienta *petalinux-packaje*. Dicha herramienta proviene de SDK Xilinx, el cual debe ser instalado antes de proceder con el paso principal de este apartado (Embedded Centric, s.f.).

Finalmente se obtiene un archivo *BOOT.BIN* e *image.ub* que corresponde a la imagen del ***kernel*** de Linux. Posteriormente, ambos se copian a la tarjeta SD.

3.6.7 Inicio de Petalinux desde la Zedboard

Al momento de iniciar Petalinux desde la tarjeta SD, los puentes JP9 (MIO4) y JP10 (MIO5) situados en la placa Zedboard deben estar a nivel lógico alto es decir a 3.3V. Asimismo, se debe establecer conexión con cable USB entre el equipo anfitrión y el puerto UART-USB de la tarjeta (Embedded Centric, s.f.).

Gtkterm proporciona el acceso al puerto serial de la Zedboard. Se ingresa a la pantalla de arranque de Petalinux, por defecto se emplean las siguientes credenciales:

- Avnet entrada-Digilent-ZedBoard-2015_4: root
- Contraseña: root

3.4 Implementación de la topología de red

Anteriormente, la información era transmitida desde el transmisor al receptor gracias a las rutas establecidas por la conmutación de circuitos. A través de este tipo de comunicación se creaba un canal que soportaba en tiempo real un servicio como voz o video y era liberado luego de la terminación de dicha sesión. La creciente demanda de servicios dio origen a la conmutación de paquetes. Esta conmutación hace referencia al mecanismo que permite que el canal sea compartido por muchos usuarios al mismo tiempo. También, está orientada a las aplicaciones de datos, ya que es muy eficiente cuando se trata de enviar y recibir datos como archivos, páginas ***web*** entre otros. En sí aplica para información que puede ser tratada con cierto retardo y que este hecho no afecta el objetivo de la comunicación.

Cuando se pretende crear una red de datos entre dispositivos es necesario emplear una estructura que permita el intercambio de información, esta configuración se conoce como topología de red. La topología de red puede ser de tipo lógica que se encarga de determinar la distribución espacial del medio de transmisión, de las máquinas y de los dispositivos de red entre otros. Dicha topología se enfoca en la manera o modo en que los dispositivos intercambian y gestionan datos a través del medio de transmisión que interconecta los dispositivos. El uso de determinada topología o estructura influye en el comportamiento de los dispositivos conectados en red. Ejemplo de ello es la capacidad de expandirse o ser escalable, la velocidad de transmisión, los tiempos de llegada entre otros.

Los ***switchs*** son equipos que operan bajo el estándar IEEE 802.3 (Ethernet). Los dispositivos en mención cumplen la función de interconectar las tarjetas Zedboard en la topología de red a través del medio cableado (UTP). Una de las ventajas es la funcionalidad de sus puertos, la utilización de los mismos responde a la necesidad de los usuarios. A modo de ejemplo, los puertos pueden operar a diferentes velocidades como 10/100 otros a 10/100/1000 (González, 2013). También, si se emplea un conector

especializado conocido como ***Gbic***, tienen la capacidad de recibir como medio de transmisión la fibra óptica. Cabe mencionar que el ***switch*** cuenta con 2 técnicas de transmisión. La técnica de reenvío directo (***cut-through***) consiste en que determinado puerto del ***switch*** no lee la trama de datos completa, sino que lee la dirección MAC y transfiere la trama al puerto de destino. La técnica de almacenamiento y reenvío (***store and forward***) se fundamenta en que un puerto recibe los datos y a su vez almacena la trama completa en un ***buffer*** para luego reenviarla al puerto destino. La utilización de la segunda técnica permite realizar algunas comprobaciones de error (control de flujo de datos) antes de enviar datos al puerto de destino (González, 2013).

En la red implementada el switch es el nodo central, se encarga de unir y transferir datos entre las tarjetas y los hosts. Para dicha tarea los switchs procesan la información contenida en la cabecera de la trama Ethernet. Este proceso se ubica en la capa 2 de enlace del modelo OSI. En dicha capa se lleva a cabo el proceso de acceso al medio a través de Ethernet. Por consiguiente, en la topología estrella el ***switch*** es el equipo central.

Ha sido establecida una configuración estrella porque mediante el uso del nodo central se facilita la comunicación de los dispositivos sobre Ethernet. En el desarrollo se ha considerado el estándar ISO/TIA/EIA-568-A. Dicho estándar recomienda la implementación de la topología estrella en un sistema de cableado estructurado. Esta estructura admite cambios a nivel aplicativo sin afectar el cableado físico lo cual representa ahorro en dinero y tiempo.

La topología estrella permite una fácil administración y control del flujo de datos ya que cada extremo del cable conecta dos dispositivos de la red. La estructura favorece el proceso de conexión o reconexión, es decir en caso de que alguna de las tarjetas Zedboard

no funcionaba correctamente se hacía fácil reemplazarla en comparación con otro tipo de topologías como la estructura bus.

El ***switch*** por sí solo no permite la conexión con otros segmentos de la red. Se plantea entonces el uso de un ***router*** con el objeto de probar la conectividad entre dos redes diferentes.

3.5 Servicios, protocolos y aplicaciones de red en comunicaciones entre tarjetas basadas en FPGA

Como etapa final del desarrollo se instalaron, configuraron y verificaron varios servicios, protocolos y aplicaciones de red. Este proceso se ejecuta con la intención de comprobar la compatibilidad y efectividad de éstos cuando son implementados en sistemas de desarrollo basados en FPGA.

Como se ha mencionado, Petalinux trae consigo diversas ventajas para las tarjetas Zedboard. Una de ellas es el acceso a los servicios y protocolos de red incluidos en el sistema operativo. A continuación, se indican los servicios, protocolos y aplicaciones implementados en las tarjetas.

3.5.1 Dropbear

La aplicación de red *Dropbear* se compone de los paquetes *dropbear* y *openssh-sftp-server dropbear*. La aplicación funciona sobre el servidor SSH. En la práctica realizada, dropbear (SSH) se ha utilizado para ingresar remotamente a los dispositivos de la red. Dicha aplicación es integrada al código fuente del ***kernel*** de Petalinux.

3.5.2 Busybox HTTPD

El programa *Busybox* incorpora dentro de sus utilidades el programa servidor Apache HTTPD (HTTP ***Daemon***). Busybox HTTPD generalmente se usa en sistemas con Linux

embebido. En la práctica realizada, Busybox HTTPD (servidor ***web***) ha sido empleada para ingresar a una página ***web*** que trae por defecto Petalinux. Dicha aplicación es integrada al código fuente del ***kernel*** de Petalinux.

3.5.3 Servidor DHCP

Este servicio de red posibilita a los dispositivos pertenecientes a una red obtener direcciones IP de manera automática desde un servidor DHCP.

El servicio DHCP que viene incluido en el Petalinux permite al ***host*** central operar como servidor DHCP. De esta manera, el ***host*** central asigna direcciones IP a las tarjetas que tengan Petalinux en ejecución y demás dispositivos que se hallen configurados como clientes DHCP. Sin embargo, esta función fue delegada en un ***router***.

El ***router*** CISCO 2901 opera en la capa de red (capa 3) del modelo OSI. La funcion pricipal radica en la capacidad de encaminar los paquetes de red por la ruta más adecuada. En la red implementada, dicho equipo tiene la función de interconectar las 2 redes LAN establecidas y por consiguiente lograr la comunicación entre todos los dispositivos pertenientes a ambas redes. Las tarjetas y los ***hosts*** fueron configurados como clientes DHCP que reciben direccionamiento IP entregado dinámicamente por el servidor DHCP configurado en el ***router.***

3.5.4 Servidor FTP

Este servicio permite transferir información guardada en ficheros desde un ordenador local a otras máquinas remotas. FTP es de uso frecuente en la internet. Los usuarios en su cotidianidad suelen utilizar algunos enlaces de descarga que son direccionados a un host que opera como servidor FTP. Dicho proceso es transparente para el cibernauta, sin embargo, hace parte del servicio de transferencia de ficheros. En la práctica realizada, FTP fue empleado para transferir ficheros fácilmente entre el ***host*** central, las Zedboard, los

demás ordenadores y viceverza. El acceso a las Zedboard se hace a través de la terminal del ***host*** donde se alojan los ficheros a compartir. Las Zedboard se comunican con los ***hosts*** vía SSH o mediante la aplicación *gtkterm* del puerto serie.

4. RESULTADOS Y DISCUSIÓN

4.1 Validación de conectividad Ethernet

La topología de red implementada obedece a la estructura estrella. La figura 4 ilustra dicha estructura:

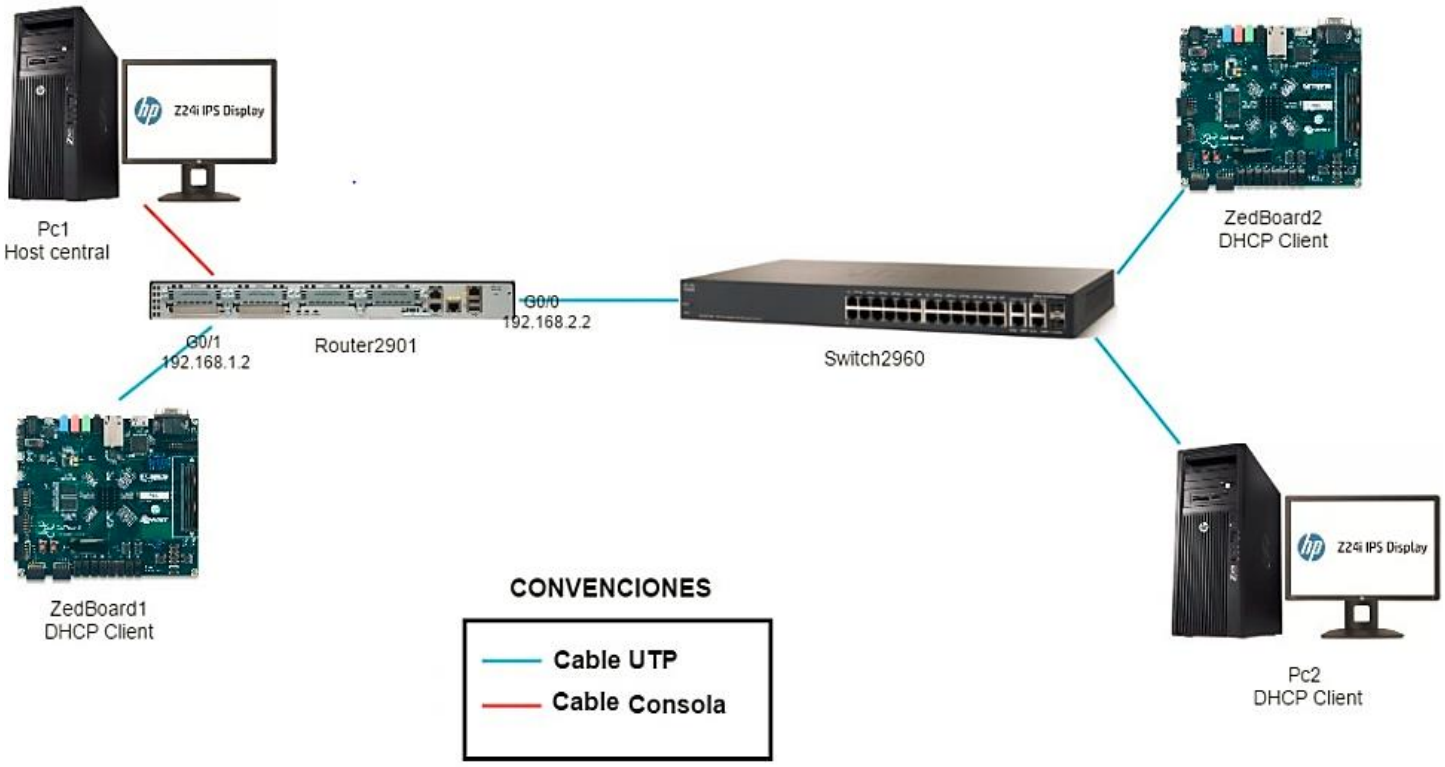


Figura 4. Esquema de topología de red tipo estrella.

El modelo OSI resume la secuencia de pasos que se llevan a cabo cada vez que se comparte datos entre las Zedboard y las estaciones de trabajo a través de la red. Es también relevante anotar que el modelo OSI ha sido empleado para efectuar la validación de conectividad Ethernet.

Desde la capa física (capa 1) del modelo OSI ha sido ejecutada una evaluación visual por medio de los indicadores ***leds*** ubicados en el panel frontal de los equipos activos de la red. Dichos indicadores permiten inferir el estado actual de la interfaz. La figura 5 muestra el ***switch*** empleado durante la práctica.

Figura 5. ***Switch*** CISCO Catalyst 2950 series. (Switch Equipment IIc, 2016).

Por ejemplo, en el caso del ***switch CISCO*** el estado de los ***leds*** se resume en 3 tipos de indicadores: ***led*** del sistema, de suministro remoto de energía (RPS) y de modo de puerto. El ***led*** del sistema advierte si el equipo está recibiendo energía y funcionando correctamente. El ***led*** de suministro remoto de energía indica si la fuente remota de potencia está en uso. Finalmente, el ***led*** de modo puerto incluye diferentes maneras de interpretar los ***leds*** de estado de puerto. En la Tabla 1 se detallan los diferentes indicadores de este ***led*** (Redes locales y globales, s.f.).

Tabla 1

*Resumen de indicador **led** modo puerto.*

LED DE MODO	COLOR	INDICADOR
STAT (Operación del puerto)	Apagado	No hay un dispositivo conectado al puerto.
	Verde sin variación	Enlace activo.
	Verde parpadeando	El puerto está enviando o recibiendo datos.
	Alternando verde/ámbar	El enlace presenta falla.
	Ámbar sin variación	El puerto no envía datos porque está desactivado por el administrador.
UTIL (Porcentaje de ancho de banda)	Apagado	Cada ***led*** apagado indica una reducción por la mitad del ancho de banda total. Los ***leds*** se apagan de derecha a izquierda. Si el ***led*** del extremo derecho está apagado, entonces el ***switch*** usa menos del 50% del ancho de banda total. Si los dos ***leds*** del extremo derecho están apagados, entonces el ***switch*** usa menos del 25% del ancho de banda total.
	Verde	Todos los ***leds*** se hallan verdes, el ***switch*** usa el 50% o más del ancho de banda total.
DUPLX (Modo de transmisión)	Apagado	Opera en modo ***half-duplex***.
	Verde	Opera en modo ***full-duplex***.
SPEED (Velocidad de transmisión)	Apagado	Opera a 10Mbps.
	Verde	Opera a 100Mbps.
	Verde parpadeando	Opera a 1000Mbps.

Nota. La tabla anterior muestra de manera resumida las interpretaciones que se pueden obtener del ***led*** estado de puerto. Adaptada de (Redes locales y globales, s.f.).

Lo anteriormente expuesto permite dar por sentado que la primera capa del protocolo OSI opera correctamente y por consiguiente se puede continuar con la transmisión. En la figura 6 y 7 se muestra la operación de los ***leds*** en un ***switch*** y en un ***router***:

Figura 6. ***Switch*** (nodo central) con las interfaces de red activas.

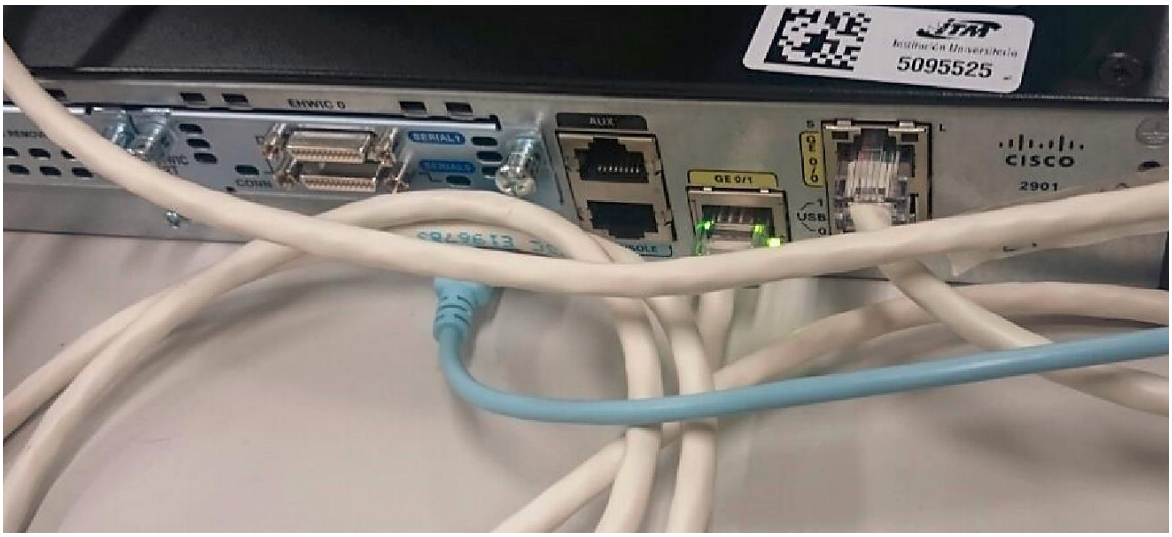


Figura 7. ***Router*** con las interfaces de red activas.

En la capa 1 es necesario validar el correcto funcionamiento del medio de transmisión cableado en este caso los ***patch cord*** de UTP. Se debe verificar que la conexión eléctrica

del cable y el ***plug*** este ajustada correctamente, además validar que se encuentren conectados de manera apropiada a las interfaces de red de los dispositivos.

Ethernet es el protocolo principal que opera en la capa de enlace (capa 2). Este estándar se encarga de determinar las características del medio físico y el formato de la unidad de datos (trama) que opera en dicha capa. Entre las principales funciones consignadas en la capa 2 se pueden mencionar la detección de errores y la topología de red (estrella). A partir de esta capa todo lo define estrictamente el software. Sin embargo, en ésta se sondea el medio (UTP) y se usa el software para interpretarlo y controlar la transmisión: se enlaza la representación física de los datos y su representación lógica (E, s.f.).

En resumen, la aplicación de la capa 2 en el desarrollo expuesto radica en la elección de UTP como medio de transmisión y Ethernet como tecnología empleada.

Partiendo de los supuestos anteriores, si el enlace de datos no se encuentra en operación las capas superiores no trabajarían correctamente. El adecuado funcionamiento de esta capa ha permitido continuar con la transmisión de datos entre los dispositivos.

Posteriormente, integrando la capa de red (capa 3) y transporte (capa 4) ha sido empleado el ***ping*** como herramienta diagnóstica de conectividad Ethernet. El ***ping*** dispone del protocolo ICMP para validar conectividad entre los dispositivos de la red. El protocolo ICMP se encarga de manejar mensajes de error y control para los sistemas de la red informando a la fuente donde se origina, con la finalidad de corregir la falencia hallada (Herramientas web para la enseñanza de protocolos de comunicación , s.f.). El ***ping*** ayuda a identificar el origen de las fallas en una red, va desde la capa 3 a la capa 1 y viceversa. Como tal, la herramienta permite probar el ***stack*** de protocolos, la configuración de direcciones IPV4 en los dispositivos y la conexión a los ***hosts***.

A través del ping han sido ejecutadas las siguientes pruebas:

- Ping desde la terminal del ***host*** central de la red a la misma dirección de dicho equipo. Esto con el fin probar si la interfaz de red se encuentra operando adecuadamente.
- Ping a la dirección 127.0.0.1 (***loopback)***. Con la intención de validar que los protocolos entre la capa 3 hasta la capa1 del modelo OSI operen adecuadamente.
- Ping entre las Zedboard y los hosts pertenecientes al mismo segmento de red. Con el fin de probar si el medio de transmisión del equipo funciona correctamente.
- Ping a la puerta de enlace del ***router***. Con el fin de probar si el cableado general de toda la topología opera apropiadamente.

Finalmente, la capa de presentación (capa 7) tiene contacto directo con los usuarios, su función radica en acondicionar la comunicación entre los usuarios y el envío de datos. Con el objeto de que los usuarios logren interactuar con la red de datos, se han implementado aplicaciones, protocolos y servicios que logran que esta comunicación sea significativa y efectiva. Las aplicaciones son programas con los cuales los usuarios se relacionan y empiezan la transferencia de datos. Los servicios de red también son programas, pero a diferencia de las aplicaciones permiten la conexión entre la capa 7 y las capas anteriores. Los protocolos de red, de manera resumida son una disposición de reglas y pautas para lograr comunicación entre aplicaciones, también procuran que los servicios activos en determinado dispositivo envíen y reciban datos desde otros pertenecientes a redes diferentes. A continuación, se describen los servicios, protocolos y aplicaciones probadas a través de la red Ethernet implementada:

- **Prueba DHCP:** el servicio-protocolo de configuración dinámica de host posibilita a los dispositivos de una red conseguir direcciones IP de un servidor DHCP. DHCP emplea el puerto de red 67/UDP (servidor) y 68/UDP (cliente). A través de dicho servicio se ha efectuado una asignación dinámica de direcciones IP a los ***hosts*** y

tarjetas de desarrollo Zedboard. Con el uso del DHCP ha sido posible ahorrar la configuración manual en cada uno de los dispositivos de la red. El DHCP ha sido empleado pensando en la escalabilidad de la red, es decir, en el crecimiento que podría requerir la misma durante el desarrollo. Gracias al DHCP se logra una administración más eficiente de los dispositivos de la red, en especial cuando estos pueden estar entrando y saliendo de la misma. En la figura 8 se muestra una tarjeta de la red recibiendo direccionamiento desde el servidor DHCP configurado en el ***router***.

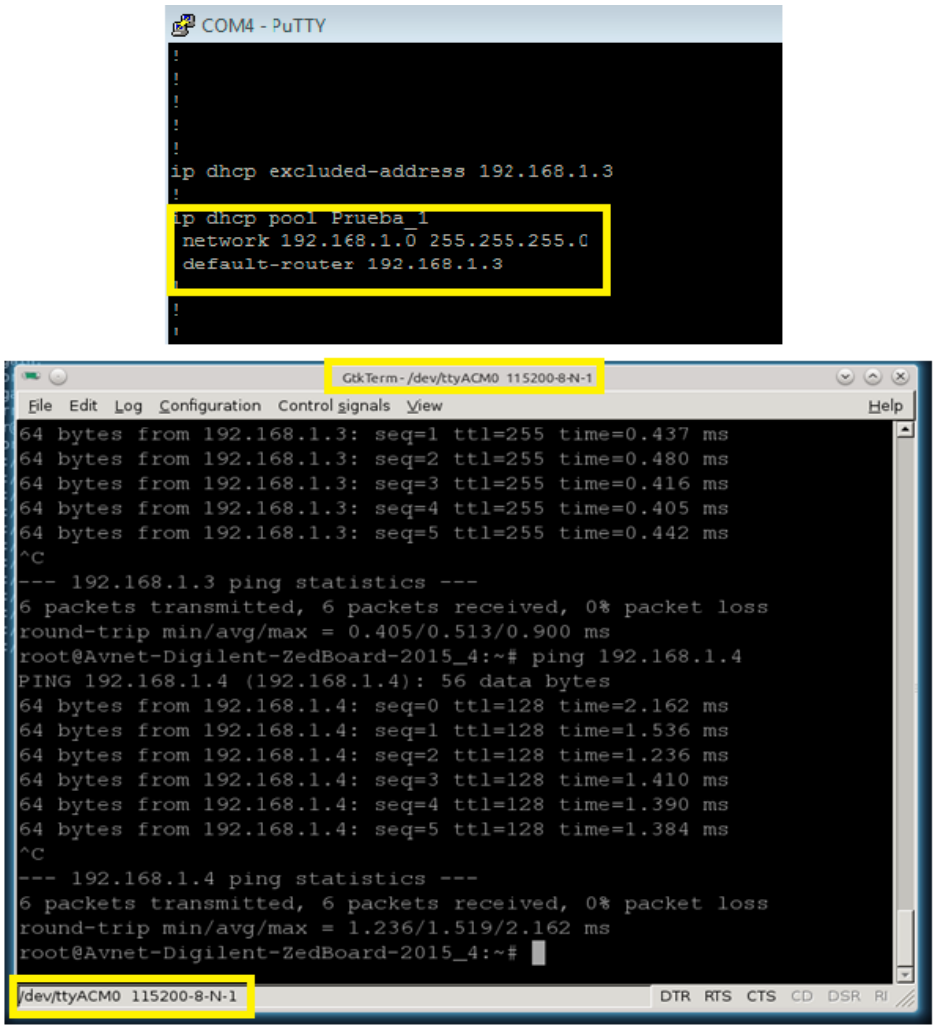


Figura 8. ***Host*** y tarjetas Zedboard recibiendo direccionamiento desde el servidor DHCP.

- **Prueba SSH:** el protocolo y programa interprete de órdenes seguras sirve para acceder vía remota a dispositivos de una red a través de una conexión cifrada. Además, SSH posibilita copiar datos de manera segura. Normalmente el puerto que escucha el servidor es el puerto de red 22. En la topología de red, el servidor SSH ha sido de gran ayuda para acceder desde el ***host*** central a la terminal de cualquiera de las Zedboard inclusive a la del ***router*** y ***hosts*** conectados en la red. SSH optimiza el tiempo de acceso a la tarjeta comparado con el que provee el puerto USB-UART de las mismas. En relación al control, SSH ha permitido manipular de manera remota el estado de algunos periféricos de las tarjetas de desarrollo. En la figura 9 se muestra la comunicación vía SSH desde el ***host*** central hacia una de las Zedboard.

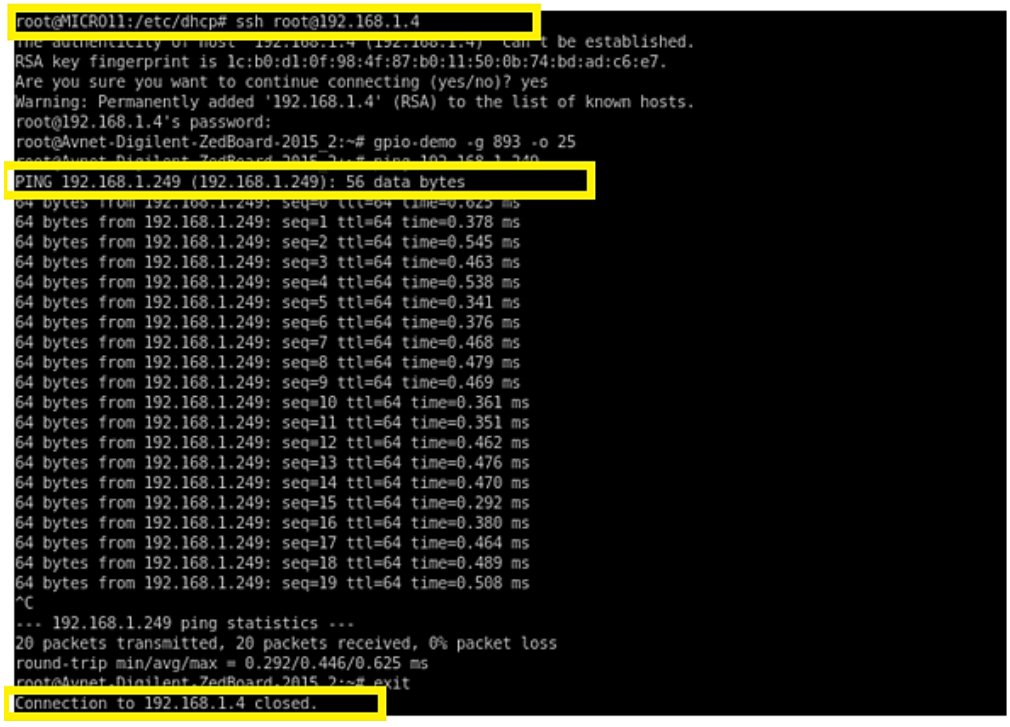


Figura 9. Conexión vía SSH desde el ***Host*** central a una de las tarjetas de la red.

- **Prueba FTP:** El protocolo de transferencia de archivos propicia la transferencia de ficheros entre un cliente y un servidor. FTP emplea el puerto de red 20 (TCP ***DATA Port***) y el 21 (TCP ***CONTROL Port***). Dicho servicio ha sido probado en las tarjetas de desarrollo por medio de Petalinux. Desde la consola del ***host*** central (modo servidor FTP) fueron establecidas sesiones FTP con las Zedboard y otros ordenadores (modo clientes FTP) pertenecientes a la red. FTP juega un papel importante al momento de enviar aplicaciones o utilidades a través de la red. Es uno de los servicios que aporta mayor beneficio en las actividades que se proyecta emprender mediante el producto realizado.

- **Prueba HTTP**: El protocolo de transferencia de hipertexto provee la transferencia de información en WWW. Dicho protocolo establece las pautas y sintaxis que emplean los clientes y servidores para comunicarse. Al escribir una URL en un explorador se inicia una comunicación con el servicio ***web*** que emplea el protocolo HTTP. El protocolo HTTP transmite por el puerto de red 80/ TCP. En la práctica realizada, fue empleado el programa servidor Apache HTTPD (HTTP ***Daemon***). Dicho programa corre de fondo en un servidor ***web*** y espera peticiones de un usuario entrante. Este programa genera respuestas de manera automática, los documentos de hipertexto lo emplean mediante el protocolo HTTP. En efecto, dicho servicio ha sido probado a través de una página ***web*** que trae consigo Petalinux. Por medio del explorador (aplicación de cliente) instalada en el ***host*** central, se escribe la dirección IP asignada a la Zedboard y se obtiene el acceso a dicha página. En la figura 10 se muestra el uso del servicio WEB.

Figura 10. Ingreso a la página ***web*** predeterminada en Petalinux.

De as evidencias anteriores se puede inferir el adecuado funcionamiento de la capa 7 del modelo OSI. Con esto se finaliza la evaluación de la topología implementada. La topología de red establecida cumple con el propósito trazado al inicio del proyecto.

4.2 Prueba con GPIO-demo de Petalinux

Se ha probado el GPIO-demo que trae consigo el petalinux. A pesar de no ser premisa en este desarrollo, a través de dicha rutina se pretende mostrar la funcionalidad y el control remoto que se logra ejercer sobre algunos de los perifericos de las Zedboard activas en la red.

GPIO-demo es una aplicación de línea de comandos empleada para establecer comunicación con los 5 botones pulsadores, 8 interruptores y 8 ***leds***. En la figura 11 Se muestran los periféricos involucrados en el GPIO-demo.

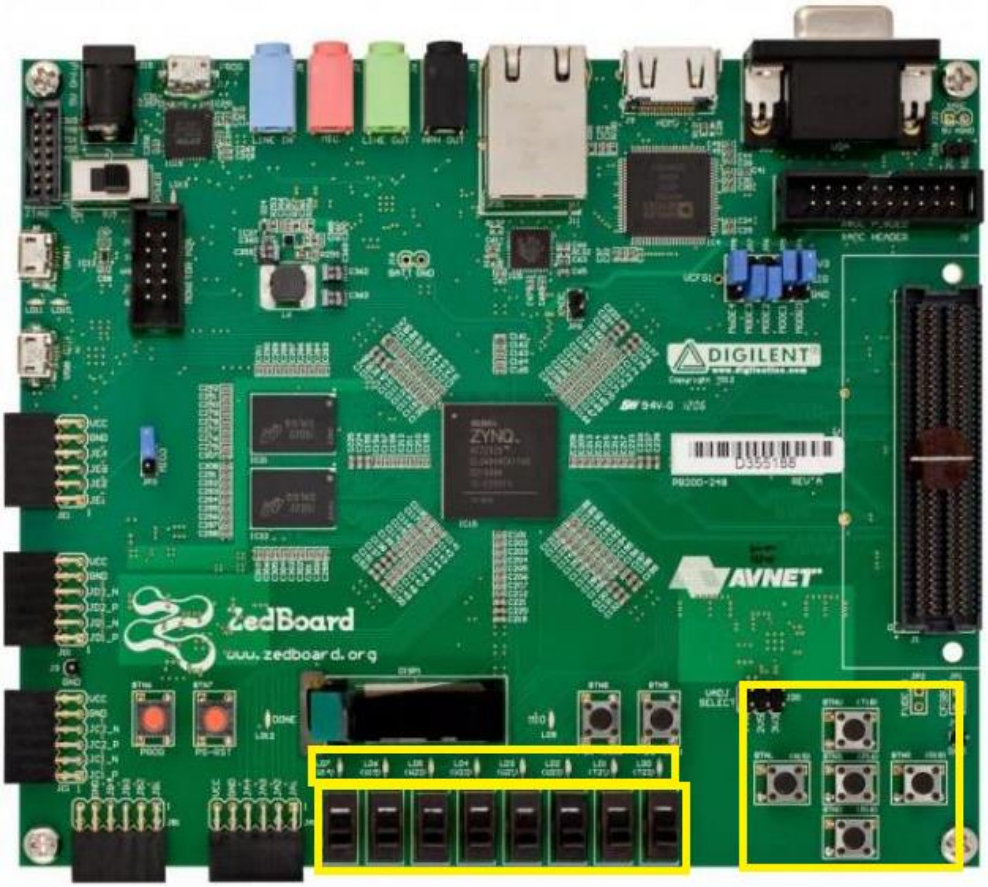


Figura 11. Periféricos probados a través del GPIO-demo. Adaptada de (Xilinx INC, s.f.).

Para llamar el programa, se escribe en la terminal de la Zedboard la siguiente línea de comando *gpio-demo –g <gpio-Id> -o <value>*. El campo <value> hace referencia al valor asociado al puerto, se escribe en notación decimal, pero en binario establece el valor de determinado ***led***. El *<gpio-id>* corresponde al identificador del dispositivo mostrado en la siguiente imagen. La figura 12 indica los identificadores de los periféricos involucrados en el GPIO-demo.

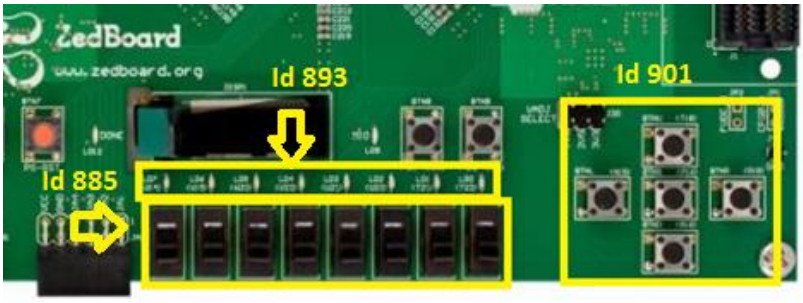


Figura 12. Identificadores de periféricos. Adaptada de (Xilinx INC, s.f.).

- Ejemplo 1: se ha indicado que a los ***leds*** (id 893) sea asignado el valor 5 o 00000101, donde 1 equivale a los ***leds*** encendidos y 0 los ***leds*** apagados. En la figura 13 y 14 se ilustra lo anterior.

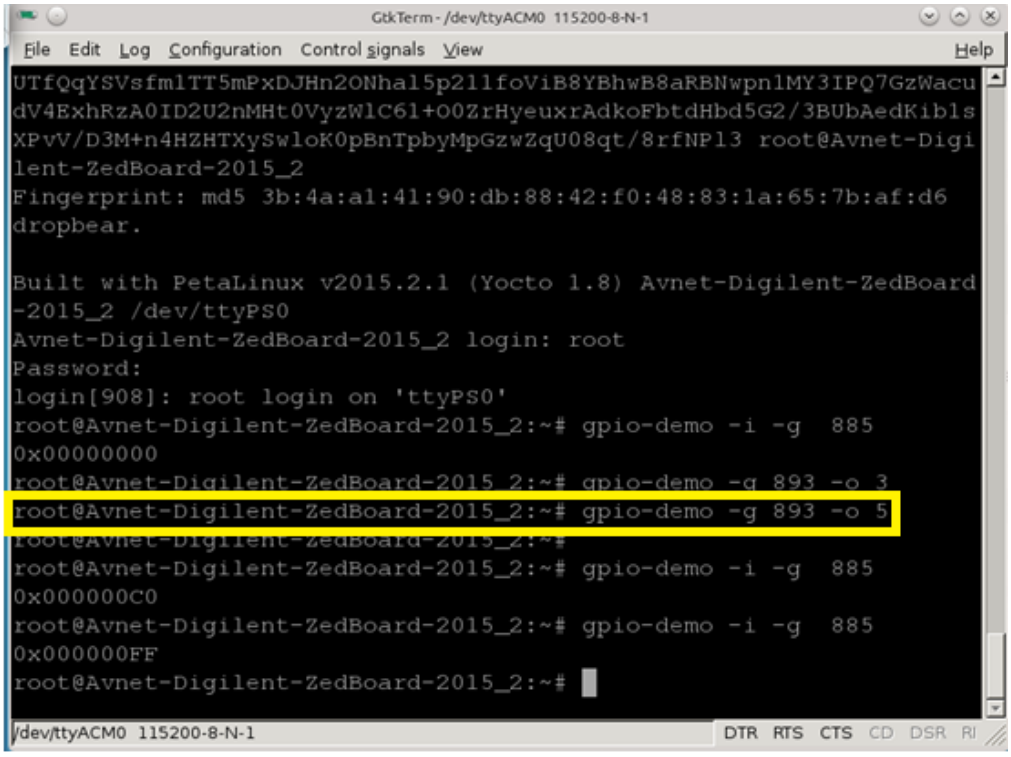


Figura 13. Ejemplo 1 línea de comando GPIO-demo.

Figura 14. Ejemplo 1 ***leds*** encendidos mediante GPIO-demo.

- Ejemplo 2: se ha indicado que a los ***leds*** (id 893) sea asignado el valor 255 o 11111111, donde 1 equivale a los ***leds*** encendidos y 0 los ***leds*** apagados. En la figura 15 y 16 se ilustra lo anterior.

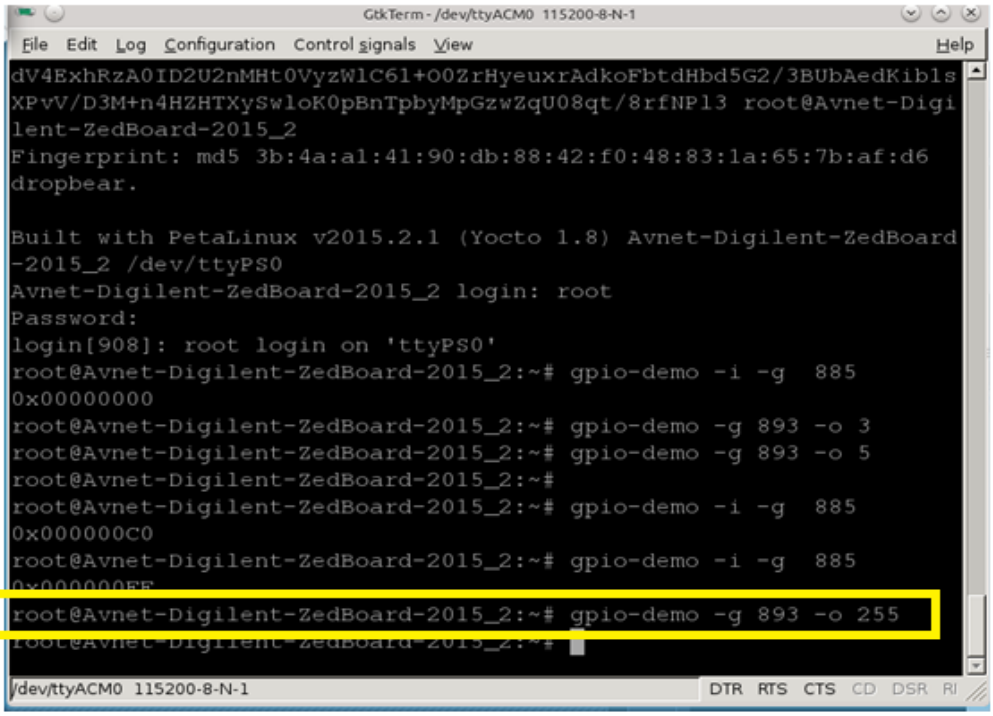


Figura 15. Ejemplo 2 líneas de comando GPIO-demo.

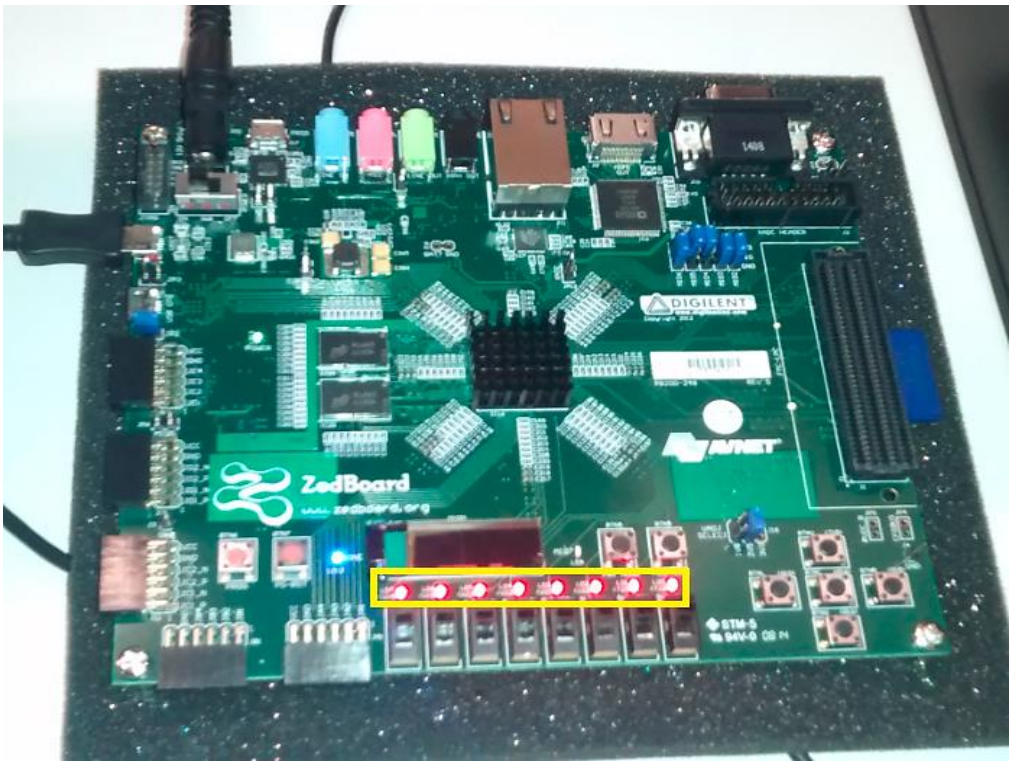


Figura 16. Ejemplo 2 ***leds*** encendidos mediante GPIO-demo.

Otra prueba efectuada, consiste en acceder desde el ***host*** central vía SSH a la terminal de la tarjeta que se deseaba intervenir. Por consiguiente, con la utilización del servidor SSH se hizo posible gestionar los periféricos de todas las Zedboard pertenecientes a la red. En la figura 17 se ilustra lo anterior.

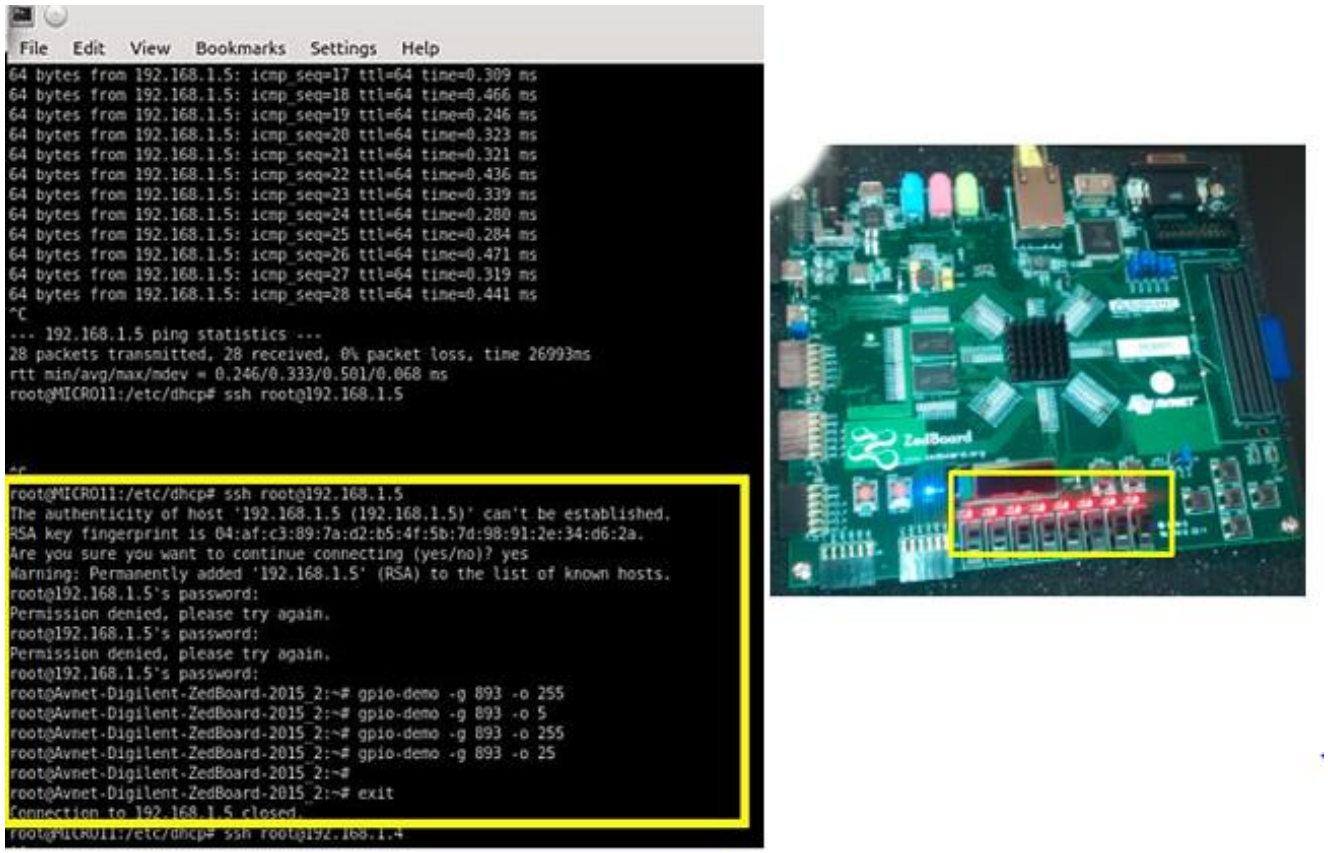


Figura 17. GPIO-demo vía SSH desde el ***host*** central a una de las Zedboard de las dos LAN implementadas en la red.

GPIO-demo ha sido probado con el fin de demostrar que dentro la red Ethernet es posible gestionar los periféricos de las tarjetas dispuestas en las diferentes LAN de la topología. Cabe resaltar, que esta tarea ahorra el trabajo manual de manipular cada tarjeta.

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

El pilar de este desarrollo radica en la necesidad de conectividad entre las tarjetas de desarrollo, el cual se logra a través de la tecnología Ethernet. Mediante el uso de distintos servicios, protocolos, herramientas y aplicaciones de red ha sido posible explorar, identificar y detallar ventajas que otorga este modo de comunicación.

La versión embebida del sistema operativo favorece la interacción, aprovechamiento y control de los componentes de las tarjetas. Al mismo tiempo, ha evitado la complejidad de emplear un lenguaje de bajo nivel. La compilación, instalación, configuración y funcionalidad de Petalinux se ha convertido en la base para lograr la configuración de la red de datos entre Zedboard.

Con la implementación de una topología de red tipo estrella se ha establecido conexión entre diferentes equipos activos de la red como Zedboard, ***switch***, ***router y hosts***. La mencionada configuración ofrece diversas ventajas como facilitar la comunicación mediante el uso del nodo central (***switch***), permitir cambios a nivel aplicativo sin afectar el cableado físico, favorecer el proceso de conexión y reconexión de dispositivos. Además, dicha estructura es fácilmente escalable para nuevos dispositivos, lo cual la hace llamativa para trabajos posteriores.

Ethernet es la tecnología empleada para comunicar las tarjetas Zedboard. Se realizó el análisis y prueba de compatibilidad de los sistemas embebidos con dicha tecnología al momento de establecer conectividad. Este postulado es a fin con la visión de la IoT donde prima la necesidad de conectar estos aparatos a Internet. Los resultados obtenidos

posibilitan validar el adecuado funcionamiento de la comunicación entre todos los dispositivos dispuestos bajo la estructura estrella. La validación de conectividad se ha efectuado mediante el modelo OSI. Las diferentes capas arrojaron resultados positivos, lo cual permite concluir que la transmisión de datos entre los dispositivos involucrados en la red ha sido efectiva y cumple con el propósito inicial de identificar una tecnología apropiada y compatible con las tarjetas de desarrollo.

Petalinux provee el acceso a la pila de servicios y protocolos constituidos en el sistema. El uso de los servicios, protocolos y aplicaciones han sido herramientas excepcionales, ya que posibilitan el intercambio de datos entre todos los dispositivos de la red. DHCP, SSH, FTP y HTTP permiten la interacción de todos los dispositivos activos en red de manera significativa y efectiva.

La existencia del SoC programable de la familia Zynq-7000 de Xilinx basado en FPGA disponible en el laboratorio de Sistemas de Control y Robótica se considera una fortaleza. Además, las investigaciones sobre las redes Ethernet con FPGAs y la documentación del fabricante sirvieron como estudios base, aportando de manera significativa durante la ejecución del producto.

No obstante, surgen diferentes limitaciones relacionadas con diversos factores como:

- Espacio: la disponibilidad de los laboratorios fuera de los tiempos establecidos para el público, ya que en repetidas ocasiones es necesario efectuar pruebas con los equipos y dispositivos pertenecientes a dicho sitio.
- Conocimiento: la carencia de conocimiento sobre las tarjetas de desarrollo ya que desde el área de telecomunicaciones no se profundiza sobre el tema. Para esta actividad es necesario investigar y adquirir los conceptos claves. Lo cual implica mayor tiempo de ejecución del producto.

- Herramientas (programas): diferentes sitios ***web*** explican cómo instalar Petalinux desde el sistema operativo Ubuntu 14.04.4, se hicieron pruebas en diferentes ocasiones sin obtener una correcta instalación. Kubuntu 14.04 es el sistema operativo con el que se ha logrado la instalación de Petalinux. Por otra parte, El uso de Vivado 2014.4 versión libre presenta algunas dificultades de compatibilidad durante la instalación de Petalinux, por esta razón es necesario emplear una versión licenciada como las que ofrece la institución. Otra de las dificultades presentadas durante la ejecución del proyecto ha sido la escasa experiencia sobre la configuración de hardware. Pero, este hecho se pudo solventar gracias al conocimiento en ***networking*** y a las capacidades del sistema operativo integrado.

Desde la perspectiva didáctica, el proyecto es útil para el uso de sistemas basados en FPGA en red y la implementación de diferentes tipos de algoritmos. Sin embargo, se recomienda investigar o tener mayor conocimiento sobre el lenguaje de *bash script* con la finalidad de aprovechar al máximo su funcionalidad y poder crear aplicaciones como el *GPIO-demo* incorporado en Petalinux. De la misma manera, es necesario profundizar sobre Linux como plataforma y las utilidades que ofrecen los comandos empleados en la terminal, debido a que por desconocimiento o falta de pericia se omiten procesos que pueden incurrir en errores. Sería interesante emplear la conexión en red no solo en un ambiente de laboratorio, sino que el presente desarrollo sea integrable en entornos reales.

Como futuros trabajos es posible implementar una topología de red tipo estrella empleando IPV6 y comparar su funcionamiento con IPV4, con el fin de identificar cuál de los dos protocolos genera mayores beneficios para los sistemas de desarrollo. Es factible integrar a la Zedboard un módulo WIFI que permita analizar el comportamiento de los sistemas embebidos conectados en red bajo este tipo de comunicación. De igual forma, las Zedboard se pueden utilizar como un dispositivo de recepción VoIP, usando los puertos

de audio de entrada y con ayuda de un micrófono es viable disponer de éstas como una estación de llamadas. Otra utilidad sería la implementación de un analizador de tráfico de red con las Zedboard.

El producto en mención también se puede aprovechar con la finalidad de intercambiar datos entre sistemas de desarrollo para tareas de multiprocesamiento. Además, sería interesante hacer herramientas en *bash script* e incluirlas en el núcleo de Petalinux y así aumentar las capacidades de la Zedboard.

REFERENCIAS

Alonso Arévalo, J. (Enero de 2016). El “Internet de las cosas...”. págs. 24-26. Obtenido de http://hdl.handle.net/10366/127354

CCM.net. (s.f.). Recuperado el 2016, de http://es.ccm.net/contents/367-tarjetas-de-red

Cisgroupla. (2015). *Cisgroupla*. Resumen del internet de las cosas. [Figura1] Obtenido de http://www.cisgroupla.com/wp/wp-content/uploads/2015/12/Internet-de-las-cosas-01.jpg

Clúster ICT - Audiovisual de Madrid. (Marzo de 2013). *Madridnetowrk.org*. Obtenido de https://actualidad.madridnetwork.org/imgArticulos/Documentos/635294387380363206.pdf.

E, C. A. (s.f.). *Informática ++*. Obtenido de http://cesarcabrera.info/blog/%C2%BFen-que-consiste-la-capa-2-del-modelo-osi-enlace-de-datos/

Embedded Centric. (s.f.). Obtenido de https://embeddedcentric.com/embedded-operating-systems

Embedded Centric. (s.f.). Obtenido de https://embeddedcentric.com/networked-systems/

Gad, V., Gad, R., & Naik, G. (Agosto de 2012). Implementation of gigabit ethernet standard using FPGA. *International Journal of Mobile Network Communications & Telematics (IJMNCT), 2*(4), 1-14.

globales, R. l. (s.f.). Obtenido de https://sites.google.com/site/redeslocalesyglobales/2-aspectos-fisicos/5-dispositivos-de-interconexion-de-redes/4-conmutadores-switch/3-indicadores-led-en-un-conmutador

González, M. S. (08 de 11 de 2013). *Redes Telemáticas*. Obtenido de http://redestelematicas.com/el-switch-como-funciona-y-sus-principales-caracteristicas/

Herramientas web para la enseñanza de protocolos de comunicación. (s.f.). Obtenido de http://neo.lcc.uma.es/evirtual/cdd/tutorial/red/icmp.html

Neri de Souza, R., Nascimento Muniz, D., & Vaz da Silva Fidalgo, A. (2011). Ethernet Communication Platform for synthesized devices in Xilinx FPGA. *EUROCON - International Conference on Computer as a Tool (EUROCON) 2011 IEEE*, 1-4. Obtenido de http://ieeexplore.ieee.org/document/5929377/

R Perrett, M., & Darwazeh, I. (2011). A Simple Ethernet Stack Implementation in VHDL to Enable FPGA logic reconfigurability. *2011 International Conference on Reconfigurable Computing and FPGAs*, 286-290. Obtenido de http://ieeexplore.ieee.org/document/6128591/

Redes locales y globales. (s.f.). Obtenido de https://sites.google.com/site/redeslocalesyglobales/2-aspectos-fisicos/5-dispositivos-de-interconexion-de-redes/4-conmutadores-switch/3-indicadores-led-en-un-conmutador

Switch Equipment IIc. (2016). Switch CISCO Catalyst 2950 series. [Figura5] Obtenido de http://www.switchequip.com/network-equipment/2950-series/cisco-c2950g-p-610.html?osCsid=abua7jf6ohipil6n5g76pam4f4

Xilinx INC. (s.f.). *Xilinx, All programmable*. Obtenido de http://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html

Xilinx INC. (s.f.). *Xilinx, All programmable*. Obtenido de https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/2015-4.html

Xilinx INC. (s.f.). *Zedboard.org*. Obtenido de http://zedboard.org/product/zedboard

Xilinx INC. (s.f.). *Zedboard.org*. Obtenido de http://zedboard.org/sites/default/files/documentations/GS-AES-Z7EV-7Z020-G-V7.pdf

Xilinx INC. (s.f.). *Zedboard.org*. Placa de desarrollo Zedboard y sus componentes. [Figura3] Adaptada de http://zedboard.org/sites/default/files/product_spec_images/Front-image-of-board_0.jpg

Xilinx INC. (s.f.). *Zedboard.org*. Periféricos probados a través del GPIO-demo. [Figura11] Adaptado de http://zedboard.org/sites/default/files/product_spec_images/ZedBoard_RevA_sideA_0_0%20%281%29_0.jpg

Xilinx INC. (s.f.). *Zedboard.org*. Identificadores de periféricos. [Figura12] Adaptado de http://zedboard.org/sites/default/files/product_spec_images/ZedBoard_RevA_sideA_0_0%20%281%29_0.jpg

APÉNDICE

GUÍA DE INSTALACIÓN DE PETALINUX

1. RECURSOS REQUERIDOS

- Equipo de cómputo 64bits
- Kubuntu 14.04
- Vivado 2014.4 (debe incluir SDK)
- Avnet Diligent Zedboard 2015.4
- Petalinux 2015.4
- Tarjeta Zedboard con FPGA de la familia Zynq-7000 de Xilinx
- Cable USB micro
- Fuente de alimentación de 12V
- Tarjeta SD de 4GB

2. PROCEDIMIENTO DE LA PRÁCTICA

El ser humano se encuentra rodeado de una gran cantidad de dispositivos o artefactos que funcionan gracias a los sistemas embebidos.

Los sistemas operativos integrados son sistemas diseñados para funcionar en plataformas con recursos limitados de hardware (memoria pequeña, de baja potencia, capacidades computacionales limitados etc).

¿Cuándo es necesario utilizar un sistema operativo incorporado? La respuesta a esta pregunta se relaciona directamente con los requisitos del sistema embebido en fase de desarrollo. En términos generales, se adopta un sistema operativo embebido si el sistema

que se diseñará requiere alto nivel de complejidad, aplica la multitarea, la creación de redes, la portabilidad o la escalabilidad.

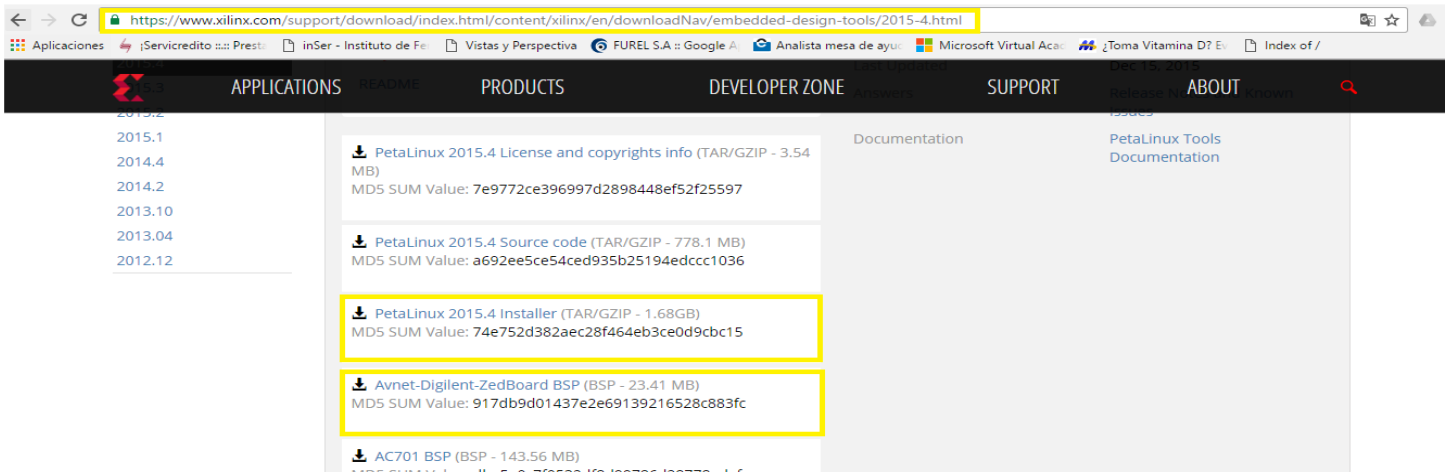
En esta guía se explica el paso a paso de la compilación, instalación, configuración y funcionamiento de un sistema operativo embebido compatible con los chips Zynq. Petalinux es la versión empleada como sistema operativo embebido. Es aconsejable utilizar Petalinux porque es un sistema operativo que está documentado, probado a fondo y Xilinx (fabricante de las Zedboard) proporciona herramientas para un desarrollo llave en mano entre la tarjeta y el sistema operativo.

El sistema Linux es configurado, construido desde la terminal de un ***host*** y se inicia desde la Zedboard usando la tarjeta SD. Los comandos de Linux y aplicaciones de demostración se ponen a prueba en la ***board***.

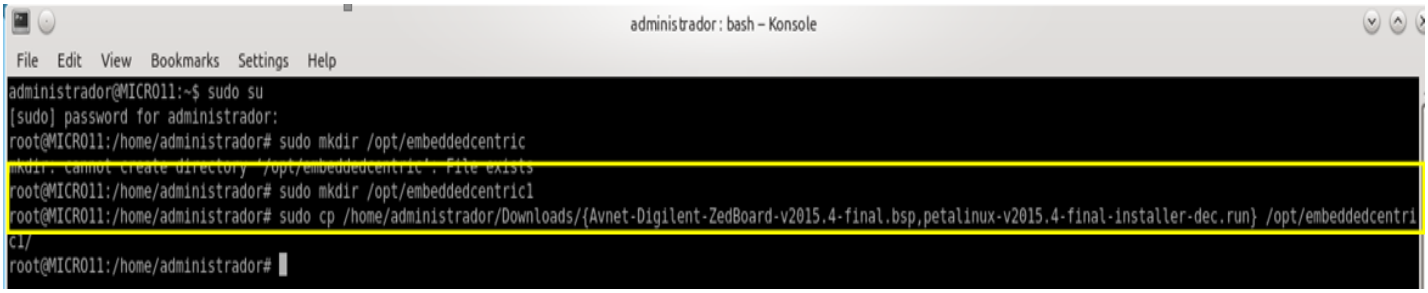
2.1 PREPARACIÓN DE LA ESTACIÓN DE TRABAJO (ANFITRIÓN)

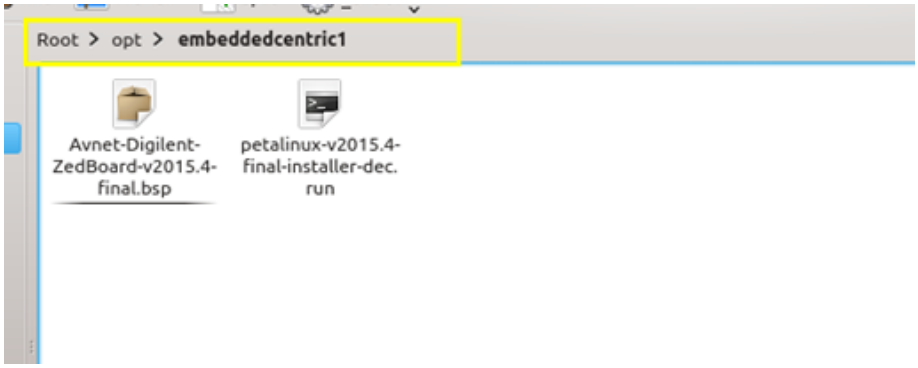
Se deben descargar del sitio ***web*** del fabricante el instalador de Petalinux 2015.4 y los paquetes de la placa Zedboard (Avnet Diligent Zedboard 2015.4):

https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/2015-4.html (Xilinx INC, s.f.).



En el directorio */opt* normalmente se alojan paquetes de aplicaciones de terceros o suplementarios. En este paso se crea una un nuevo directorio en la siguiente ruta: */opt/embeddedcentric1* y se copian los dos archivos descargados en el paso anterior. Este directorio contiene las herramientas de configuración de Petalinux, los paquetes de la placa Zedboard, el código fuente del ***Kernel*** y la imagen ISO generada.

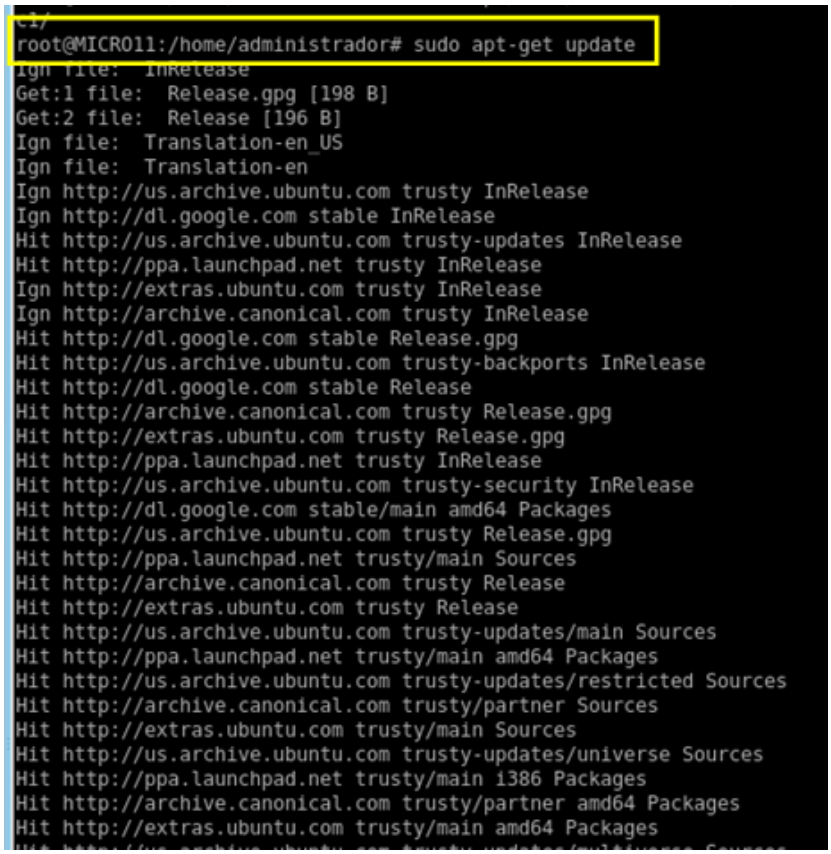


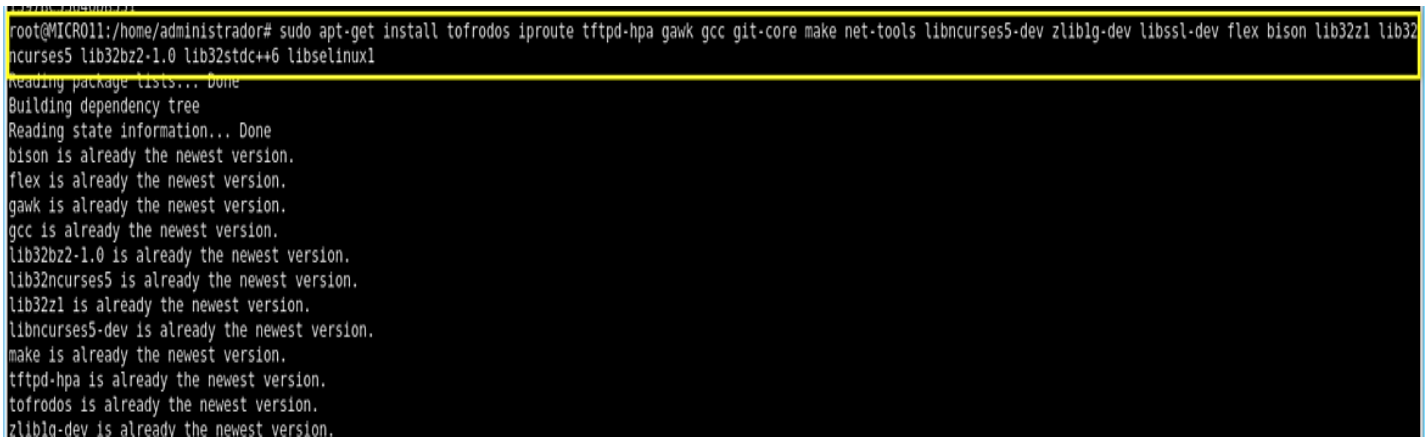


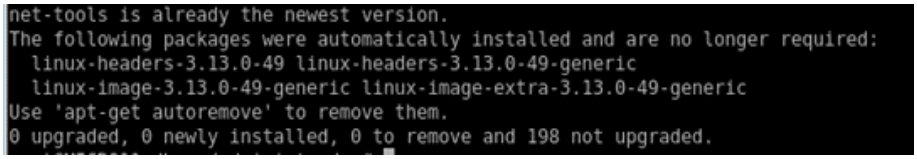
Antes de iniciar la creación de la imagen se debe tener la última versión de todos los paquetes de Kubuntu. Por otra parte, es necesario instalar herramientas y blibliotecas requeridas para el correcto funcionamiento de Petalinux como tofrodos, iproute, tftpd-hpa, gawk y gcc git-core y algunas herramientas de red como libncurses5-dev-dev zlib1g libssl-dev flexión bisontes lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc ++ 6 libselinux1 (Embedded Centric, s.f.). Cabe aclarar que dichos procesos tardan un poco y no se pueden detener. Digitar los siguientes comandos:

sudo apt-get update

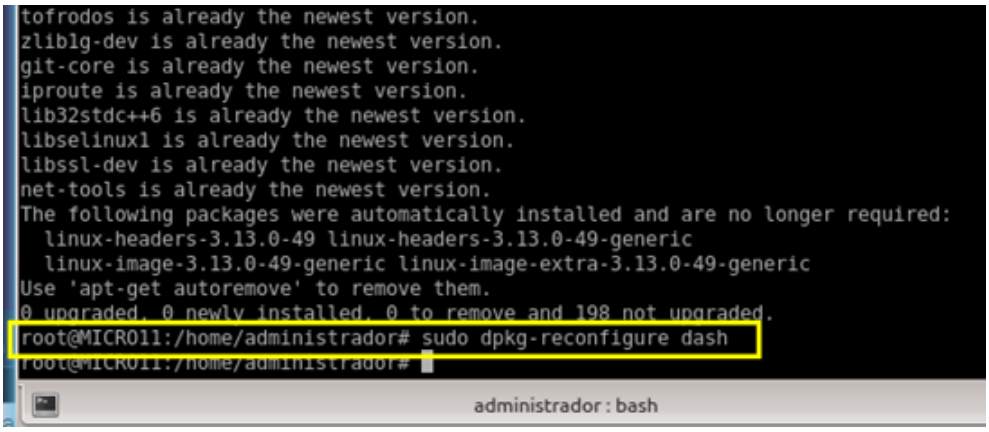
sudo apt-get install tofrodos iproute tftpd-hpa gawk gcc git-core make net-tools libncurses5-dev zlib1g-dev libssl-dev flex bison lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc++6 libselinux1

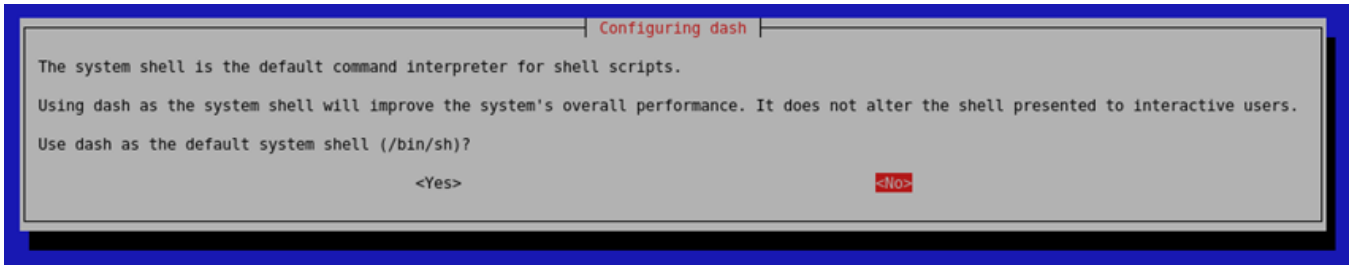




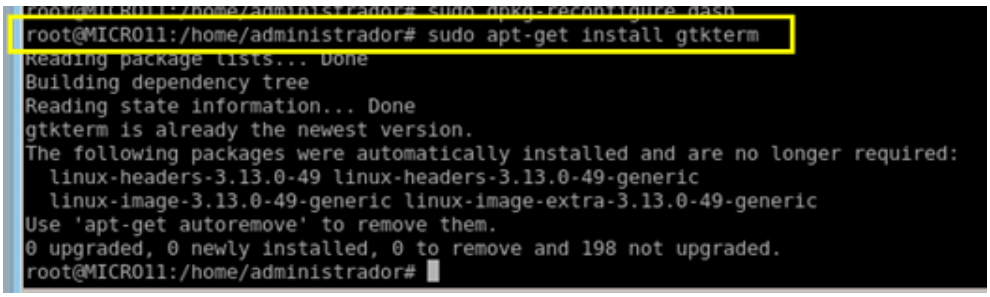


Petalinux requiere emplear *bash* como ***shell*** en reemplazo de *dash* (***shell*** por defecto de la estación de trabajo). Muchas aplicaciones de Petalinux son configuradas con el interpretador de órdenes *bash.* Por tanto, se emplea *bash* con la intención de no generar conflictos. Para cambiar entre *dash* y *bash* basta con introducir el siguiente comando: *sudo dpkg-reconfigure dash.*

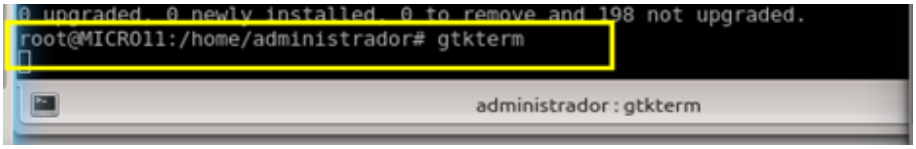


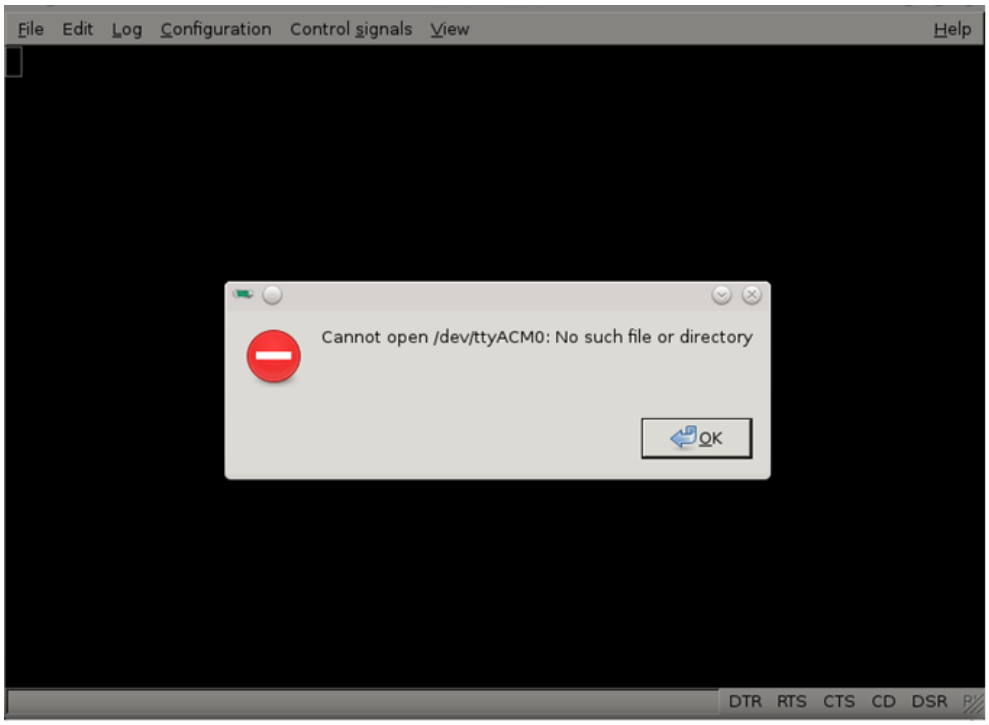


También es requisito instalar la aplicación *gtkterm* que corresponde a la terminal para acceder a la Zedboard através del puerto serie cuando ya se tenga la imagen del sistema operativo lista. *Gtkterm* es el acceso a la Zedboard una vez el ***Kernel*** de Petalinux inicie operación (Embedded Centric, s.f.). Escribir el siguiente comando: *sudo apt-get install gtkterm*

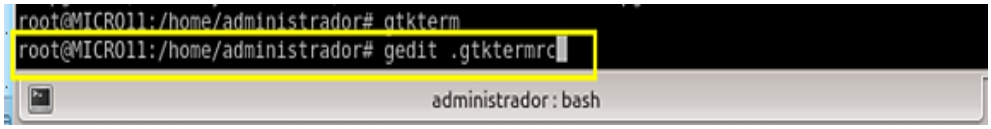


Inicialmente, aparece un pantallazo en el cual se evidencia que no es posible acceder al puerto.

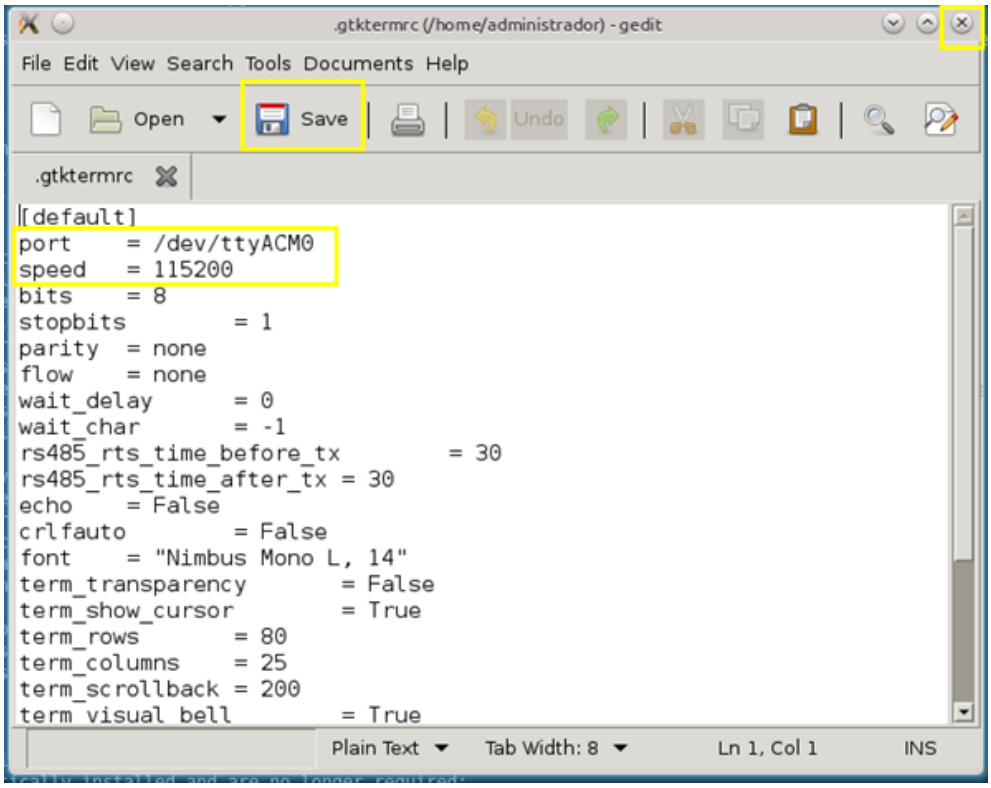


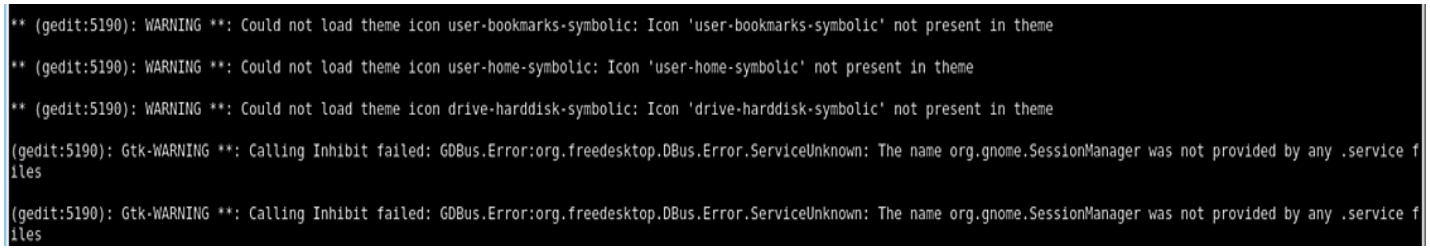


Posteriormente, se configura el puerto serie *gtkterm* de manera que corresponda a las características de la tarjeta Zedboard. Para cambiar los ajustes, se emplea *gedit* (editor de texto de Kubuntu). Ejecutar el siguiente comando: *gedit .gtktermrc.*



Las configuraciones que muestra el editor se dejan tal cual, solo se modifica el puerto y la velocidad. Debe quedar como el siguiente pantallazo:



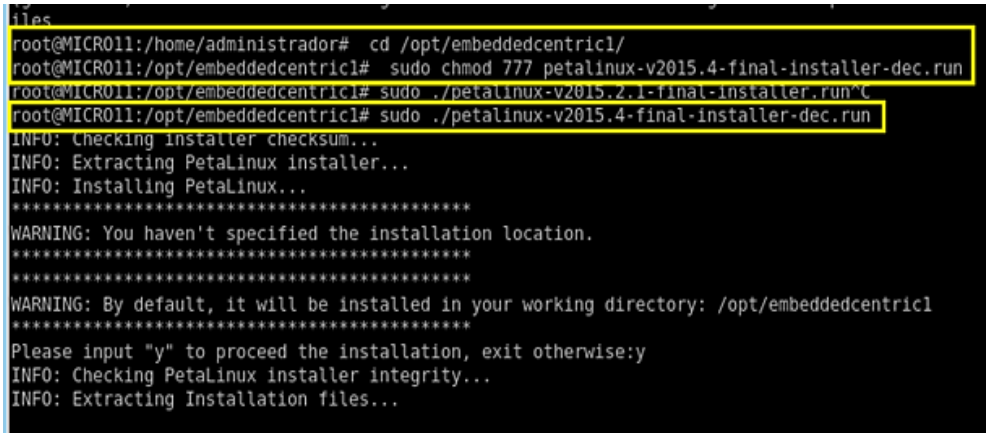


2.2 CORRER EL INSTALADOR DE PETALINUX

Desde el directorio */op/embeddedcentric1* se ejecuta el instalador de Petalinux, esto con el fin de extraer el código fuente del ***Kernel*** y toda su cadena de herramientas (Embedded Centric, s.f.). Entre dicho lineamiento, cabe señalar la importancia de dar los permisos de ejecución sobre el instalador e iniciarlo con los comandos:

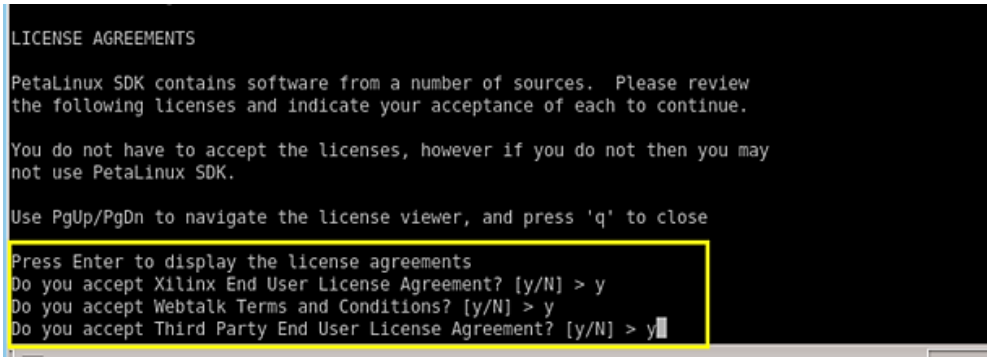
sudo chmod 777 petalinux-v2015.4-final-installer-dec.run

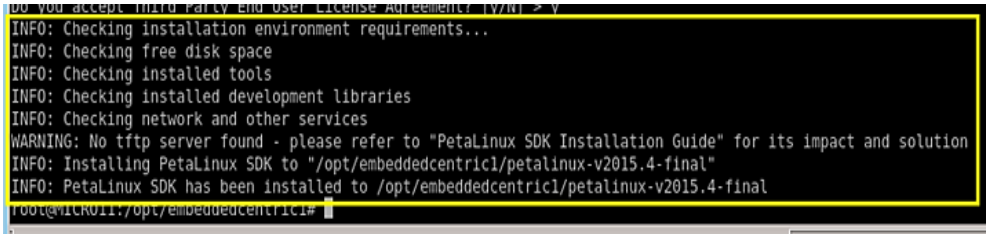
sudo ./petalinux-v2015.4-final-installer-dec.run



Tener presente que el instalador tarda un poco. Durante la ejecución del instalador aparece una ventana donde se solicita la confirmación del proceso de instalación. Para continuar con la extracción se debe pulsar *y* y después ***Enter***.

Seguidamente, aparecen mensajes relacionados con los acuerdos de licencia. Se pulsa *Aceptar* y ***Enter*** para visualizar dicho acuerdo. Leer el acuerdo es opcional, de lo contrario se pulsa *q* para salir del documento y volver al instalador. Desde el instalador, aceptar los acuerdos de licencia de usuario presionando la tecla *y* y ***Enter*** (Embedded Centric, s.f.)**.** Finalmente, la instalación confirma que fue instalado el SDK (Kit de Desarrollo de Software) de Petalinux.

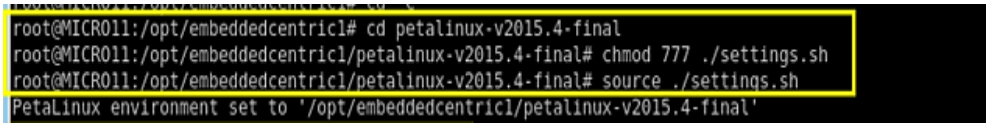




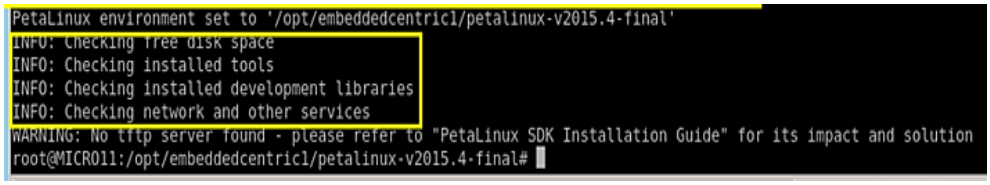
Por otra parte, desde la ruta donde el instalador extrajo los archivos, en este caso *cd /opt/embeddedcentric1/petalinux-v2015.4-final*, se deben configurar las variables de entorno y rutas de búsqueda para las herramientas de Petalinux. Iniciar sesión como súper usuario (*sudo su*) y digitar los siguientes comandos: *chmod 777 ./settings.sh* y posteriormente *source ./settings.sh,* con el fin de hacer ejecutable dicho archivo. Cabe aclarar que este paso debe ser repetido en cada sesión de desarrollo (Embedded Centric, s.f.).

chmod 777 ./settings.sh

source ./settings.sh



En una correcta instalación se debe obtener lo siguiente, lo cual indica que las herramientas de Petalinux se encuentran en listas para su uso.

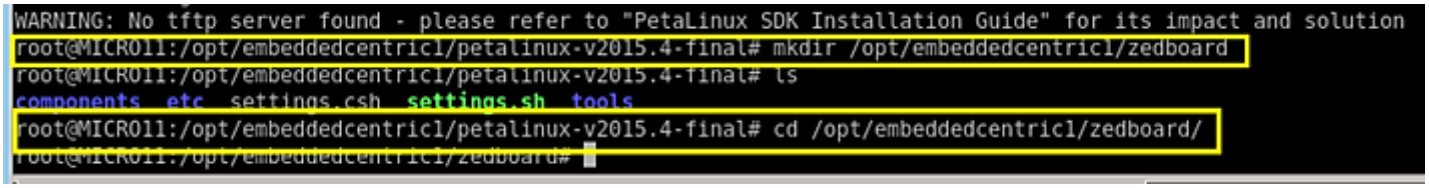


2.3 CREACIÓN DE NUEVO PROYECTO DE PETALINUX

La creación de un nuevo proyecto de Petalinux se efectúa con el proposito de instalar el BSP de la Zedboard.

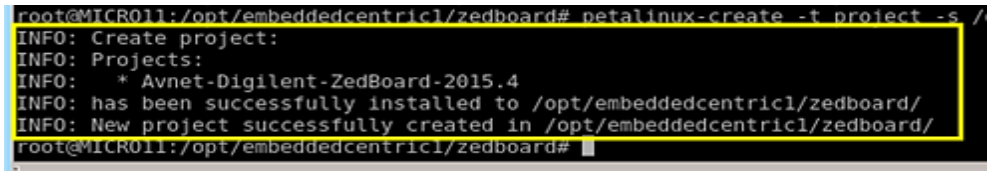
El BSP de la Zedboard es un diseño de referencia que sirve como base para crear nuevos proyectos. Inclusive, el BSP posibilita al diseñador centrarse en sus nuevas aplicaciones ahorrando tareas de configuración de bajo nivel.

Inicar sección como como súper usuario (*sudo su*). Se crea una un nuevo directorio en la siguiente ruta: */opt/embeddedcentric1/zedboard* y se cambia de directorio.



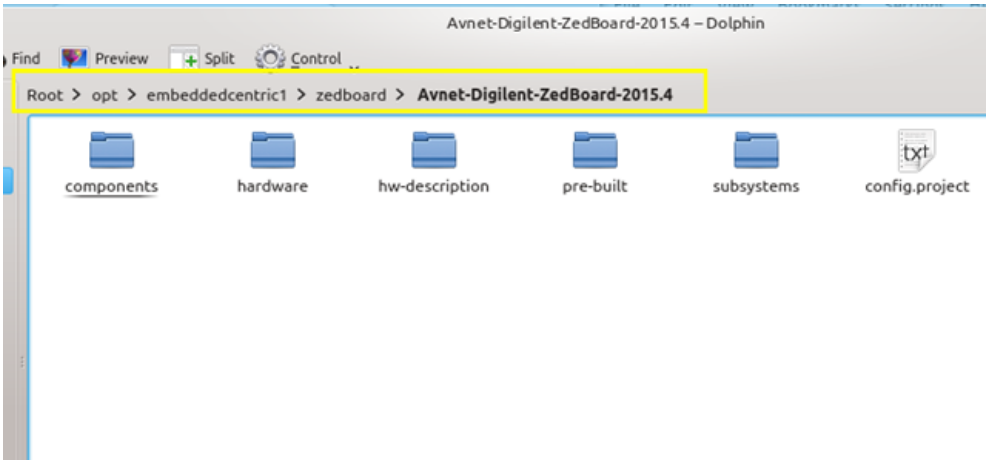
Ahora bien, se crea un nuevo proyecto con referencia a los paquetes de la placa Zedboard (Avnet Diligent Zedboard 2015.4) descargados del sitio del fabricante. Digitar el siguiente comando: *petalinux-create -t project -s /opt/embeddedcentric1/Avnet-Digilent-ZedBoard-v2015.4-final.bsp*

En una correcta instalación se debe obtener lo siguiente, lo cual indica que se pudo crear un nuevo proyecto en directorio Zedboard.



En el directorio donde quedaron almacenados los archivos generados durante la instalación, se debe hallar la siguiente estructura. Es de resaltar que esta distribución se obtiene al utilizar el BSP (Embedded Centric, s.f.).

- **Components:** contiene aplicaciones, librerías, módulos del ***kernel***, ***drivers*** de dispostivos Xilinx y códigos genéricos.
- **Hardware:** contiene Vivado y proyectos SDK para la base del diseño
- **Hw-description:** contiene metadatos para el hardware
- **Pre-built:** contiene compiladores, ejecutables y flujo de datos.
- **Subsystems:** contiene archivos de configuración para los diferentes componentes del sistema.



2.4 CONFIGURACIÓN DE LOS ARCHIVOS RAÍZ DEL SISTEMA

Esta acción se realiza con el fin de configurar los componentes del sistema de archivo tales como herramientas de utilidad, librerías, aplicaciones entre otros. Además de adicionar la aplicación GPIO-demo que viene por defecto con el Petalinux. GPIO-demo ha sido ejecutada con el fin de probar la funcionalidad de los ***leds***, pulsadores e interruptores contenidos en la placa de desarrollo.

Iniciar sección como súper usuario (*sudo su*). Ir al directorio */opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-v2015.4/.* Probar si las herramientas de Petatalinux aún tienen su origen en la sección actual con el siguiente comando: *echo $ PETALINUX.*

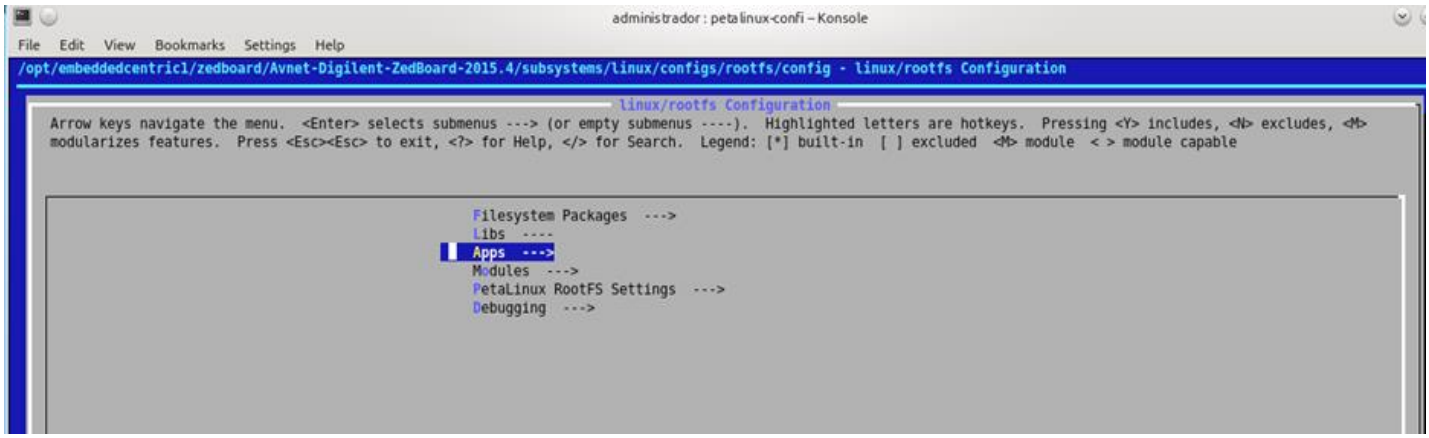
```
Avnet-Digilent-ZedBoard-2015.4: command not found
root@MICRO11:/opt/embeddedcentric1/zedboard# cd Avnet-Digilent-ZedBoard-2015.4
root@MICRO11:/opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4# echo $PETALINUX
/opt/embeddedcentric1/petalinux-v2015.4-final
root@MICRO11:/opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4#
```

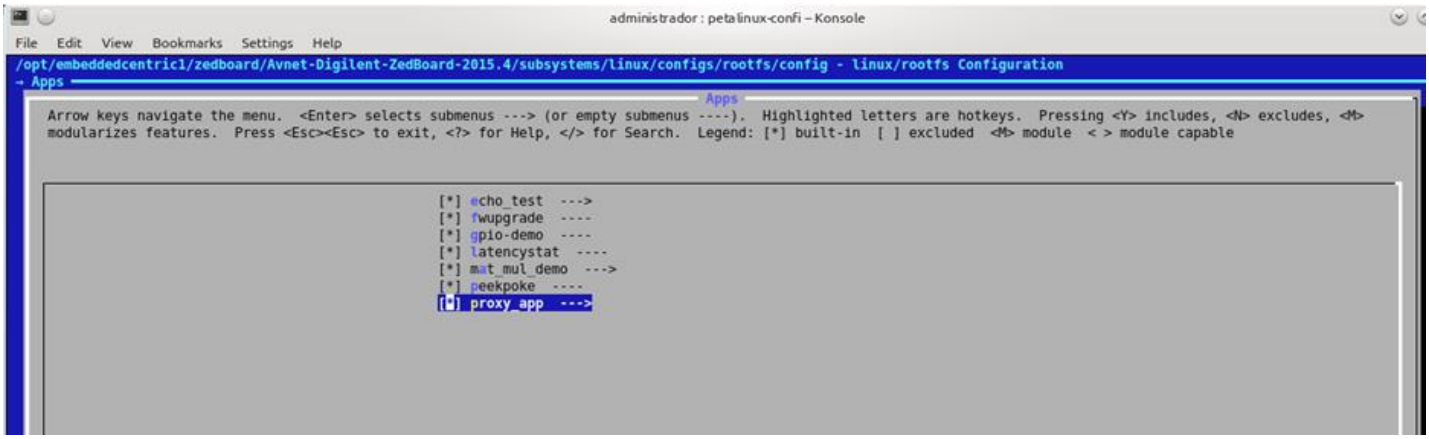
Si el comando anterior no arroja la ruta correcta donde fueron instaladas las herramientas de Petalinux, se debe regresar a la *sección 2.2* donde se configura la fuente de las herramientas (settings.sh*).*

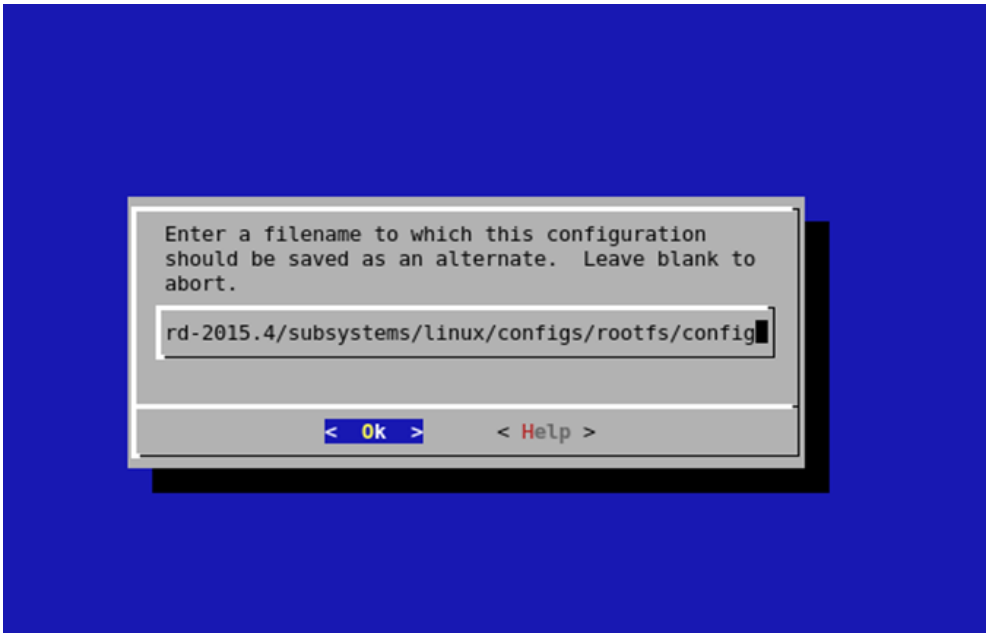
Después, se inicia el menú de configuración del sistema de archivos raíz de Petalinux. Digitar el siguiente comando: *petalinux-config -c rootfs.*

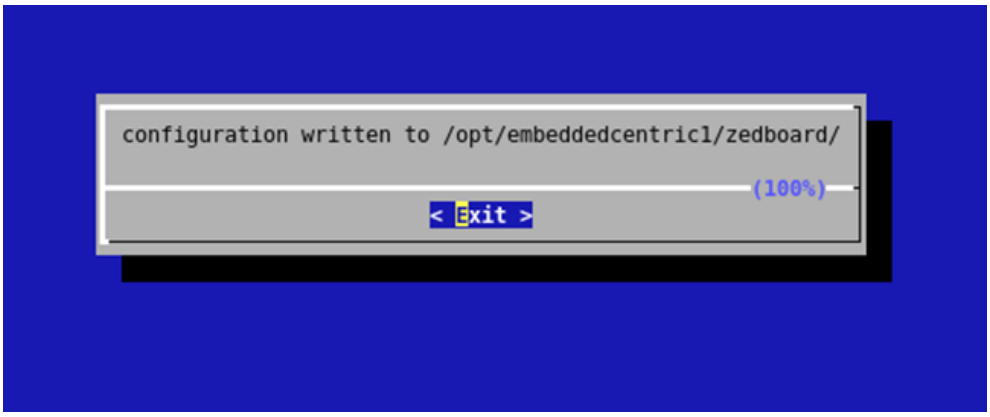
```
root@MICRO11:/opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4# petalinux-config -c rootfs
INFO: Checking component...
```

En la siguiente pantalla, moverse con las teclas de desplazamiento arriba y abajo hasta el submenú ***Apps*** y seleccionar la opción *gpio-demo* con la barra espaciadora. Finalmente ir hasta la opción ***Select,*** presionar ***Enter*** 3 veces y luego ***Save.***


administrador : petalinux-confi – Konsole
File Edit View Bookmarks Settings Help
/opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4/subsystems/linux/configs/rootfs/config - linux/rootfs Configuration
linux/rootfs Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
Filesystem Packages --->
Libs ----
Apps --->
Modules --->
PetaLinux RootFS Settings --->
Debugging --->
<Select> < Exit > < Help > < Save > < Load >


administrador : petalinux-confi – Konsole
File Edit View Bookmarks Settings Help
/opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4/subsystems/linux/configs/rootfs/config - linux/rootfs Configuration
→ Apps
Apps
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] echo_test --->
[*] fwupgrade ----
[*] gpio-demo ----
[*] latencystat ----
[*] mat_mul_demo --->
[*] peekpoke ----
[*] proxy_app --->

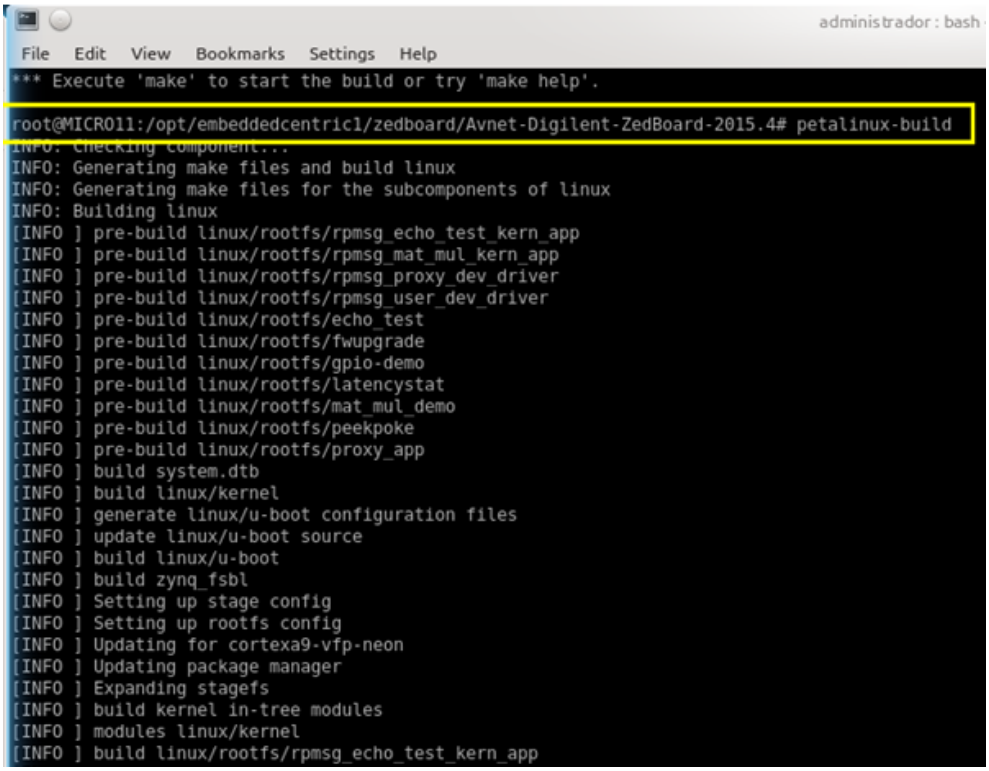

Enter a filename to which this configuration should be saved as an alternate. Leave blank to abort.
rd-2015.4/subsystems/linux/configs/rootfs/config
< Ok >
< Help >

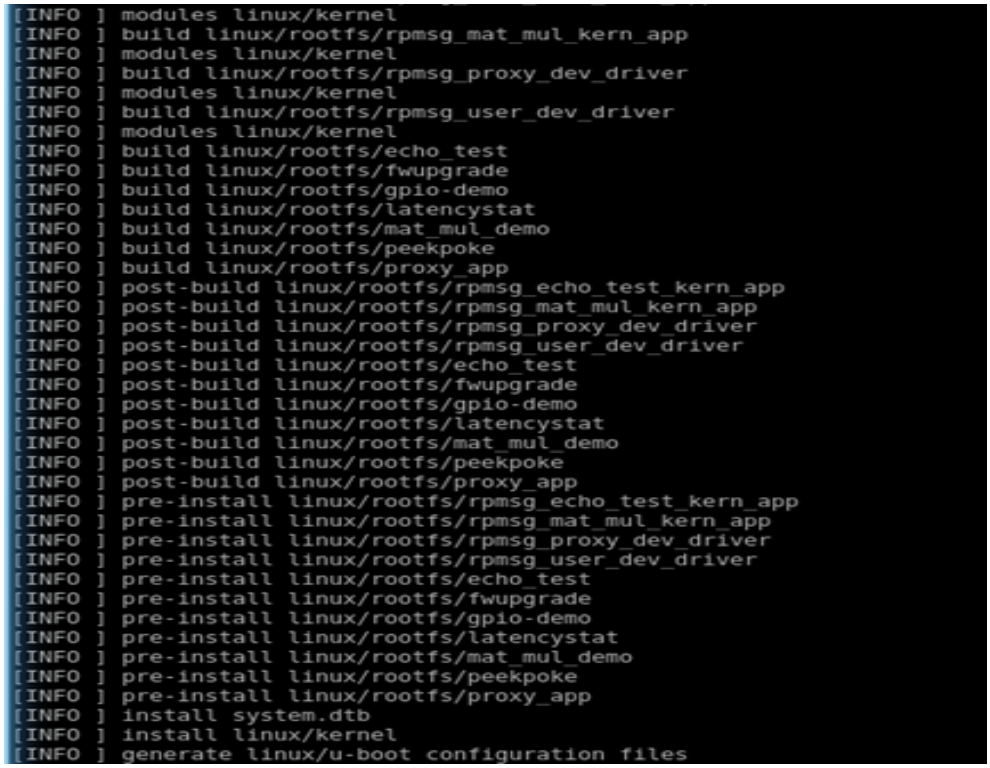


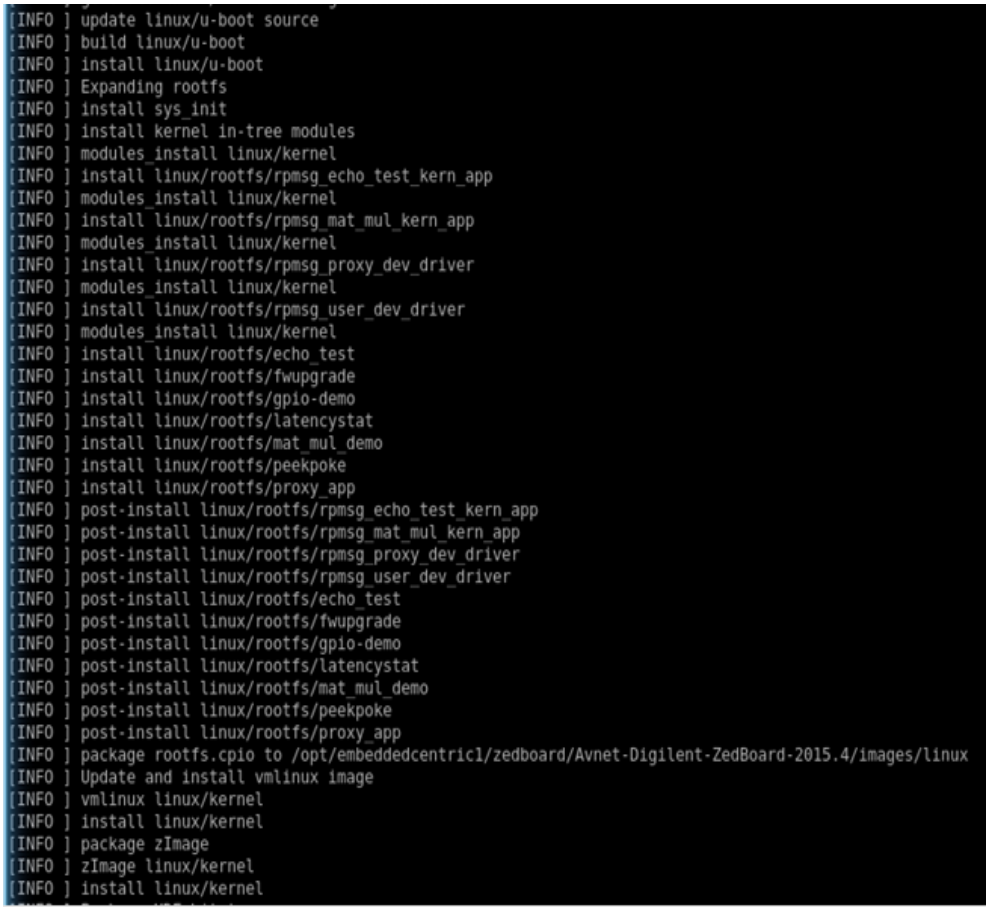
2.5 CONSTRUCCIÓN DE IMAGEN PETALINUX

En esta etapa ya se tiene el núcleo de Linux gracias a la instalación del BSP en pasos anteriores. Ir al directorio */opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-v2015.4/.* En esta sección se inicia un proceso de compilación cruzada. La construcción de la imagen se puede iniciar solo con escribir el siguiente comando: *petalinux-build*.

En un proceso de compilación cruzada éxitoso, se debe obtener lo siguiente:

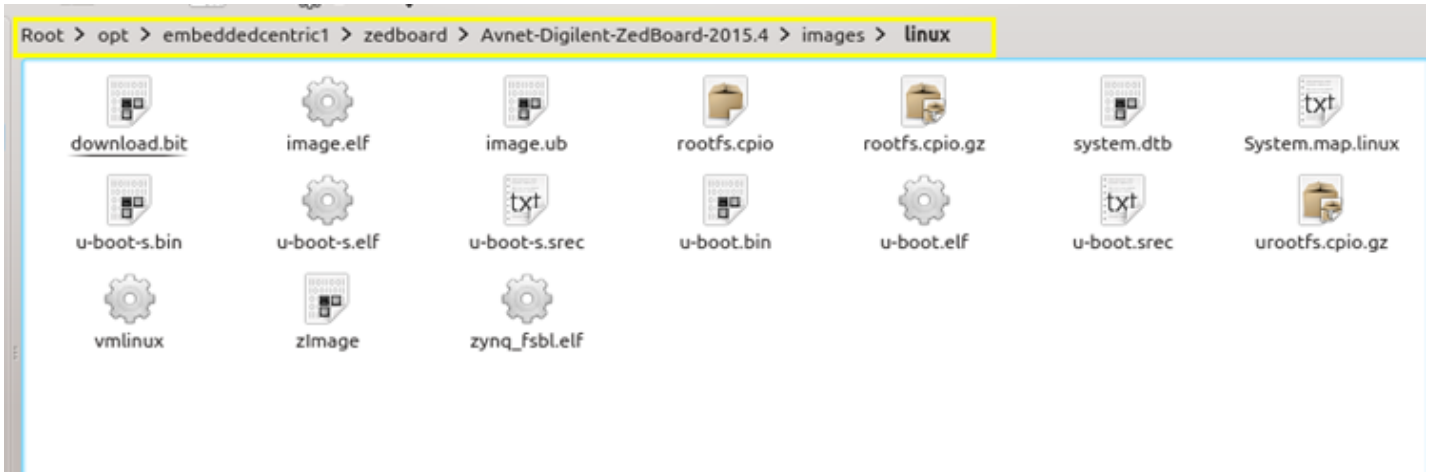



[INFO] modules linux/kernel
[INFO] build linux/rootfs/rpmsg_mat_mul_kern_app
[INFO] modules linux/kernel
[INFO] build linux/rootfs/rpmsg_proxy_dev_driver
[INFO] modules linux/kernel
[INFO] build linux/rootfs/rpmsg_user_dev_driver
[INFO] modules linux/kernel
[INFO] build linux/rootfs/echo_test
[INFO] build linux/rootfs/fwupgrade
[INFO] build linux/rootfs/gpio-demo
[INFO] build linux/rootfs/latencystat
[INFO] build linux/rootfs/mat_mul_demo
[INFO] build linux/rootfs/peekpoke
[INFO] build linux/rootfs/proxy_app
[INFO] post-build linux/rootfs/rpmsg_echo_test_kern_app
[INFO] post-build linux/rootfs/rpmsg_mat_mul_kern_app
[INFO] post-build linux/rootfs/rpmsg_proxy_dev_driver
[INFO] post-build linux/rootfs/rpmsg_user_dev_driver
[INFO] post-build linux/rootfs/echo_test
[INFO] post-build linux/rootfs/fwupgrade
[INFO] post-build linux/rootfs/gpio-demo
[INFO] post-build linux/rootfs/latencystat
[INFO] post-build linux/rootfs/mat_mul_demo
[INFO] post-build linux/rootfs/peekpoke
[INFO] post-build linux/rootfs/proxy_app
[INFO] pre-install linux/rootfs/rpmsg_echo_test_kern_app
[INFO] pre-install linux/rootfs/rpmsg_mat_mul_kern_app
[INFO] pre-install linux/rootfs/rpmsg_proxy_dev_driver
[INFO] pre-install linux/rootfs/rpmsg_user_dev_driver
[INFO] pre-install linux/rootfs/echo_test
[INFO] pre-install linux/rootfs/fwupgrade
[INFO] pre-install linux/rootfs/gpio-demo
[INFO] pre-install linux/rootfs/latencystat
[INFO] pre-install linux/rootfs/mat_mul_demo
[INFO] pre-install linux/rootfs/peekpoke
[INFO] pre-install linux/rootfs/proxy_app
[INFO] install system.dtb
[INFO] install linux/kernel
[INFO] generate linux/u-boot configuration files


[INFO] update linux/u-boot source
[INFO] build linux/u-boot
[INFO] install linux/u-boot
[INFO] Expanding rootfs
[INFO] install sys_init
[INFO] install kernel in-tree modules
[INFO] modules_install linux/kernel
[INFO] install linux/rootfs/rpmsg_echo_test_kern_app
[INFO] modules_install linux/kernel
[INFO] install linux/rootfs/rpmsg_mat_mul_kern_app
[INFO] modules_install linux/kernel
[INFO] install linux/rootfs/rpmsg_proxy_dev_driver
[INFO] modules_install linux/kernel
[INFO] install linux/rootfs/rpmsg_user_dev_driver
[INFO] modules_install linux/kernel
[INFO] install linux/rootfs/echo_test
[INFO] install linux/rootfs/fwupgrade
[INFO] install linux/rootfs/gpio-demo
[INFO] install linux/rootfs/latencystat
[INFO] install linux/rootfs/mat_mul_demo
[INFO] install linux/rootfs/peekpoke
[INFO] install linux/rootfs/proxy_app
[INFO] post-install linux/rootfs/rpmsg_echo_test_kern_app
[INFO] post-install linux/rootfs/rpmsg_mat_mul_kern_app
[INFO] post-install linux/rootfs/rpmsg_proxy_dev_driver
[INFO] post-install linux/rootfs/rpmsg_user_dev_driver
[INFO] post-install linux/rootfs/echo_test
[INFO] post-install linux/rootfs/fwupgrade
[INFO] post-install linux/rootfs/gpio-demo
[INFO] post-install linux/rootfs/latencystat
[INFO] post-install linux/rootfs/mat_mul_demo
[INFO] post-install linux/rootfs/peekpoke
[INFO] post-install linux/rootfs/proxy_app
[INFO] package rootfs.cpio to /opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4/images/linux
[INFO] Update and install vmlinux image
[INFO] vmlinux linux/kernel
[INFO] install linux/kernel
[INFO] package zImage
[INFO] zImage linux/kernel
[INFO] install linux/kernel

```
[INFO ] install linux/kernel
[INFO ] Package HDF bitstream
root@MICRO11:/opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4#
```

En el directorio */opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-v2015.4/images/linux/* quedaron almacenados 17 ficheros. Dichos ficheros posibilitan el arranque de Petalinux en la Zdeboard bajo diferentes métodos como el inicio desde el puerto JTAG, puerto Ethernet, QSPI Flash y la tarjeta SD (Embedded Centric, s.f.).



- **Download.bit:** flujo de bits para para el PL.
- **Image.elf:** imagen de Linux en formato ELF.
- **Image.ub:** imagen de Linux en formato U-Boot.
- **Rootfs.cpio:** imagen del sistema de archivos raíz.
- **Rootfs.cpio.gz:** imagen del sistema de archivos raíz (comprimido)
- **System.dtb:** árbol de dispositivos.
- **System.map.linux:** dirección de memoria de mapas para el sistema Linux.
- **U-boot.bin:** imagen U-Boot en formato binario.
- **U-boot.elf:** imagen U-Boot en formato ELF.
- **U-boot.srec:** imagen U-Boot en formato SREC.
- **U-boot-s.bin:** reubicable U-Boot para el sistema operativo en formato binario.
- **U-boot-s.elf:** reubicable U-Boot para el sistema operativo en formato ELF.

- **U-boot-s.srec:** reubicable U-Boot para el sistema operativo en formato SREC.
- **Urootfs.cpio.gz:** U-Boot del sistema de archivos raíz (comprimido)
- **Vmlinux:** ***Kernel*** del Linux en formato de archivo ejecutable vinculado estáticamente.
- **Zlmage:** Imagen del ***Kernel*** de Linux.
- **Zynq_fsbl.elf:** primera etapa del gestor de arranque en formato ELF.

2.6 GENERACIÓN DE PAQUETE BOOT.BIN

Para iniciar Petalinux en las Zedboard se conocen diferentes métodos, en este caso ha sido iniciado desde una memoria SD. El archivo de arranque se copia directamente a la tarjeta SD. El paquete *BOOT.BIN* contiene los siguientes archivos (Embedded Centric, s.f.):

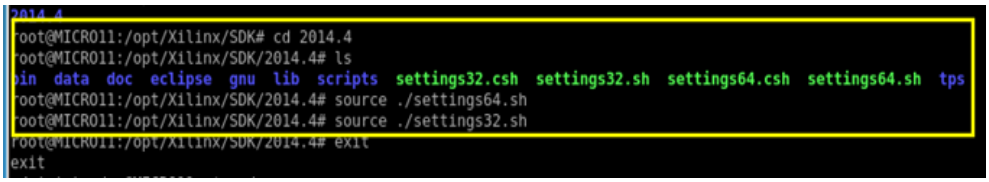
- **Download.bit**: corresponde al archivo de flujo de datos para la parte PL de la tarjeta.
- **Zynq_fsbl.elf:** corresponde a la primera etapa del gestor de arranque LPSA para la parte del PS de la tarjeta.
- **U-BOOT.ELF:** corresponde al gestor de arranque universal, que se encarga de cargar la imagen del núcleo de Linux.

El paquete BOOT.BIN se genera mediante la configuración de la herramienta *petalinux-packaje*. Cabe mencinar que dicha herramienta proviene de SDK Xilinx, el cual debe ser instalado antes de proceder con el paso principal de este apartado (Embedded Centric, s.f.).

Antes de continuar con el propósito principal de este paso, es necesario ir la ruta *cd /opt/Xilinx/SDK/2014.4/* y repetir los pasos de la sección 2.2 cuando se configura la fuente de las herramientas (settings.sh). Digitar los siguientes comandos:

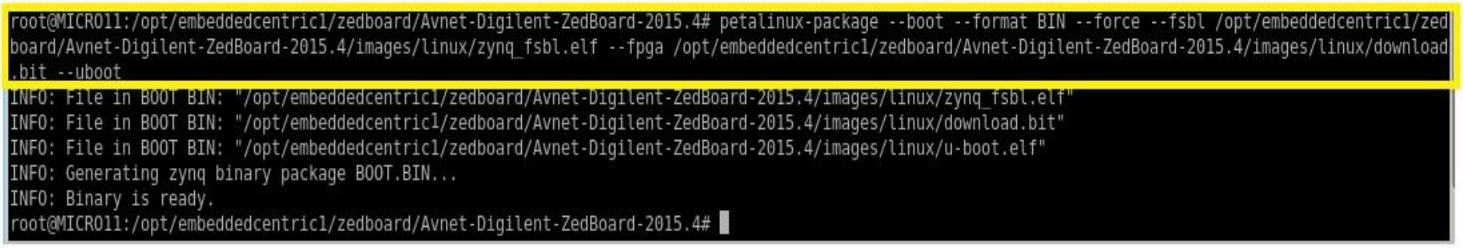
source ./settings64.sh

source ./settings32.sh

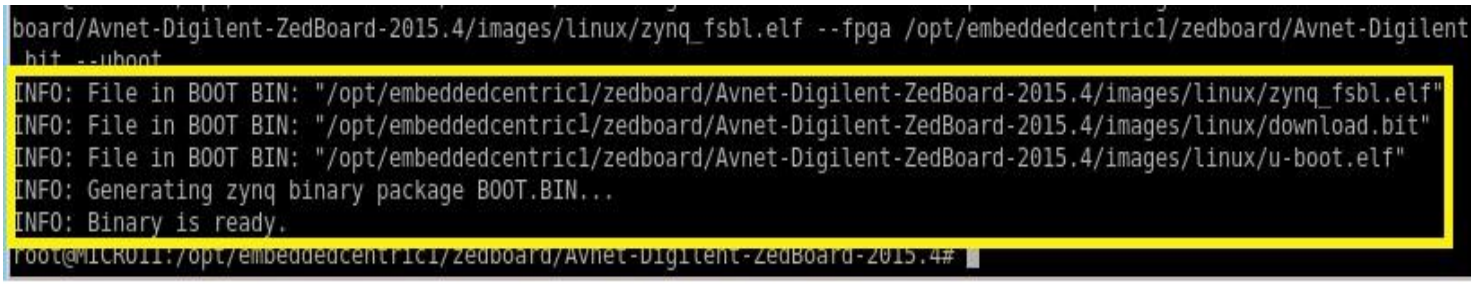


Para generar el paquete BOOT.BIN se digita el siguiente comando:

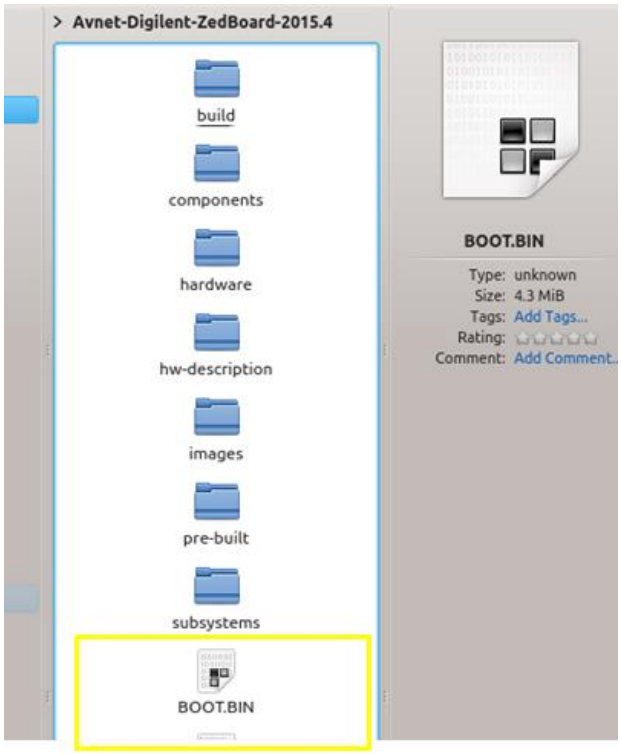
petalinux-package --boot --format BIN --force --fsbl /opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4/images/linux/zynq_fsbl.elf --fpga /opt/embeddedcentric1/zedboard/Avnet-Digilent-ZedBoard-2015.4/images/linux/download.bit --uboot

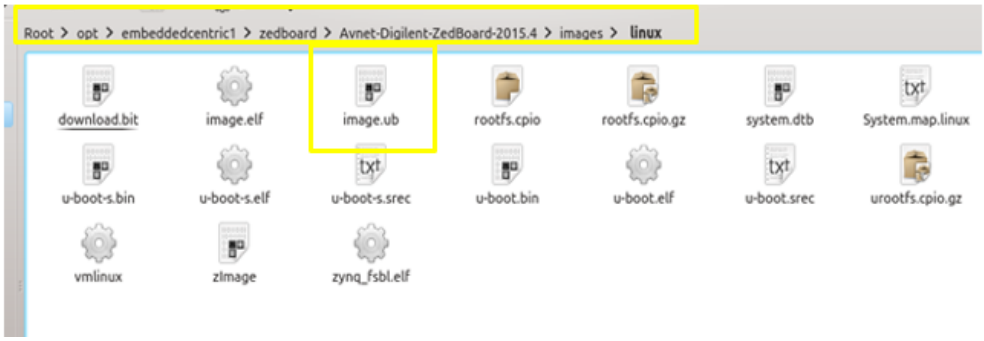


El siguiente pantallazo evidencia que el paquete BOOT.BIN ha sido creado de manera exitosa.



Finalmente se obtiene un archivo “BOOT.BIN” e “image.ub” que corresponde a la imagen del ***kernel*** de Linux. Posteriormente, ambos se copian a la tarjeta SD. Ir a las siguientes rutas:



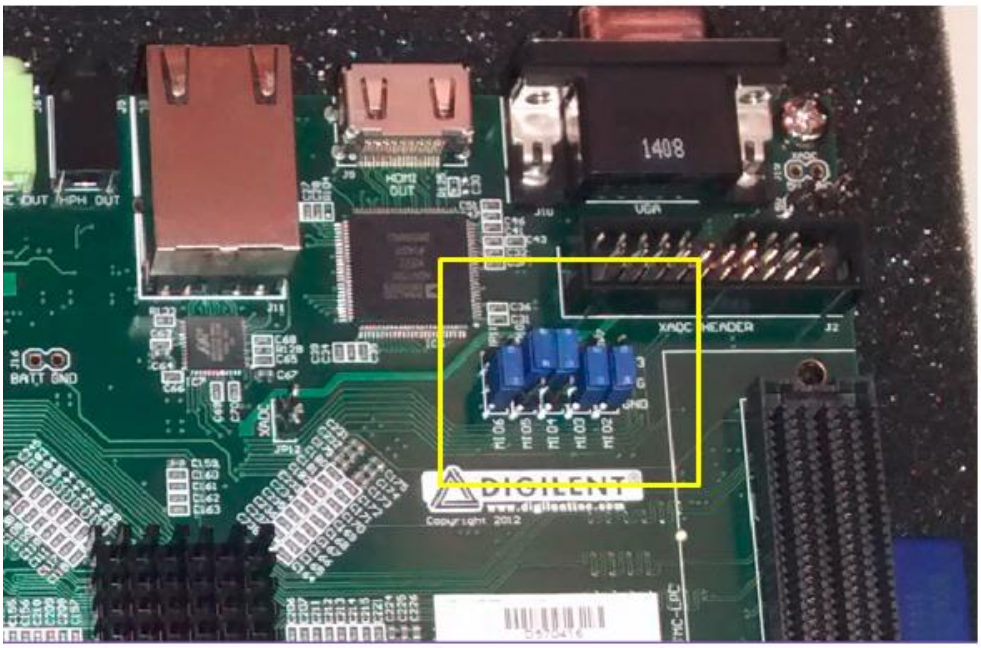


2.7 INICIO DE PETALINUX DESDE LA ZEDBOARD

Al momento de iniciar Petalinux desde la tarjeta SD, tener presente que los puentes JP9 (MIO4) y JP10 (MIO5) situados en la placa Zedboard deben estar a nivel lógico alto es decir a 3.3V.

En la siguiente tabla se evidencia el modo de configuración de la Zedboard (Embedded Centric, s.f.):

Xilinx TRM→	MIO[6] Boot_Mode[4]	MIO[5] Boot_Mode[0]	MIO[4] Boot_Mode[2]	MIO[3] Boot_Mode[1]	MIO[2] Boot_Mode[3]
JTAG Mode					
Cascaded JTAG					0
Independent JTAG					1
Boot Devices					
JTAG		0	0	0	
Quad-SPI		1	0	0	
SD Card		1	1	0	
PLL Mode					
PLL Used	0				
PLL Bypassed	1				
Bank Voltages					
MIO Bank 500			3.3V		
MIO Bank 501			1.8V		

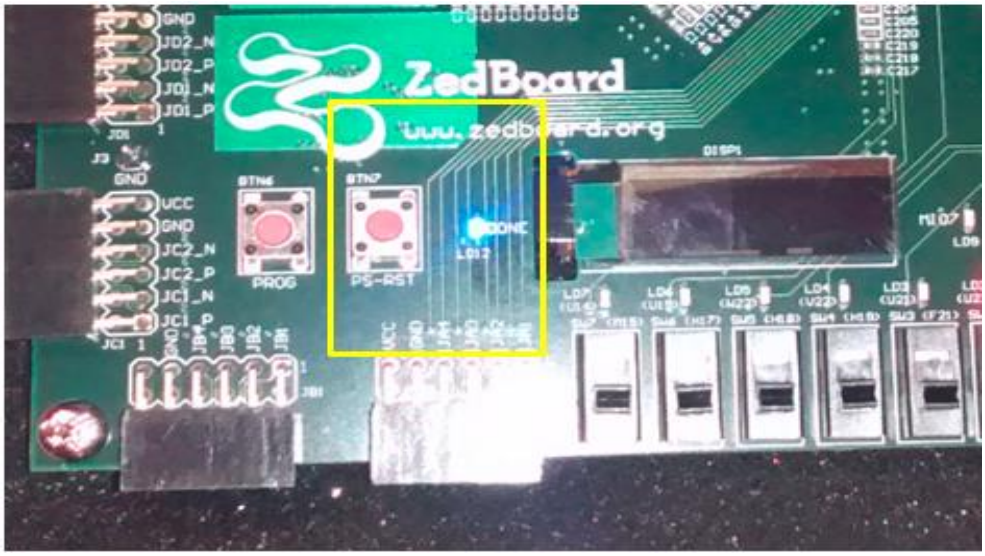


Asimismo, se debe establecer conexión con cable USB entre el equipo anfitrión y el puerto UART-USB de la tarjeta.

Para acceder a la tarjeta Zedboard a través del puerto serie se debe digitar desde el host el siguiente comando: sudo *chmod 666 /dev/ttyACM0.* Luego de cambiar los respectivos permisos sobre el puerto serial, se emplea la aplicación *gtkterm.* Digitar el siguiente comando: *gtkterm.*

En caso de presentar algun error al tratar de ingresar al puerto ttyACM0, repetir el paso de la sección 2.1. En resumen, digitar el comando *gedit .gtktermrc* y modificar otra vez el puerto y la velocidad del puerto serial.

Si al ingresar a *gtkterm* no se logra visualizar la ventana de inicio de la Zedboard es posible que se haya perdido la secuencia de arranque. Para esto es necesario reiniciar Petalinux desde el botón *PS-RST* de la Zedboard.



La imagen muestra la pantalla de arranque de Petalinux, por defecto se emplean las siguientes credenciales.

- **Avnet entrada-Digilent-ZedBoard-2015_2:** root
- **Contraseña:** root

Cabe mencionar que las credenciales aplican para la versión de Avnet Diligent Zedboard 2015.4 y Avnet Diligent Zedboard 2015.2.

3. REFERENCIAS

Embedded Centric. (s.f.). Obtenido de https://embeddedcentric.com/embedded-operating-systems

Xilinx INC. (s.f.). Zedboard.org. Obtenido de http://zedboard.org/sites/default/files/product_spec_images/Front-image-of-board_0.jpg

Xilinx INC. (s.f.). Xilinx, All programmable. Obtenido de https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/2015-4.html

GUÍA DE PRUEBA DE SERVICIOS, PROTOCOLOS Y APLICACIONES DE RED

1. RECURSOS REQUERIDOS

- Equipo de cómputo 64bits
- Kubuntu 14.04
- Vivado 2014.4 (debe incluir SDK)
- Avnet Diligent Zedboard 2015.4
- Petalinux 2015.4
- Tarjeta Zedboard con FPGA de la familia Zynq-7000 de Xilinx
- Cable USB micro
- Fuente de alimentación de 12V
- Tarjeta SD de 4GB
- Patch cord UTP

2. PROCEDIMIENTO DE LA PRÁCTICA

Los sistemas embebidos son un pilar fuerte en el ámbito del IoT. La lógica embebida proporciona el control remoto, la monitorización y la oportunidad de inspeccionar y analizar fuentes de datos. La necesidad de conectar estos aparatos a Internet se ha vuelto prioridad y al día de hoy los fabricantes de circuitos integrados y componentes electrónicos tienen una amplia gama de productos que cumplen dicha finalidad. Ethernet permitirá orientar hacia la conexión las aplicaciones desarrolladas gracias a los sistemas embebidos.

Los sistemas operativos integrados son sistemas diseñados para funcionar en plataformas con recursos limitados de hardware por ejemplo en tarjetas de desarrollo. El sistema operativo propicia la interacción y el aprovechamiento de los componentes de las tarjetas, en este caso para las tarjetas Zedboard con FPGA de la familia Zynq-7000 de Xilinx.

Ejemplo de ello es la utilización de los servicios y protocolos de red que vienen incorporados en el sistema operativo. El uso del sistema operativo permite ahorrar la complejidad de emplear un lenguaje de bajo nivel. Tal es el caso que esta herramienta favorece a una persona con conocimientos en ***networking*** y sin la experiencia en hardware configurar una red de datos empleando las tarjetas Zedboard.

Esta guía permite identificar las capacidades de red integradas en Petalinux. Básicamente, se explica el paso a paso para lograr que una tarjeta Zedboard se conecte a una red Ethernet. Dicha práctica se verifica a través de la instalación y configuración de diferentes servicios, protocolos y aplicaciones de red.

2.1 CORRER EL INSTALADOR DE PETALINUX

En esta sección se debe repetir la secuencia de pasos empleados para configurar las variables de entorno y rutas de búsqueda para las herramientas de Petalinux. Remitirse a *GUÍA DE INSTALACIÓN DE PETALINUX ítem 2.2.*

Tener presente que desde la ruta *cd /opt/embeddedcentric2/petalinux-v2015.4-final*, iniciar sesión como súper usuario (*sudo su*) y digitar los siguientes comandos: *chmod 777 ./settings.sh* y posteriormente *source ./settings.sh,* con el fin de hacer ejecutable dicho archivo (Embedded Centric, s.f.).

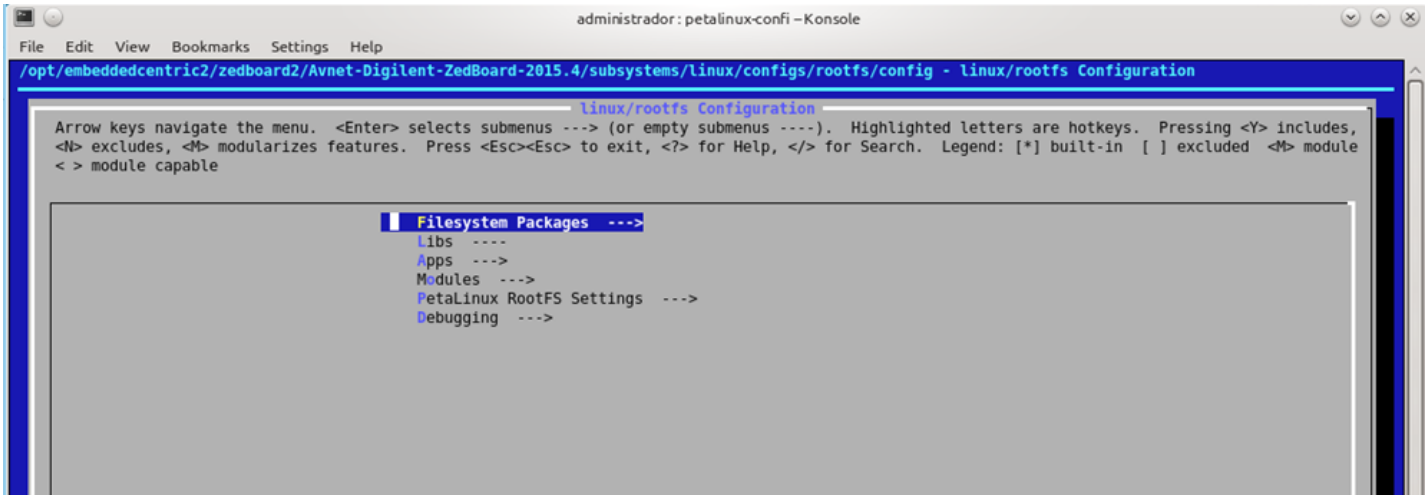
2.2 CONFIGURACIÓN DE LOS ARCHIVOS RAÍZ DEL SISTEMA

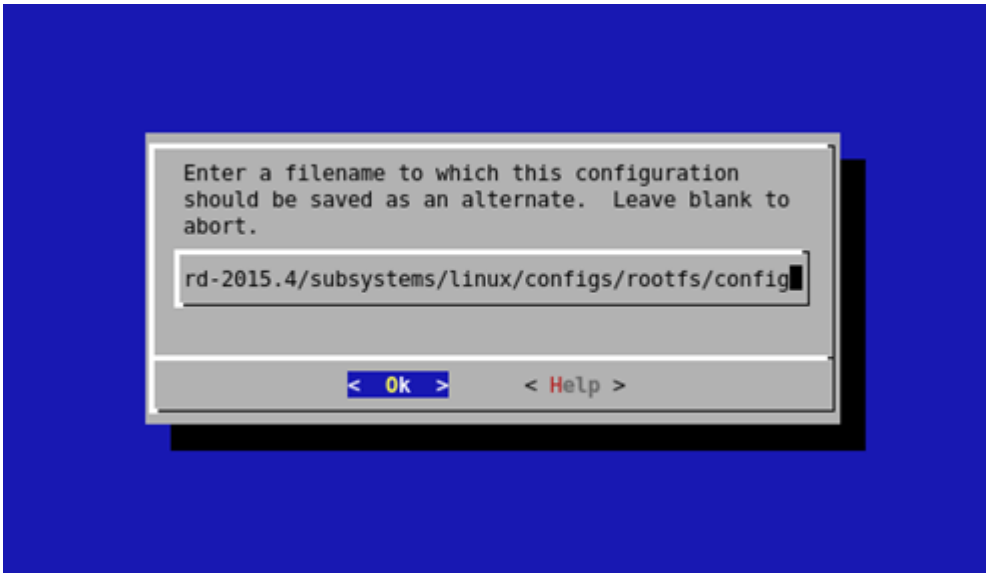
Iniciar sección como súper usuario (*sudo su*). Ir al directorio */opt/embeddedcentric2/zedboard/Avnet-Digilent-ZedBoard-v2015.4/.* Digitar el comando: *petalinux-config -c rootfs.*

El comando *petalinux-config -c rootfs* permite desplegar el menú de configuración del sistema de archivos raíz. En este paso se adiciona la aplicación *Dropbear* y *Busybox HTTPD*.

La aplicación de red *Dropbear* se compone de los paquetes dropbear y openssh-sftp-server dropbear. La aplicación funciona sobre el servidor SSH. Dropbear (SSH) sirve para ingresar remotamente a los dispositivos de la red. Dicho proceso, integra la aplicación al código fuente del ***kernel*** de Petalinux.

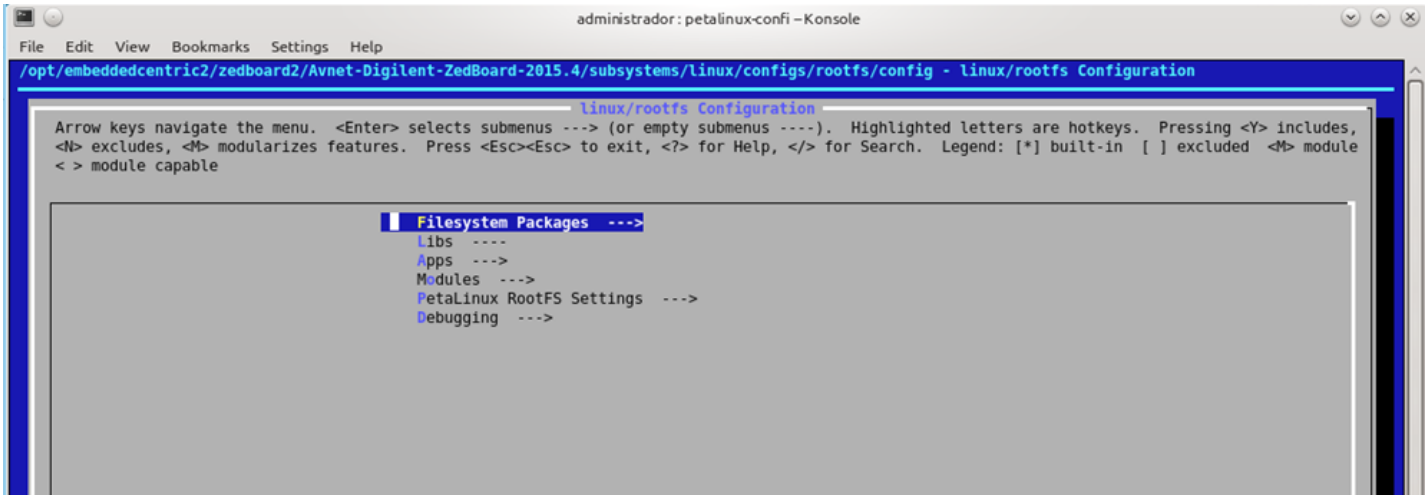
En la siguiente pantalla, moverse con las teclas de desplazamiento arriba y abajo hasta el submenú ***Filesystem Packages*** -> ***console/network*** -> ***dropbear*** y seleccionar la opción *dropbear* y *dropbear-openssh-sftp-server* con la barra espaciadora. Presionar ***Escape*** 4 veces para regresar al menú de configuración del sistema de archivos raíz.





El programa *Busybox* incorpora dentro de sus utilidades el programa servidor Apache HTTPD (HTTP ***Daemon***). Busybox HTTPD generalmente es empleado en sistemas con Linux embebido. El programa funciona como servidor ***web***. Dicha aplicación es integrada al código fuente del ***kernel*** de Petalinux.

En la siguiente pantalla, moverse con las teclas de desplazamiento arriba y abajo hasta el submenú ***Filesystem Packages*** -> ***base*** -> ***busybox*** y seleccionar la opción *busybox-httpd.* Desplazarse hasta la opción ***Save,*** presionar ***Enter*** 3 veces y salir del menú de configuración.



Con la configuración de *busybox-httpd* se logra el ingreso a una página ***web*** que trae por defecto Petalinux. Por medio del explorador (aplicación de cliente) instalada en el equipo anfitrión, se escribe la dirección IP asignada a la Zedboard y se accede a dicha página.

2.3 CONSTRUCCIÓN DE IMAGEN PETALINUX

En esta sección se debe repetir la secuencia de pasos empleados para iniciar el proceso de compilación cruzada y construir la nueva imagen de Petalinux incluyendo las aplicaciones dropbear y busybox (Embedded Centric, s.f.). Remitirse a *GUÍA DE INSTALACIÓN DE PETALINUX ítem 2.5.*

Tener presente, ir al directorio */opt/embeddedcentric2/zedboard2/Avnet-Digilent-ZedBoard-v2015.4/.* Para empezar la construcción de la imagen digitar el comando: *petalinux-build* (Embedded Centric, s.f.).

2.4 GENERACIÓN DE PAQUETE BOOT.BIN

En esta sección se debe repetir la secuencia de pasos empleados para generar y copiar el archivo de arranque para la Zedboard. Remitirse a *GUÍA DE INSTALACIÓN DE PETALINUX ítem 2.6.*

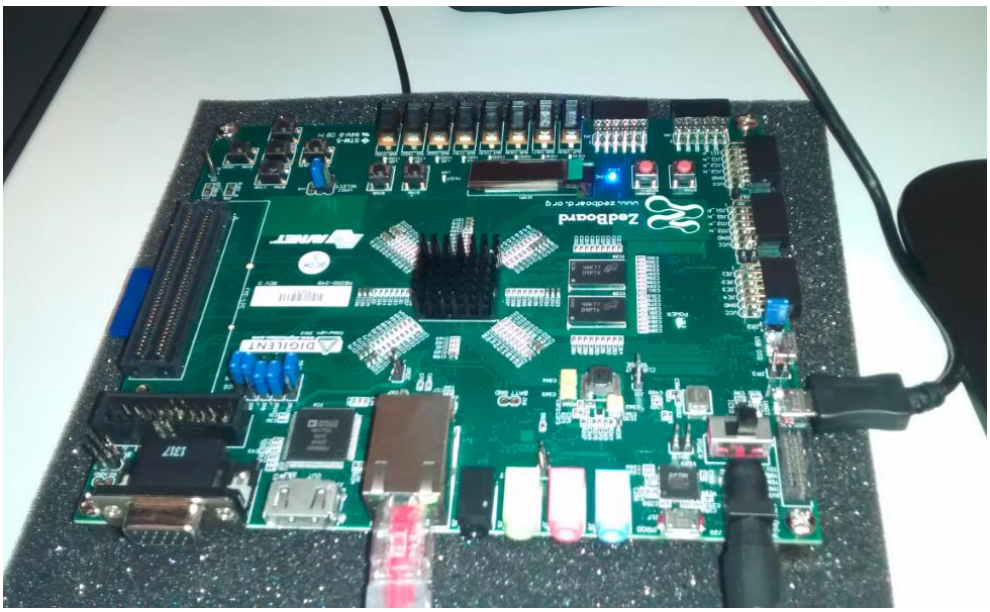
Considerar que en este caso se inicia Petalinux desde una tarjeta SD. El archivo de arranque se copia directamente a la tarjeta SD. Para generar el paquete BOOT.BIN digitar el siguiente comando:

petalinux-package --boot --format BIN --force --fsbl /opt/embeddedcentric2/zedboard2/Avnet-Digilent-ZedBoard-2015.4/images/linux/zynq_fsbl.elf --fpga /opt/embeddedcentric2/zedboard2/Avnet-Digilent-ZedBoard-2015.4/images/linux/download.bit –uboot

2.5 INICIO DE PETALINUX DESDE LA ZEDBOARD

En esta sección se debe repetir la secuencia de pasos empleados para iniciar Petalinux desde la tarjeta SD. Adicional a esto, seguir el procedimiento de preparación para el arranque de las Zedboard. Remitirse a *GUÍA DE INSTALACIÓN DE PETALINUX ítem 2.7.*

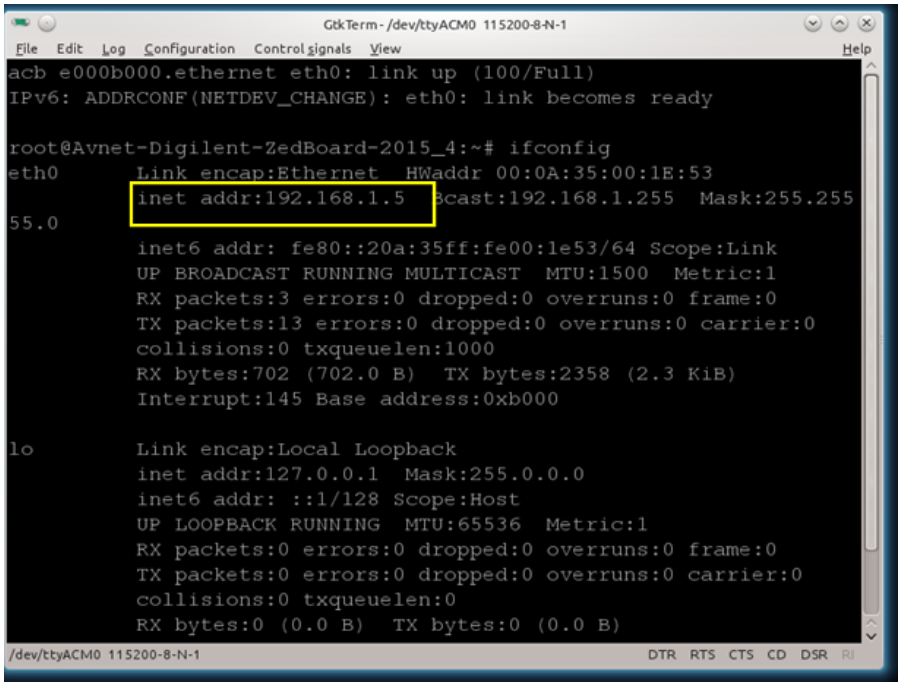
La forma más básica para establecer una conexión de red entre una estación de trabajo (anfitrión) y una tarjeta Zedboard, es disponer una conexión punto a punto entre las interfaces de red de ambos dispositivos mediante un ***patch cord*** de UTP (Embedded Centric, s.f.).

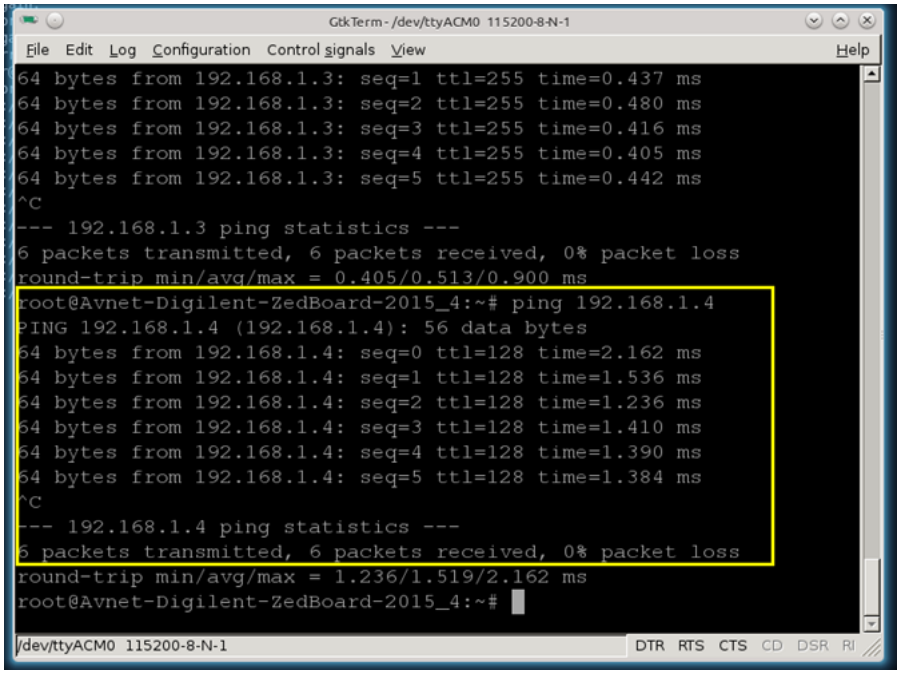


2.6 PRUEBA DE CONECTIVIDAD

El ***ping*** dispone del protocolo ICMP para validar conectividad entre los dispositivos de la red. Dicha herramienta diagnóstica ayuda a identificar el origen de las fallas en una red. Solo se requiere escribir el siguiente comando seguido de la dirección IP del dispositivo con el que se requiere probar la conectividad:

ping dirección IP





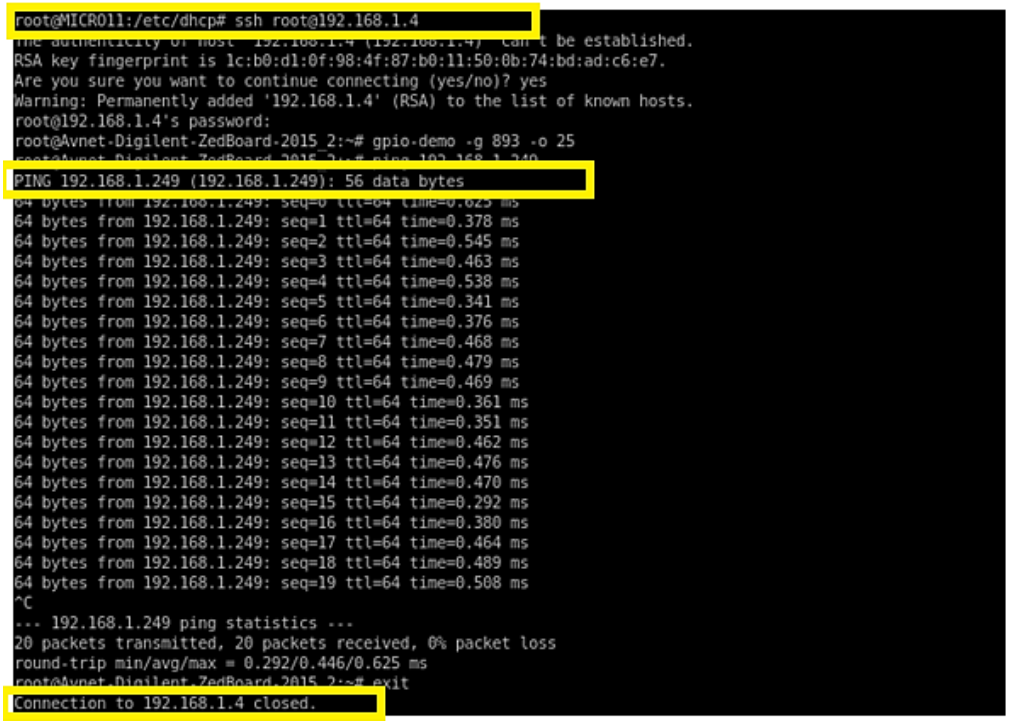
2.7 CONEXIÓN ENTRE ZEDBOARD A TRAVÉS DE SSH

En vista de que en pasos anteriores fue adicionada la aplicación necesaria para utilizar el servidor SSH (dropbear), basta con abrir la terminal del host central y colocar lo siguiente:

ssh root@ dirección IP

Nota: en la línea de comando se coloca la dirección IP de la Zedboard o **host** con el que se desea establecer conexión.

Luego, se solicita la confirmación del registro y la contraseña del dispositivo, para este caso la contraseña por defecto de la tarjeta es *root.* La siguiente figura muestra la conexión desde el equipo anfitrión a una Zedboard, en la terminal se efectúa un ***ping*** que evidencia la comunicación entre las mismas.



Con el servicio SSH incorporado en Petalinux se consigue acceso completo a los recursos de la Zedboard (Embedded Centric, s.f.).

3. REFERENCIAS

Embedded Centric. (s.f.). Obtenido de https://embeddedcentric.com/embedded-operating-systems

Embedded Centric. (s.f.). Obtenido de https://embeddedcentric.com/networked-systems/

Xilinx INC. (s.f.). *Xilinx, All programmable*. Obtenido de https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/2015-4.html

Institución Universitaria	MODALIDAD TRABAJO DE GRADO PRODUCTO OBTENIDO EN TALLERES O LABORATORIOS DEL ITM Registro de actividades y cumplimiento de horas / Talleres o Laboratorios de DOCENCIA	Código	FDE 146
		Versión	02
		Fecha	2015-09-30

Documento de identidad:	*1128435201*		
Nombre completo del estudiante:	*Leidy Johana Beltrán Usme*		
Programa académico ITM:	*Ingeniería en telecomunicaciones*		
Nombre completo del Docente Asesor:	*Luis Fernando Castaño Londoño*		
Fecha de iniciación del producto (asa/mm/de):	*15/02/2016*	**Fecha de terminación del producto (asa/mm/de):**	*17/06/2016*
Nombre Taller o Laboratorio:	*Sistemas de control y robótica*		
Ubicación:	*Parque I*		
Campus:	*Fraternidad*		

Fecha A	Fecha M	Fecha D	Actividad desempeñada por el estudiante	Hora ingreso	Hora salida	Total horas	Firma Laboratorista	Firma Estudiante
16	2	15	Socialización del producto a ejecutar con el docente	6.00pm	8.00pm	2	Luis Fernando Castaño L	Leidy B
16	2	20	Socialización de cronograma y de plan de trabajo con el docente	10.00am	12.00pm	2	Luis Fernando Castaño L	Leidy B.
16	2	29	Ambientación con las Zedboard (componentes, funcionalidades, usos, servicios...)	7:30pm	9.30pm	2	Luis Fernando Castaño L	Leidy B.
16	3	4	Ambientación con las Zedboard (componentes, funcionalidades, usos, servicios...)	7:30pm	9.30pm	2	Luis Fernando Castaño L	Leidy B
16	3	14	Socialización de guía de laboratorio. #10 "Sistemas operativos embebidos (embeddedcentric)	7:30pm	9.30pm	2	Luis Fernando Castaño L	Leidy B
16	3	18	Descarga y ubicación en sistema de instalador de Petalinux (Petalinux-v2015.4) y Zedboard BSP (Avnet-Digilent-ZedBoard-v2015.4)	7.30pm	9.30pm	2	Luis Fernando Castaño L	Leidy B
16	3	19	Inicia construcción de la imagen de Linux (Petalinux), instalación de librerías, configuración de bash, Gtkterm	10.00am	12.00pm	2	Luis Fernando Castaño L	Leidy B.
16	3	28	Adecuación de host para posterior instalación de Linux embebido, es decir instalación de librerías, configuración de bash, Gtkterm (terminal de la Zedboard)	7:30pm	9.30pm	2	Luis Fernando Castaño L	Leidy B
16	4	1	Ejecución del instalador de Petalinux, creación de nuevo proyecto, instalación de GPIO-demo...	7:30pm	9.30pm	2	Luis Fernando Castaño L	Leidy B
16	4	2	Creación de imagen de Petalinux	10.00am	12.00pm	2	Luis Fernando Castaño L	Leidy B

16	4	4	Creación de imagen de Petalinux (errores y búsqueda de solución)	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	8	Creación de imagen de Petalinux (errores y búsqueda de solución)	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	11	Creación de imagen de Petalinux (comprobación de correcta instalación)	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	15	Arranque del sistema desde la tarjeta SD (Zedboard) y conexión entre host y Zedboard vía serial (UART-USB) entre otros	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	16	Configuración de Gtkterm, acceso a la tarjeta mediante puerto serial, pruebas de comandos y aplicaciones del Petalinux cargado en la Zedboard	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B
16	4	18	Comprobación y validación del contenido del GPIO-demo (encendido y apagado de led mediante pulsadores, switchs)	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	20	Socialización de guía de laboratorio #11 "Sistemas conectados en red (embeddedcentric)	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	23	Instalación necesaria para crear ambiente de trabajo adecuado para poner en practica la guía # 11 (construir nuevamente imagen de Petalinux la guía #10)	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B
16	4	25	Instalación necesaria para crear ambiente de trabajo adecuado para poner en practica la guía # 11 (construir nuevamente imagen de Petalinux la guía #10)	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	4	30	Búsqueda de información acerca de servicios de DHCP, SSH, FTP, WEB (caracteristicas, funcionalidad entre otros)	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B
16	5	2	Búsqueda de información acerca de configuración de servicios de DHCP, SSH, FTP, WEB en ambiente Linux, cómo aplicarlos a Petalinux	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	5	7	Inclusión de aplicaciones de red y protocolos en Petalinux	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B
16	5	11	Activación y configuración del DHCP Server en el host central y verificación desde Zedboard via serial, conexión via SSH desde el host hacia la Zedboard	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	5	13	Activación y configuración de servidor FTP intercambiando archivos entre el host central y la Zedboard, prueba de Comunicación con el servidor de la página web por defecto de Petalinux	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	5	14	Búsqueda de información sobre equipos de networking a utilizar en este caso se empleó el router 2901.	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B
16	5	16	Búsqueda de información acerca de topologías de red, elección de topología adecuada para el montaje con las Zedboard	7:30pm	9:30pm	2	Luis Fernando Cortés	Leidy B
16	5	21	Diseño de la topología de red a implementar, estableciendo equipos activos en la red y direccionamiento	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B

16	5	23	Diseño de la topología de red a implementar, estableciendo equipos activos en la red y direccionamiento.	6:00pm	8:00pm	2	Luis Fernando Cortés	Leidy B
16	6	1	Gestión y préstamo de equipos para el montaje	10:00am	12:00pm	2	Luis Fernando Cortés	Leidy B
16	6	1	Montaje de la topología tipo estrella, es decir conexionado físico entre Switch, router, Zedboard y host.	01:00pm	04:00pm	3	Luis Fernando Cortés	Leidy B
16	6	2	Instalación, configuración y verificación de Team viewer para proveer una conexión remota con el fin de facilitar el acceso a los equipos.	2:00pm	4:00pm	2	Luis Fernando Cortés	Leidy B
16	6	4	Configuración del interfaces del router 2901 y activación del DHCP Server en el mismo, configuración de Zedboard como DHCP Client (Vía remota)	2:00pm	4:00pm	2	Luis Fernando Cortés	Leidy B
16	6	7	Comprobación de conectividad (vía remota) entre los dispositivos de la topología estrella (errores y búsqueda de solución)	2:00pm	4:00pm	2	Luis Fernando Cortés	Leidy B
16	6	8	Comprobación de conectividad entre los dispositivos de la topología estrella (comprobación de correcto funcionamiento)	2:00pm	4:00pm	2	Luis Fernando Cortés	Leidy B
			TOTAL HORAS					69

Leidy B.

Firma Estudiante

Luis Fernando Cortés Londoño

Nombre y firma Laboratorista

Nombre y firma Profesional Universitario - Centro de Laboratorios

FIRMA ESTUDIANTES Leidy B.

FIRMA ASESOR Luis Fernando Cortés Londoño

FECHA ENTREGA: 29/09/2016

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO___ ACEPTADO____ CON MODIFICACIONES_____

ACTA NO._____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD_____

ACTA NO._____

FECHA ENTREGA: _____