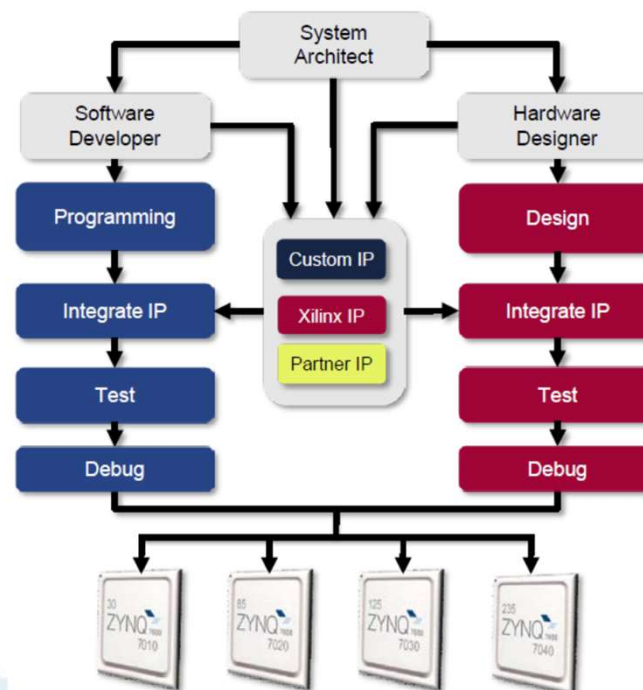


Embedded Linux on Zynq



김현옥 (hokim1972@naver.com)

- More efficient parallel hardware and software development.
-
- ```
graph TD; SA[System Architect] --> P1[Parallel Hardware Development]; SA --> P2[Parallel Software Development]; SA --> P3[Parallel Hardware and Software Development];
```



- **Bare-Metal** without OS

Possible to write fast programs.

Difficult to realize complex systems with components like webserver, FTP server, USB support and so on.

SDK Provides ARM Toolchain.

- **Real-Time OS**

For applications with strict time constraints.

Most are commercial.

- **Linux**

Greatest flexibility.

Many drivers, libraries and software package.



- **Vivado** GUI, CUI(2015.4)

Design hardware

Generate bit/hdf files

- **HSI** Hardware Software Interface

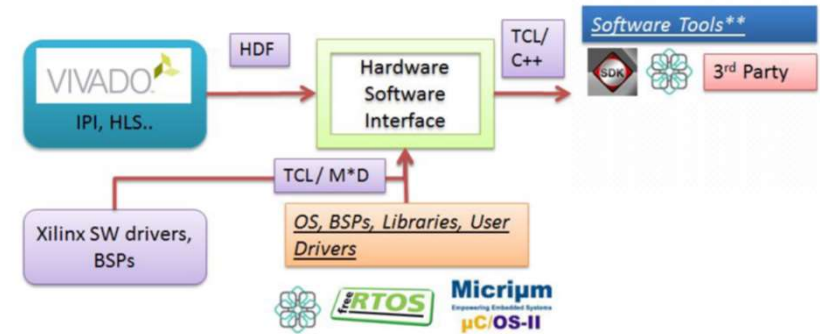
Scalable framework enabled SW tool integration with Vivado

Compile FSBL / Generate DT

- **Ubuntu** (16.04 LTS)

(Host) Compile u-boot, kernel and DT

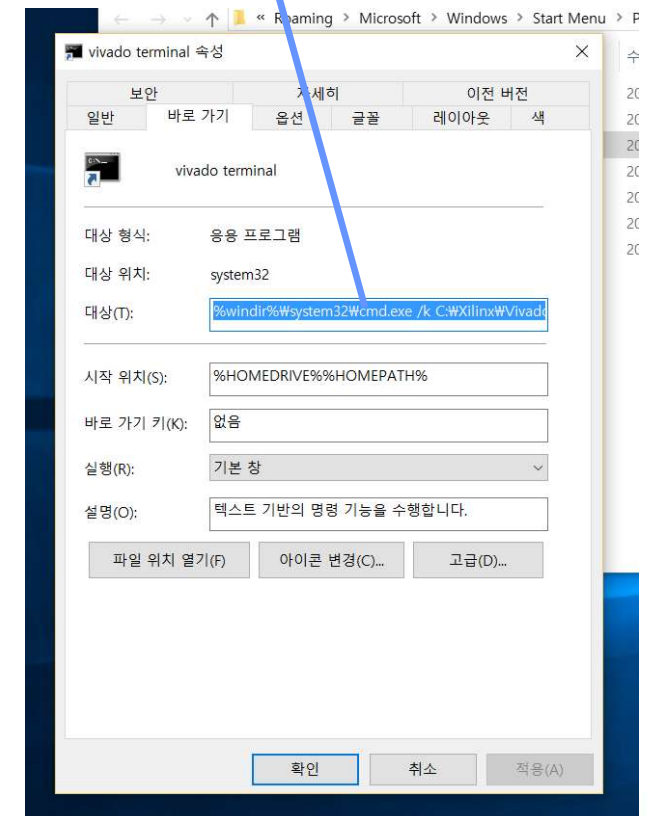
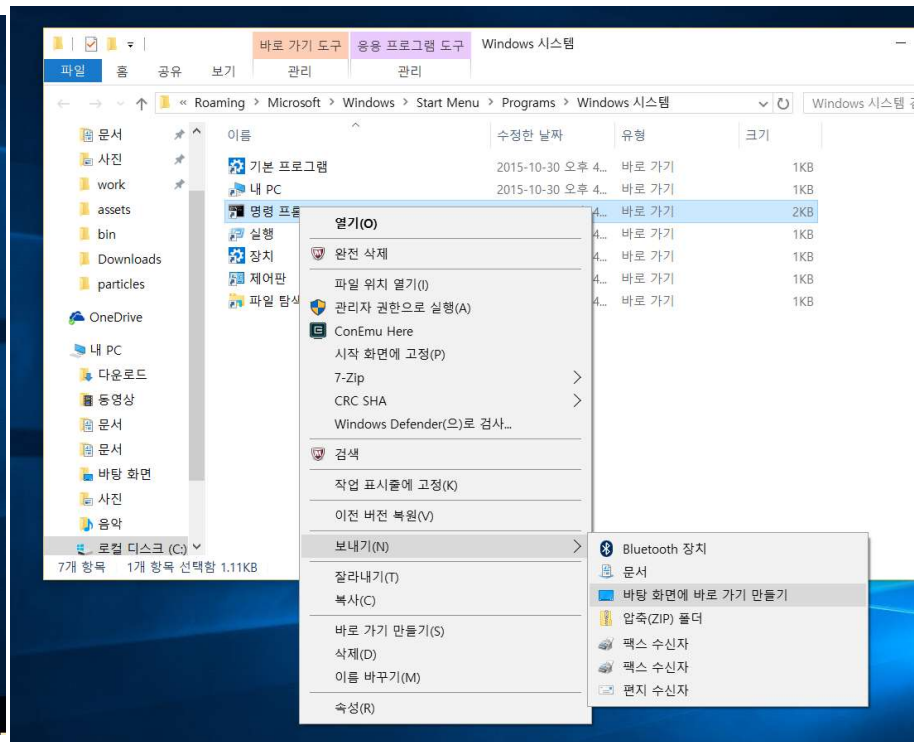
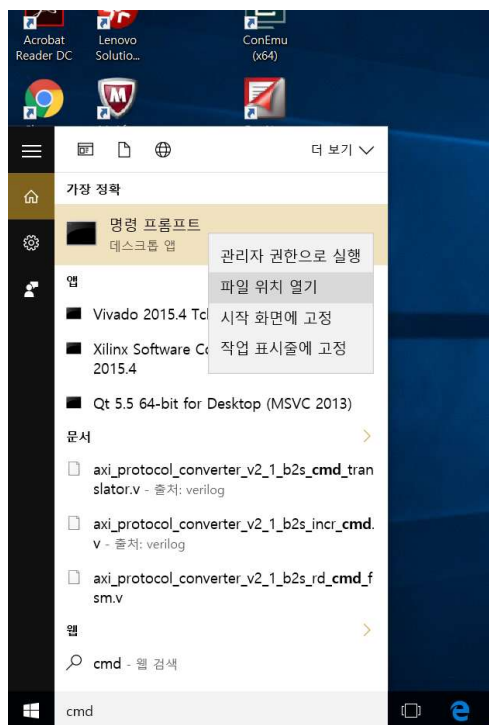
(Target) Compile drivers / applications



## ■ Windows 10

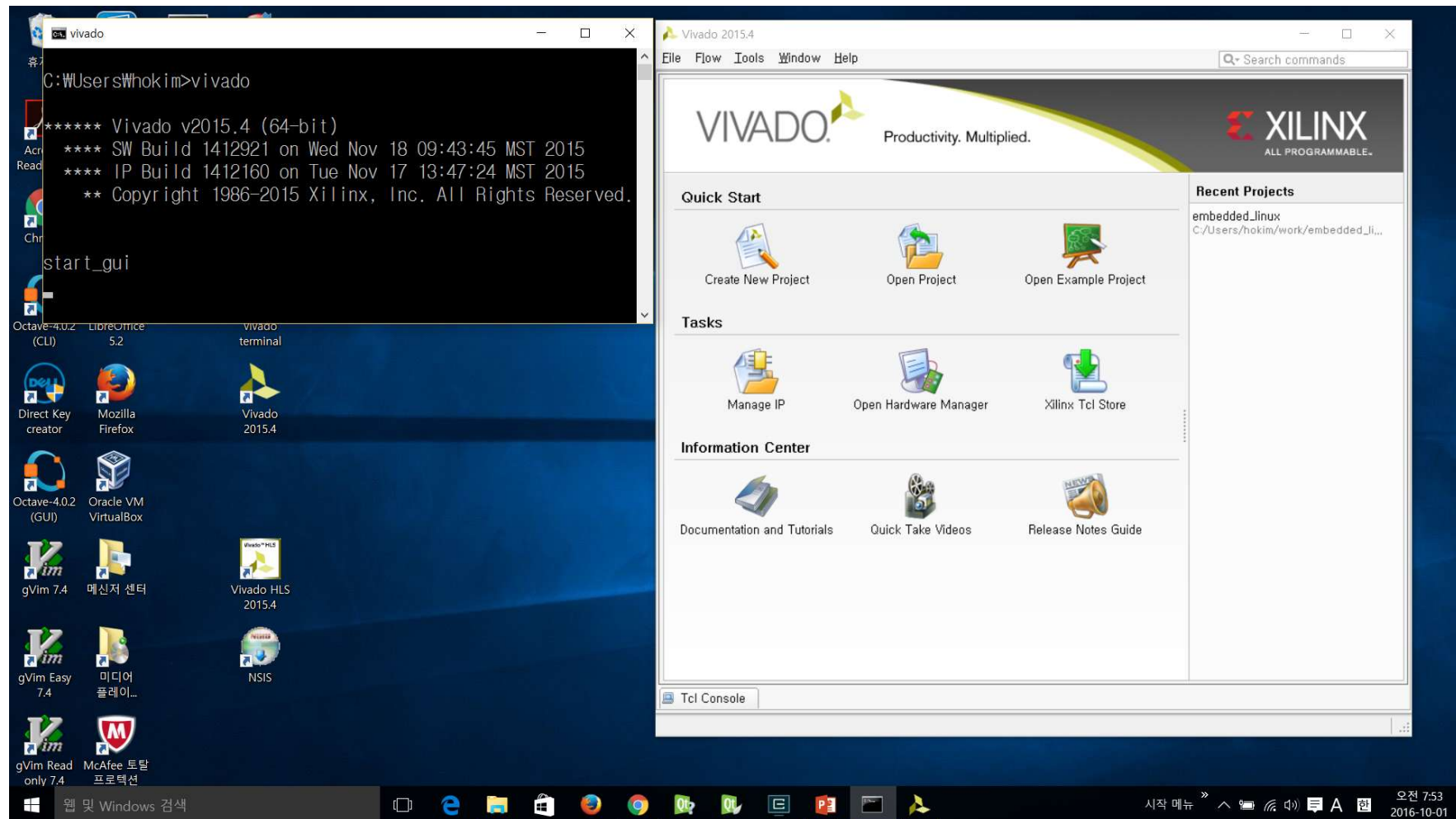
Install Vivado 2015.4 w/ SDK.  
cmd terminal for Vivado.

`%windir%\system32\cmd.exe /k C:\Xilinx\Vivado\2015.4\settings64.bat`

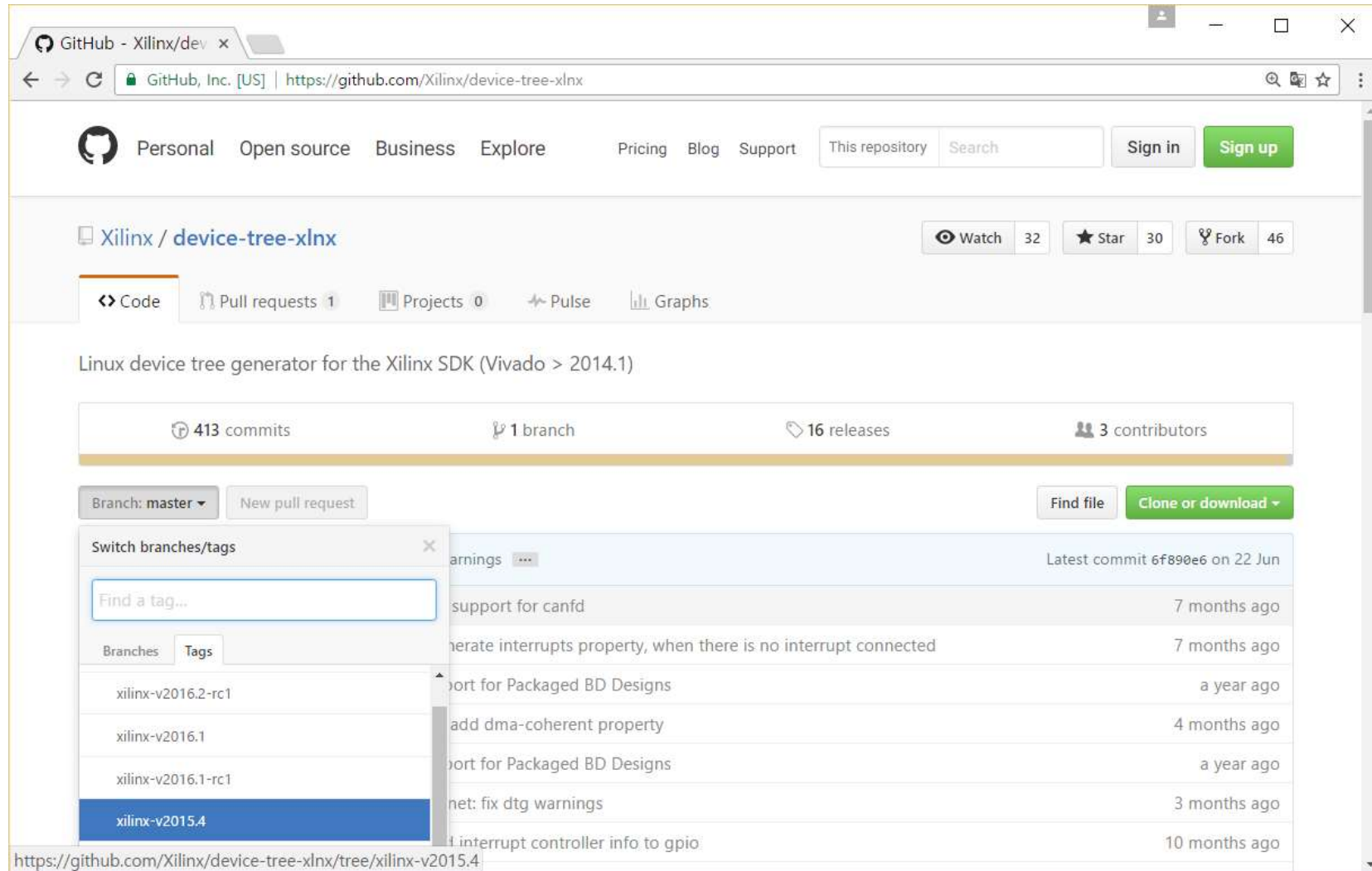




- **cmd terminal** for Vivado.



## ■ Device Tree Source



GitHub - Xilinx/dev x

GitHub, Inc. [US] | <https://github.com/Xilinx/device-tree-xlnx>

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

Xilinx / device-tree-xlnx

Watch 32 Star 30 Fork 46

Code Pull requests 1 Projects 0 Pulse Graphs

Linux device tree generator for the Xilinx SDK (Vivado > 2014.1)

413 commits 1 branch 16 releases 3 contributors

Branch: master New pull request Find file Clone or download

Switch branches/tags

Find a tag...

Branches Tags

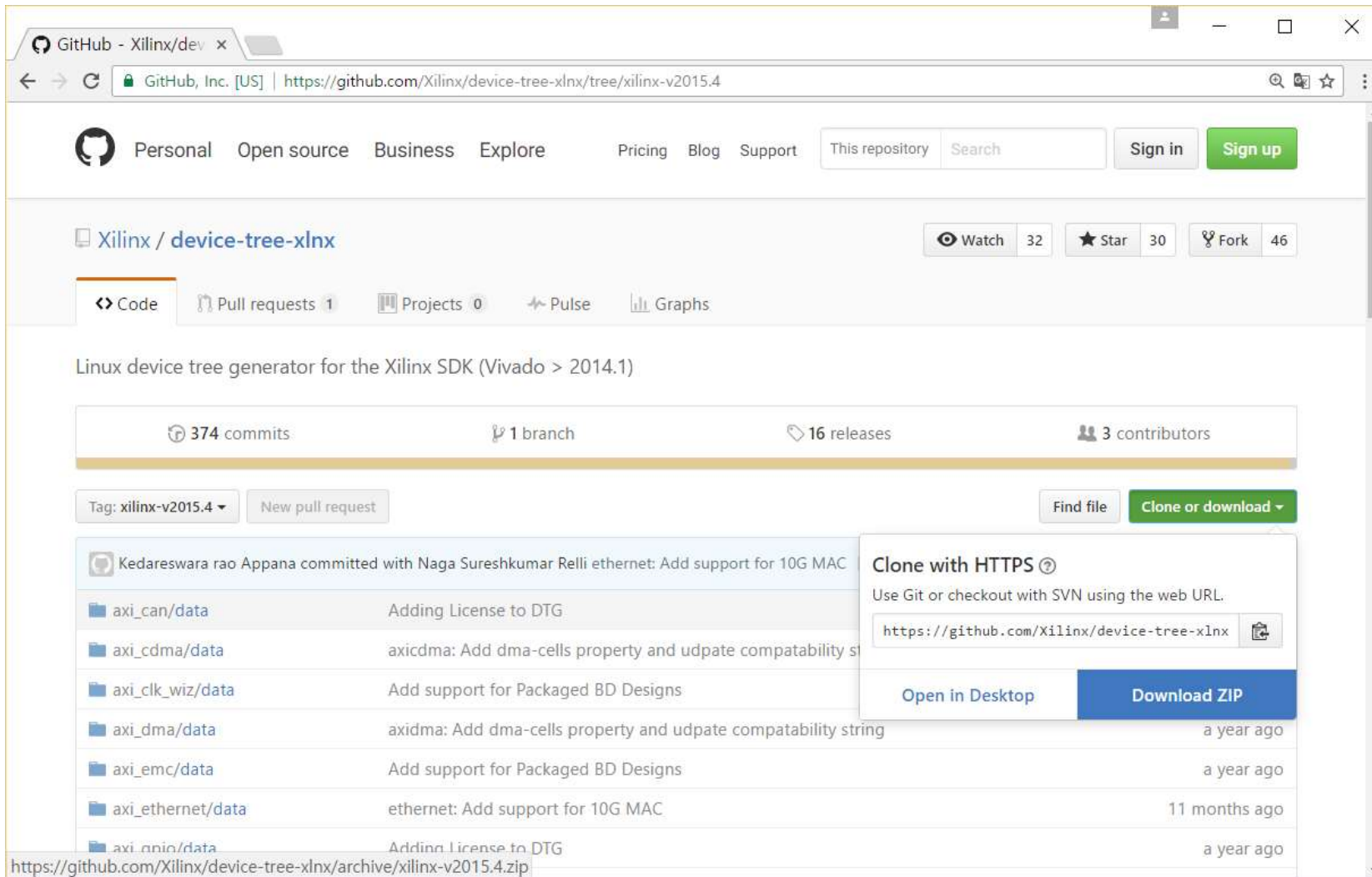
- xilinx-v2016.2-rc1
- xilinx-v2016.1
- xilinx-v2016.1-rc1
- xilinx-v2015.4**

Latest commit 6f890e6 on 22 Jun

- support for canfd 7 months ago
- generate interrupts property, when there is no interrupt connected 7 months ago
- support for Packaged BD Designs a year ago
- add dma-coherent property 4 months ago
- support for Packaged BD Designs a year ago
- net: fix dtg warnings 3 months ago
- add interrupt controller info to gpio 10 months ago

<https://github.com/Xilinx/device-tree-xlnx/tree/xilinx-v2015.4>

## ■ Device Tree Source



GitHub - Xilinx/dev x

GitHub, Inc. [US] | <https://github.com/Xilinx/device-tree-xlnx/tree/xilinx-v2015.4>

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

Xilinx / **device-tree-xlnx** Watch 32 Star 30 Fork 46

Code Pull requests 1 Projects 0 Pulse Graphs

Linux device tree generator for the Xilinx SDK (Vivado > 2014.1)

374 commits 1 branch 16 releases 3 contributors

Tag: xilinx-v2015.4 New pull request Find file Clone or download

Kedareswara rao Appana committed with Naga Sureshkumar Relli ethernet: Add support for 10G MAC

|                   |                                                                 |               |
|-------------------|-----------------------------------------------------------------|---------------|
| axi_can/data      | Adding License to DTG                                           |               |
| axi_cdma/data     | axicdma: Add dma-cells property and update compatability string |               |
| axi_clk_wiz/data  | Add support for Packaged BD Designs                             |               |
| axi_dma/data      | axidma: Add dma-cells property and update compatability string  | a year ago    |
| axi_emc/data      | Add support for Packaged BD Designs                             | a year ago    |
| axi_ethernet/data | ethernet: Add support for 10G MAC                               | 11 months ago |
| axi_nnino/data    | Adding License to DTG                                           | a year ago    |

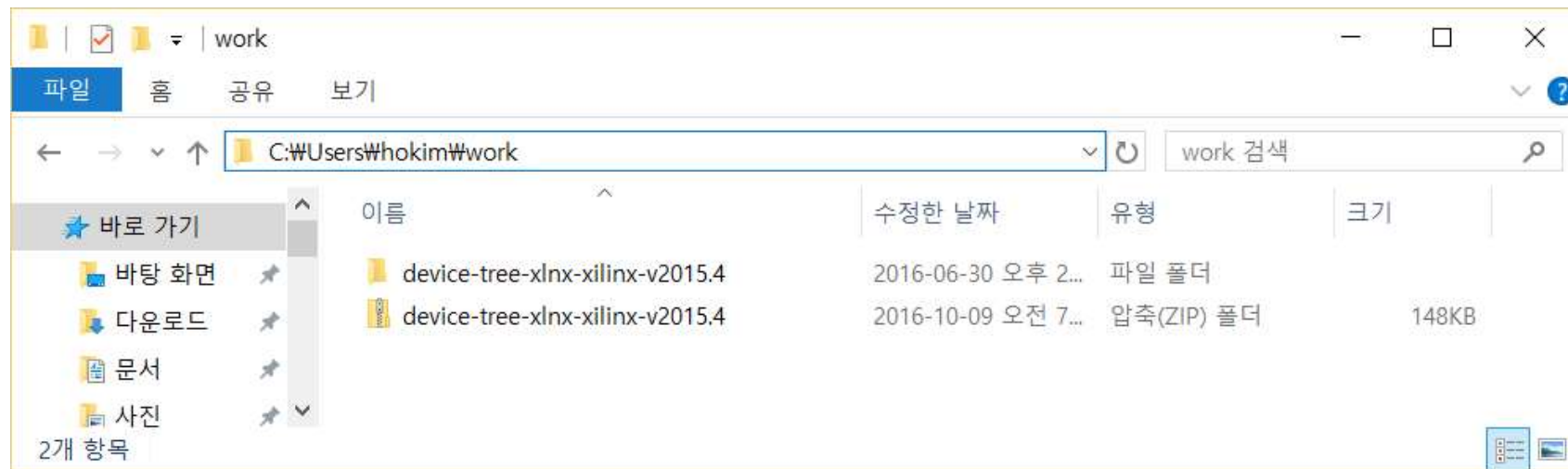
<https://github.com/Xilinx/device-tree-xlnx/archive/xilinx-v2015.4.zip>

Clone with HTTPS ?  
Use Git or checkout with SVN using the web URL.  
<https://github.com/Xilinx/device-tree-xlnx>

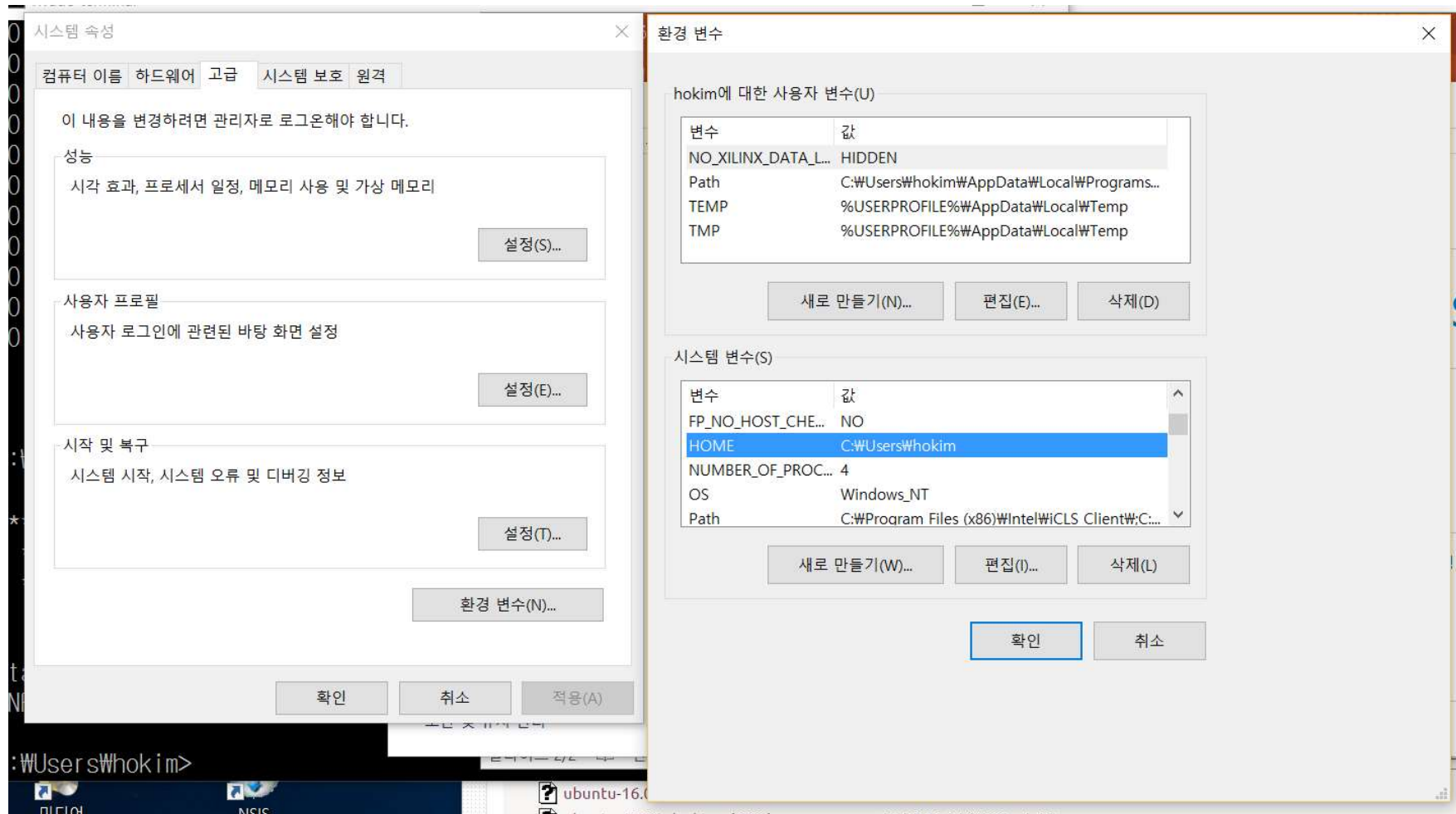
Open in Desktop Download ZIP



## ■ Device Tree Source



## ■ Environment Variable(HOME)



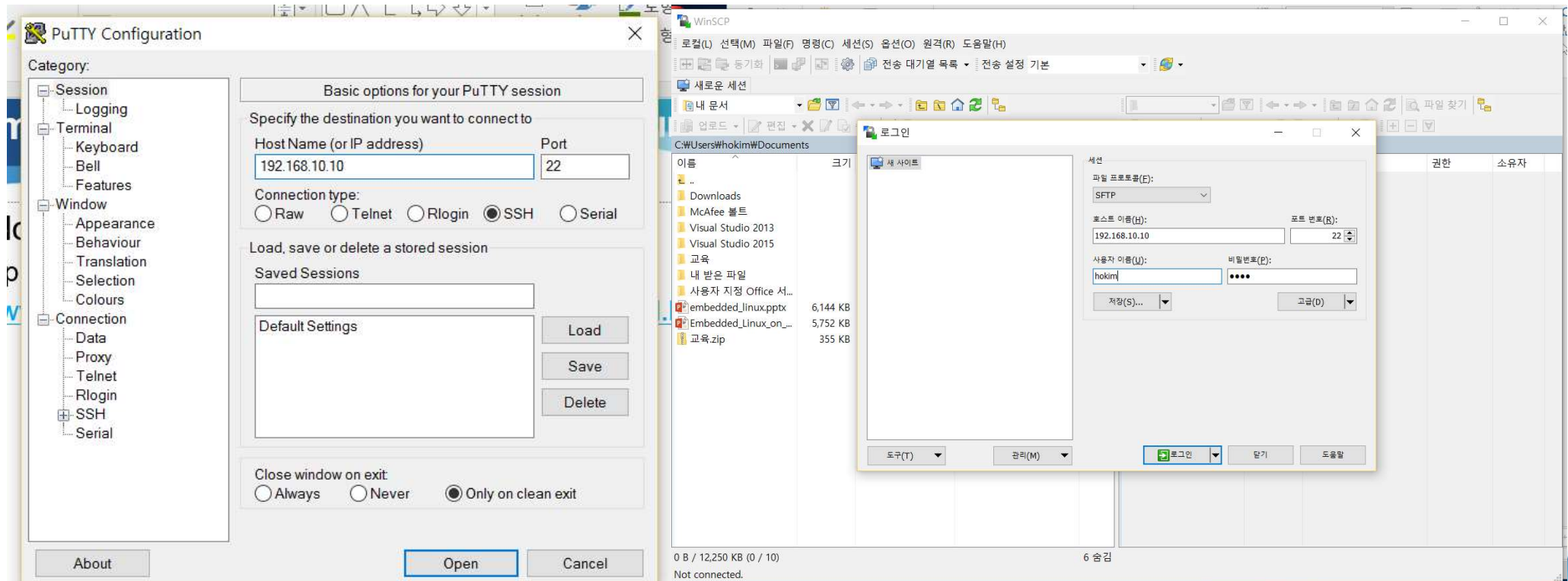
## ■ Windows ↔ Ubuntu

Install putty.exe for remote login shell.

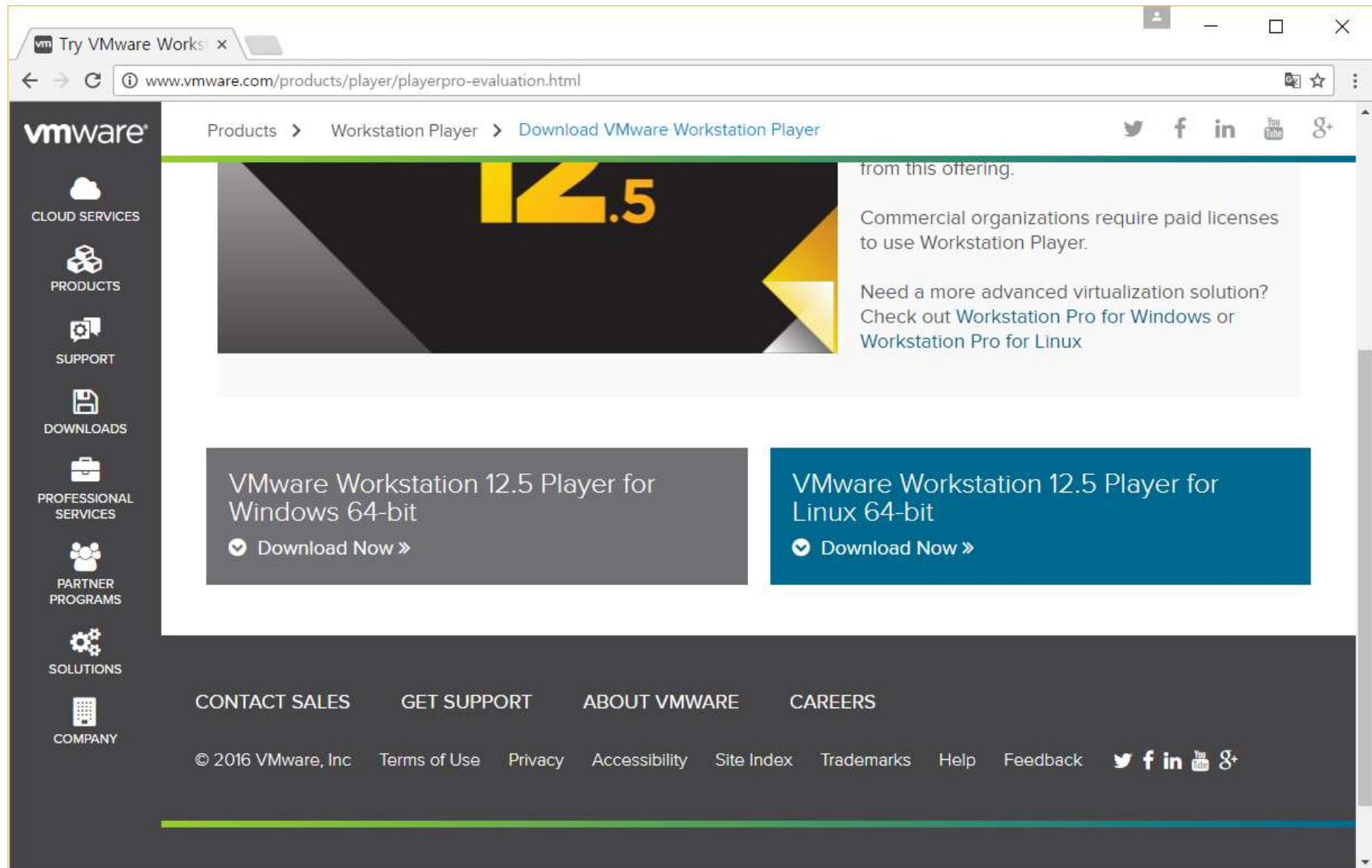
(<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)

Install WinSCP for file transfer.

(<https://winscp.net/eng/download.php> )



## ■ Ubuntu on VMware-player



The screenshot shows the VMware Workstation 12.5 Player download page. The browser address bar displays [www.vmware.com/products/player/playerpro-evaluation.html](http://www.vmware.com/products/player/playerpro-evaluation.html). The page features a dark sidebar with navigation links: CLOUD SERVICES, PRODUCTS, SUPPORT, DOWNLOADS, PROFESSIONAL SERVICES, PARTNER PROGRAMS, SOLUTIONS, and COMPANY. The main content area includes a large banner for VMware Workstation 12.5 Player, a note about commercial licenses, and two download buttons: "VMware Workstation 12.5 Player for Windows 64-bit" and "VMware Workstation 12.5 Player for Linux 64-bit". The footer contains links for CONTACT SALES, GET SUPPORT, ABOUT VMWARE, and CAREERS, along with copyright information and social media icons.

vmware Try VMware Workstation x

www.vmware.com/products/player/playerpro-evaluation.html

Products > Workstation Player > Download VMware Workstation Player

12.5

from this offering.

Commercial organizations require paid licenses to use Workstation Player.

Need a more advanced virtualization solution? Check out [Workstation Pro for Windows](#) or [Workstation Pro for Linux](#)

VMware Workstation 12.5 Player for Windows 64-bit  
Download Now »

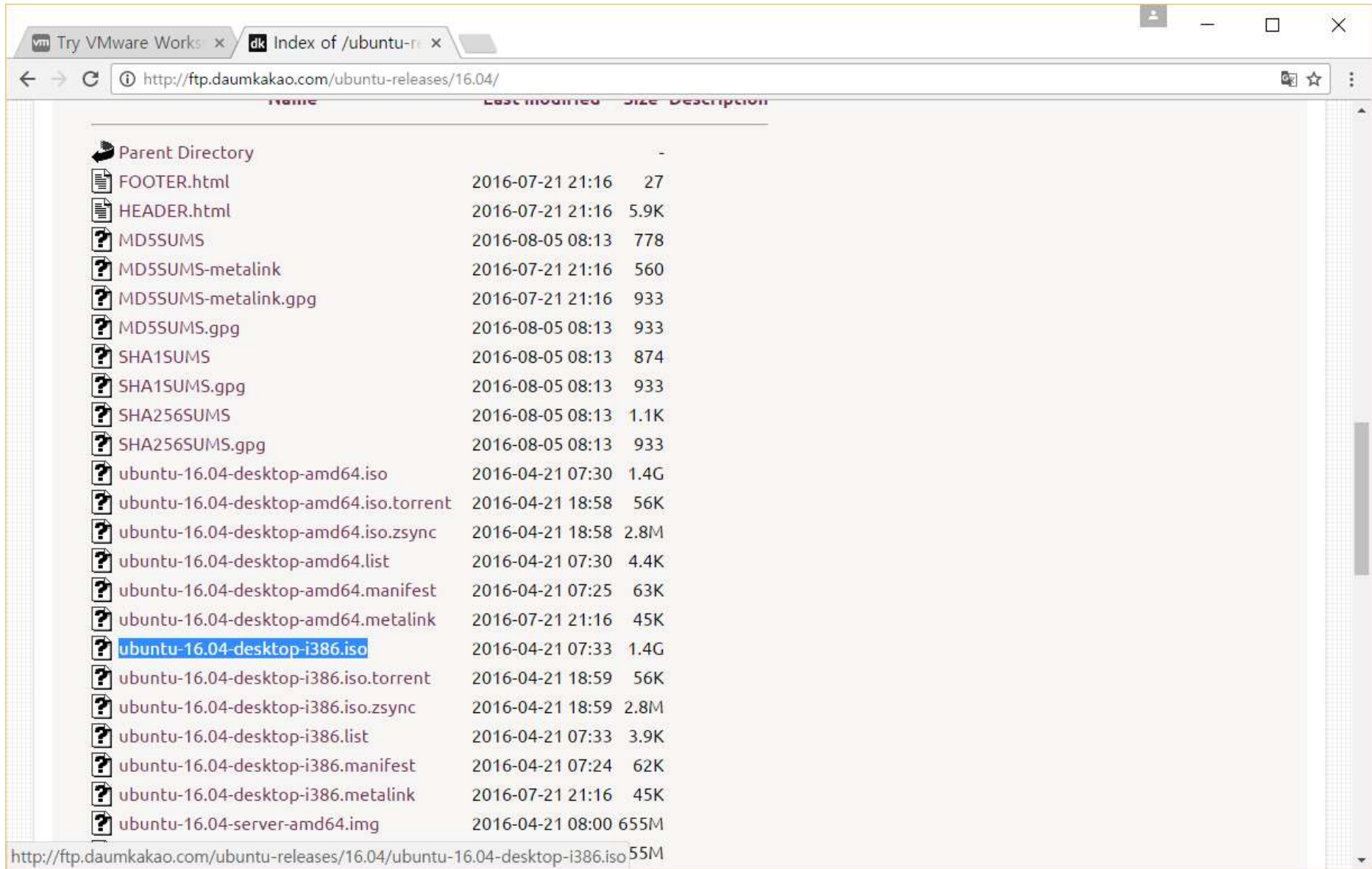
VMware Workstation 12.5 Player for Linux 64-bit  
Download Now »

CONTACT SALES GET SUPPORT ABOUT VMWARE CAREERS

© 2016 VMware, Inc Terms of Use Privacy Accessibility Site Index Trademarks Help Feedback

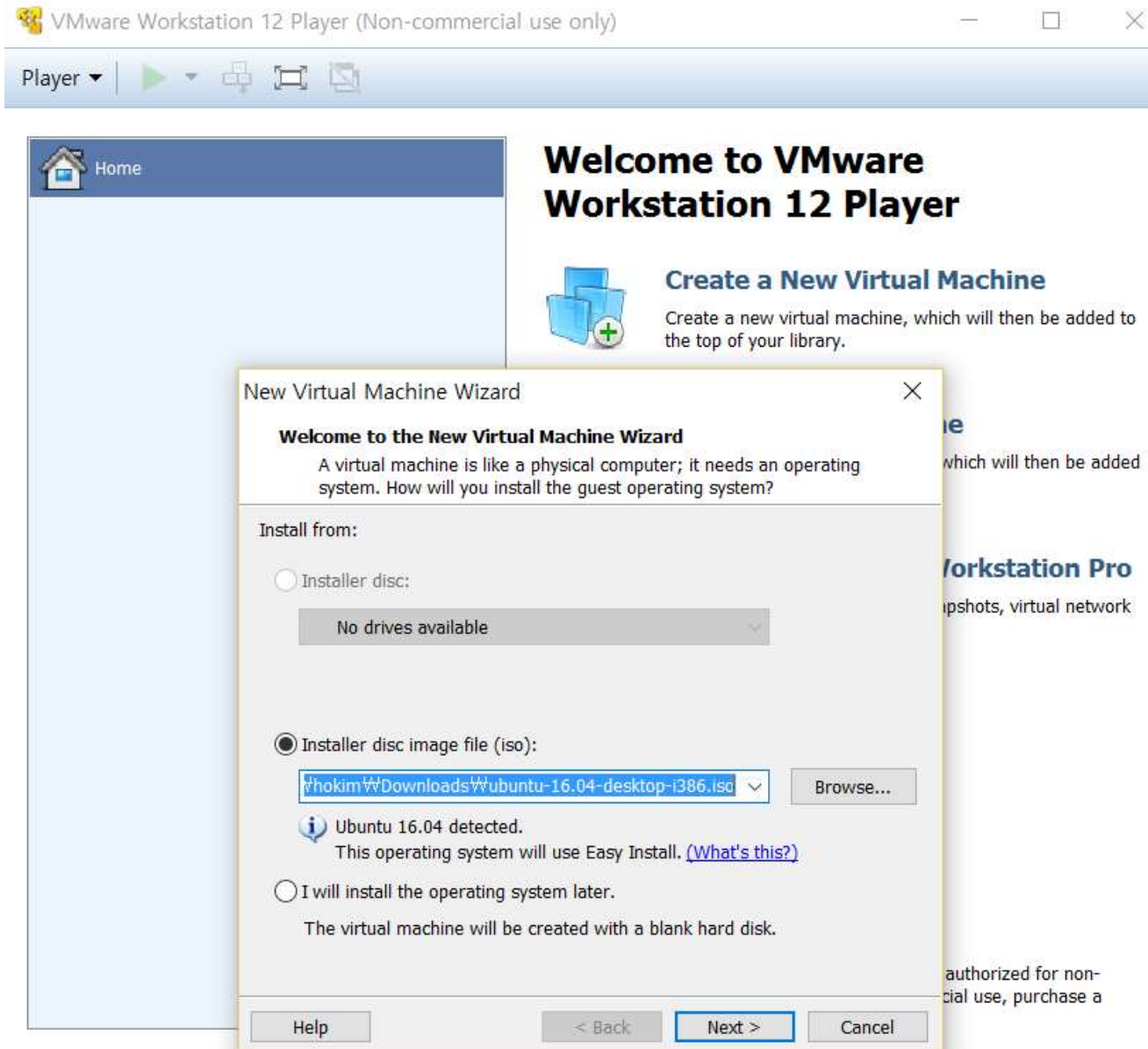


## ■ Ubuntu on VMware-player

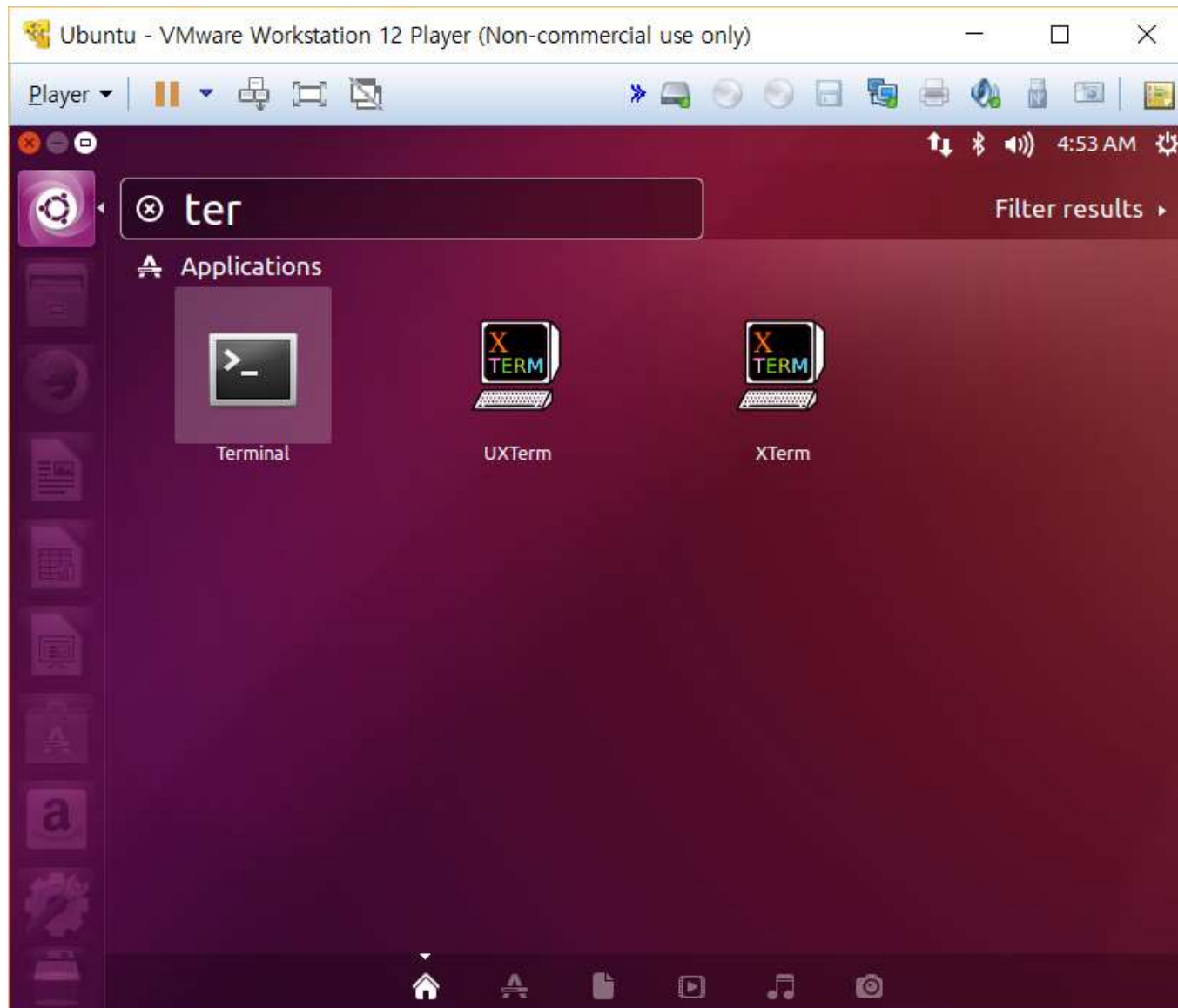




## ■ Ubuntu on VMware-player

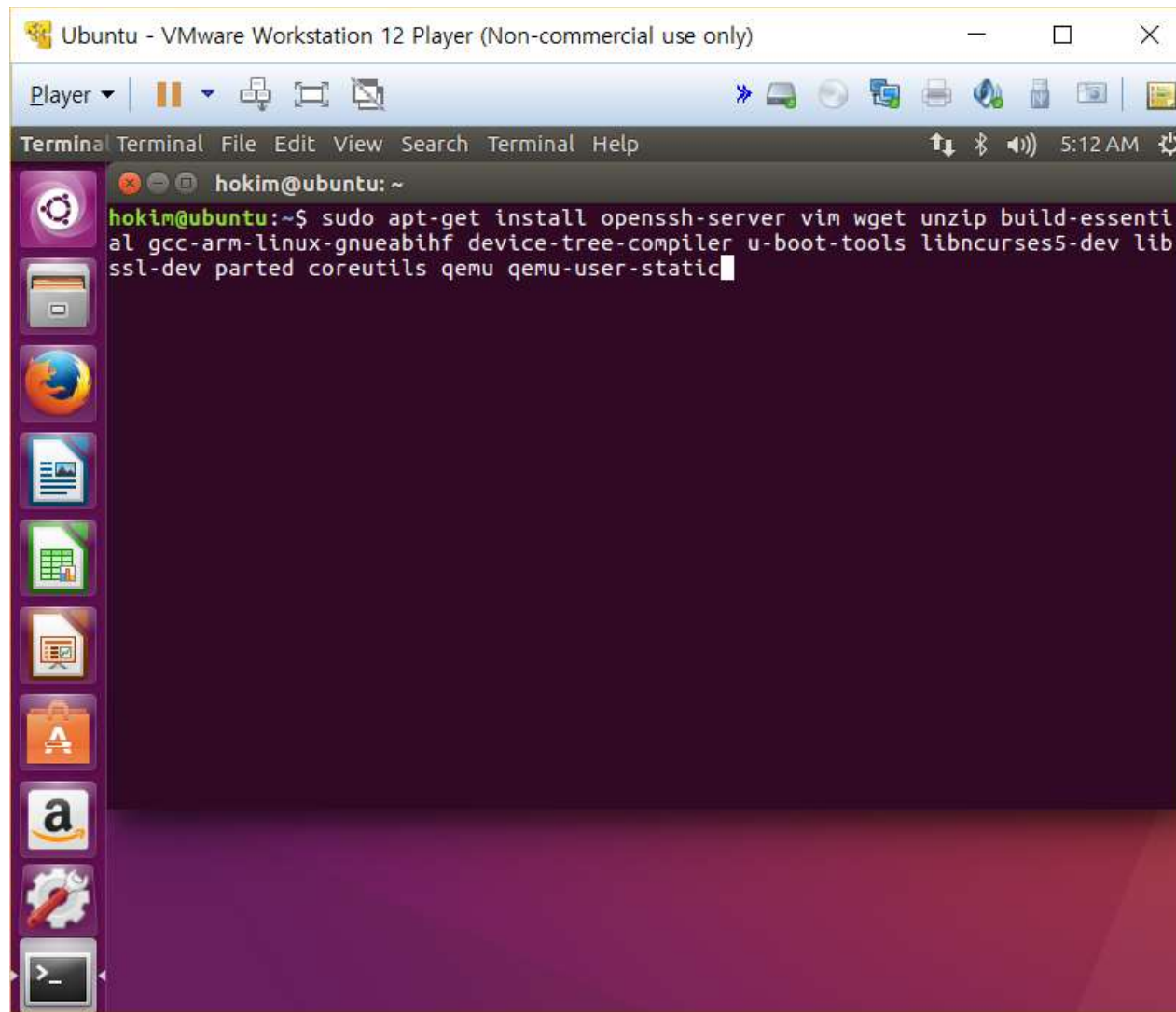


## ■ Ubuntu on VMware-player



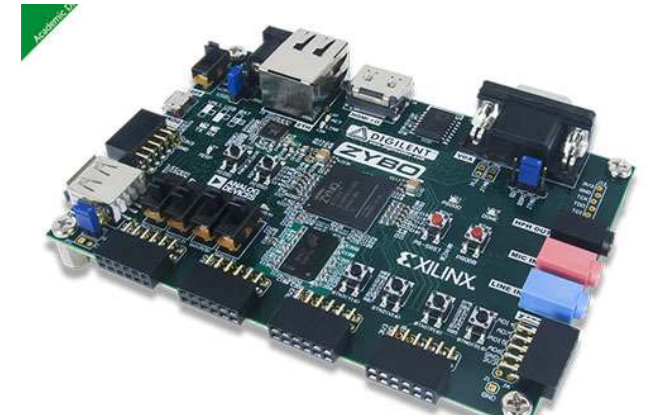
## ■ Ubuntu on VMware-player

```
$ sudo apt-get install openssh-server vim wget unzip build-essential gcc-arm-linux-gnueabi device-tree-compiler u-boot-tools libncurses5-dev libssl-dev parted coreutils qemu qemu-user-static
```



## ■ Target Board Zybo(Zynq-7010)

Gigabit Ethernet, USB, SD, UART  
28,000 logic cells, 240 KB BRAM



## ■ Development Process

1. bit stream

2. FSBL

3. DT Generation

4. u-boot

5. DT Compile

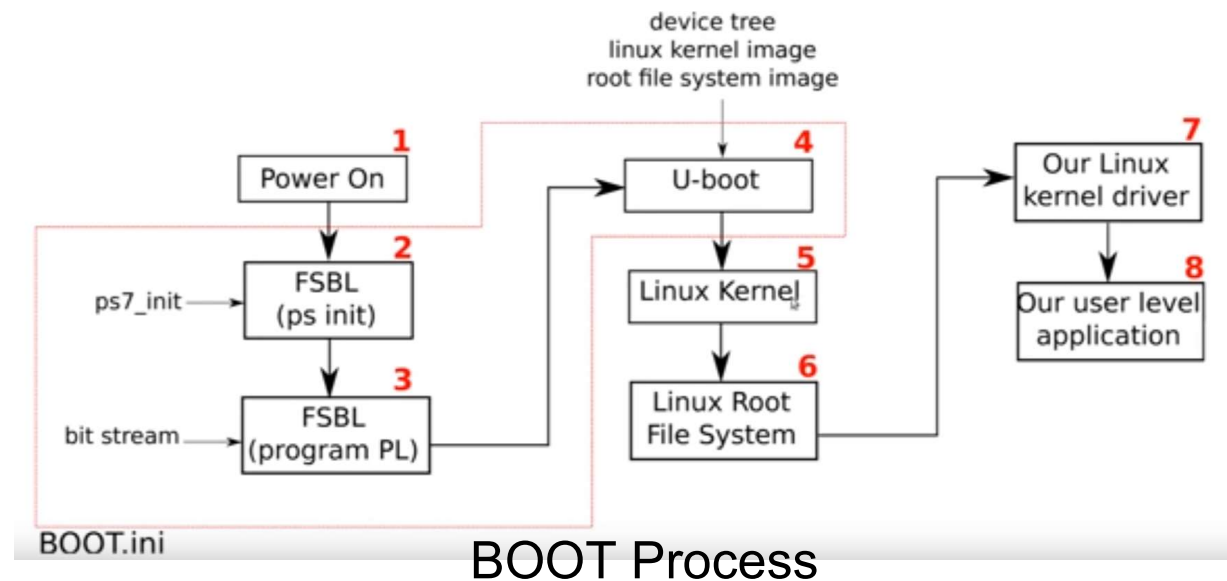
6. linux kernel image

7. root file system image

8. BOOT.bin(fsbl + bit + u-boot)

9. Update BOOT.bin / DT / kernel

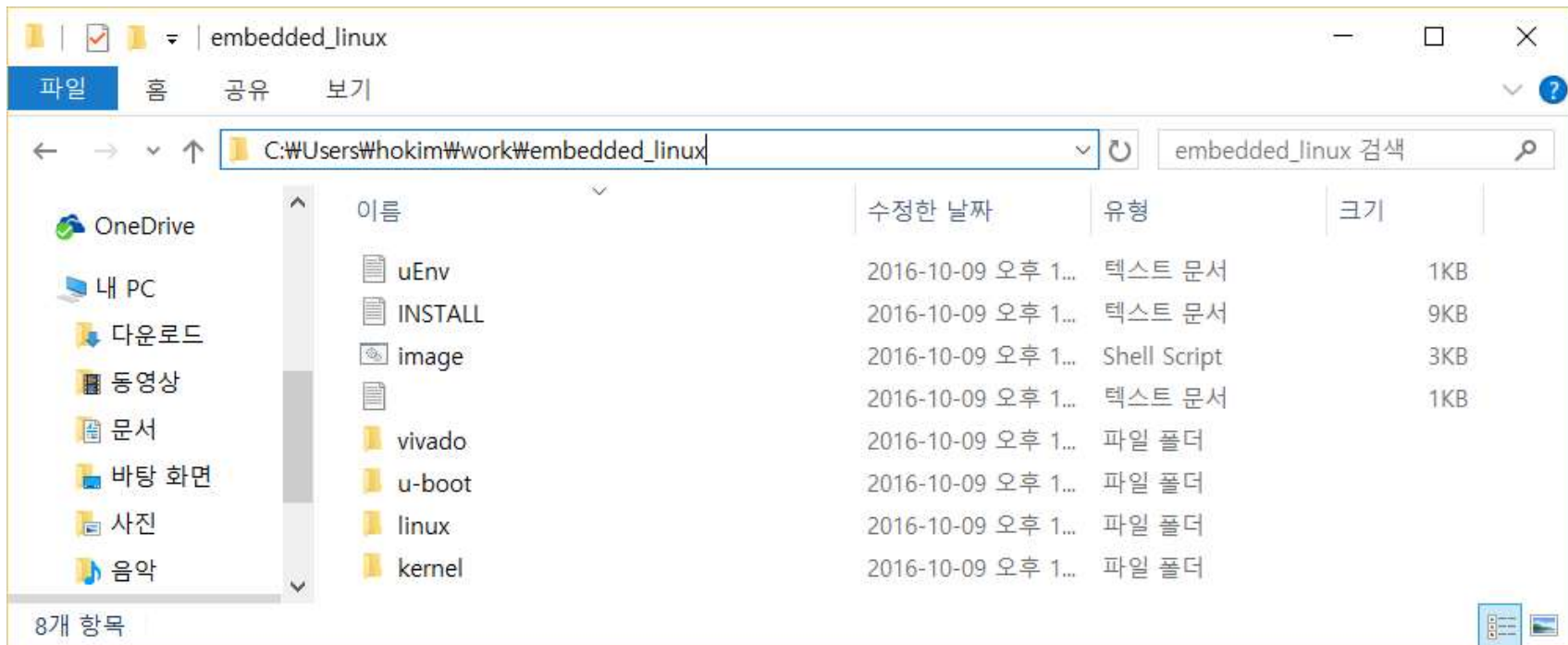
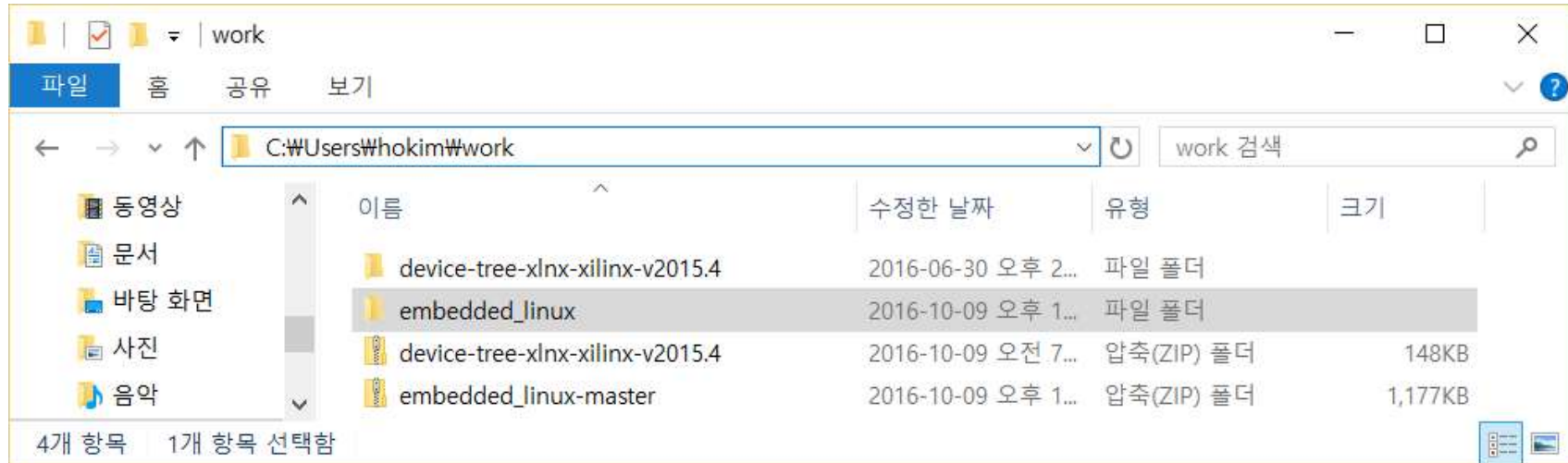
10. (user kernel driver)/user application





## ■ Source code

[https://github.com/inipro/embedded\\_linux](https://github.com/inipro/embedded_linux)





## ■ Vivado Design

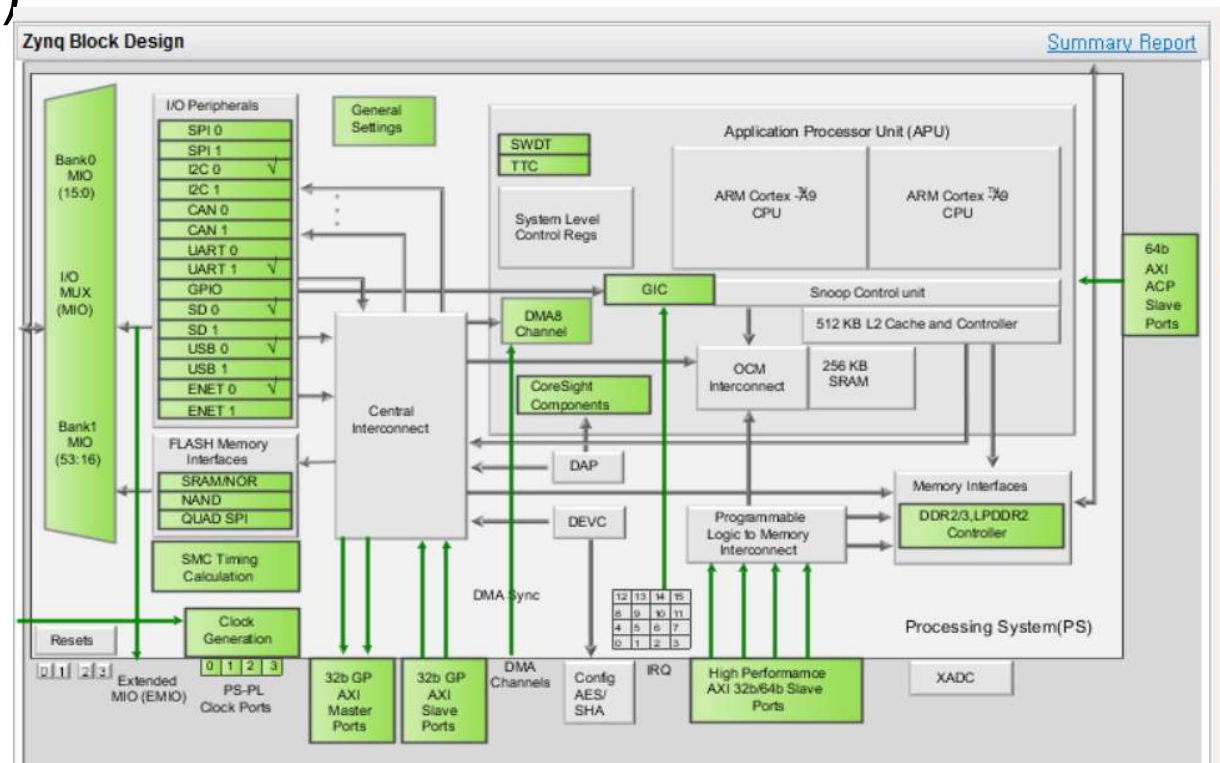
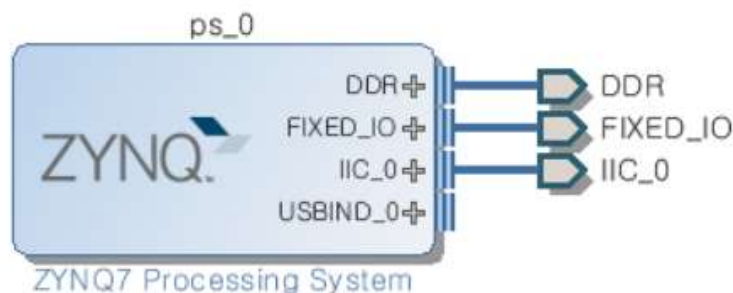
SD0 (boot files, root file system)

UART1 (console)

I2C0 (eeprom w/ Ethernet mac address; EMIO)

USB0 (usb host)

ENET0 (Gigabit Ethernet)



## *embedded\_linux.tcl*

```
set project_name embedded_linux
set part_name xc7z010clg400-1
#set ip_dir ip
set bd_path $project_name/$project_name.srcs/sources_1/bd/system
file delete -force $project_name
create_project -part $part_name $project_name $project_name
#set_property ip_repo_paths $ip_dir [current_project]
#update_ip_catalog
create_bd_design system

create_bd_cell -type ip -vlnv xilinx.com:ip:processing_system7:5.5 ps_0
source embedded_linux_preset.tcl
set_property -dict [apply_preset IPINST] [get_bd_cells ps_0]
apply_bd_automation -rule xilinx.com:bd_rule:processing_system7 -config {
 make_external {FIXED_IO, DDR}
} [get_bd_cells ps_0]
create_bd_intf_port -mode Master -vlnv xilinx.com:interface:iic_rtl:1.0 IIC_0
connect_bd_intf_net [get_bd_intf_pins ps_0/IIC_0] [get_bd_intf_ports IIC_0]

generate_target all [get_files $bd_path/system.bd]
make_wrapper -files [get_files $bd_path/system.bd] -top
add_files -norecurse $bd_path/hdl/system_wrapper.v
add_files -norecurse -fileset constrs_1 zybo.xdc
set_property verilog_define {TOOL_VIVADO} [current_fileset]
close_project
```

In cmd window for Vivado

```
C:\ cd C:\Users\hokim\work\embedded_linux\vivado
C:\ vivado -nolog -nojournal -mode batch -source embedded_linux.tcl
```

output : embedded\_linux\embedded\_linux.xpr...



# Bit Generation

The screenshot displays the Vivado 2015.4 IDE interface for a project named 'embedded\_linux'. The top menu bar includes File, Edit, Flow, Tools, Window, Layout, View, and Help. The toolbar shows various icons for file operations, simulation, and implementation. The 'Flow Navigator' on the left lists the project's workflow: Simulation, RTL Analysis, Synthesis, Implementation, and Program and Debug. The 'Generate Bitstream' option under 'Program and Debug' is highlighted with a red rectangle. The main workspace is divided into several panes. The 'Block Design' pane shows a hierarchical view of the design, including 'system', 'External Interfaces', 'Interface Connections', and 'ps\_0 (ZYNQ7 Processing System:5,5)'. The 'Diagram' pane displays a block diagram of the 'ps\_0' component, which is a ZYNQ7 Processing System. It shows connections for DDR, FIXED\_IO, IIC\_0, and USBIND\_0. The 'Tcl Console' pane at the bottom shows the following log messages:

```
Adding component instance block -- xilinx.com:ip:processing_system7:5.5 - ps_0
Successfully read diagram <system> from BD file <C:/Users/hokim/work/embedded_linux/vivado/embedded_linux/srcs/sources_1/bd/system/s
open_bd_design: Time (s): cpu = 00:00:14 ; elapsed = 00:00:11 . Memory (MB): peak = 877.746 ; gain = 106.148
```

At the bottom of the window, a status bar indicates: 'Generate a programming file after implementation'.

## *hwdef.tcl*

```
vivado -nolog -nojournal -mode batch -source hwdef.tcl

set project_name embedded_linux

open_project $project_name/$project_name.xpr

if {[get_property PROGRESS [get_runs synth_1]] != "100%"} {
 launch_runs synth_1
 wait_on_run synth_1
}

file delete -force $project_name/$project_name.hwdef

write_hwdef -force -file $project_name/$project_name.hwdef

close_project
```

```
C:\ vivado -nolog -nojournal -mode batch -source hwdef.tcl
```

```
output : embedded_linux\embedded_linux.hwdef
```



## *fsbl.tcl*

```
hsi -nolog -nojournal -mode batch -source fsbl.tcl

set project_name embedded_linux

set hard_path $project_name/$project_name.hard
set fsbl_path $project_name/$project_name.fsbl

file delete -force $hard_path $fsbl_path

file mkdir $hard_path
file copy -force $project_name/$project_name.hwdef $hard_path/$project_name.hdf

open_hw_design $hard_path/$project_name.hdf
create_sw_design -proc ps7_cortexa9_0 -os standalone fsbl

add_library xilffs
add_library xilrsa

generate_app -proc ps7_cortexa9_0 -app zynq_fsbl -dir $fsbl_path -compile

close_hw_design [current_hw_design]
```

```
C:\ hsi -nolog -nojournal -mode batch -source fsbl.tcl
```

output : embedded\_linux\embedded\_linux.fsbl\executable.elf





## *devicetree.tcl*

```
hsi -nolog -nojournal -mode batch -source devicetree.tcl

set project_name embedded_linux

set boot_args {console=ttyPS0,115200 root=/dev/mmcblk0p2 ro rootfstype=ext4 earlyprintk rootwait}

set hard_path $project_name/$project_name.hard
set tree_path $project_name/$project_name.tree

file delete -force $hard_path $tree_path

file mkdir $hard_path
file copy -force $project_name/$project_name.hwdef $hard_path/$project_name.hdf

set_repo_path $::env(HOME)/work/device-tree-xlnx-xilinx-v2015.4

open_hw_design $hard_path/$project_name.hdf
create_sw_design -proc ps7_cortexa9_0 -os device_tree devicetree

set_property CONFIG.kernel_version {2015.4} [get_os]
set_property CONFIG.bootargs $boot_args [get_os]

generate_bsp -dir $tree_path

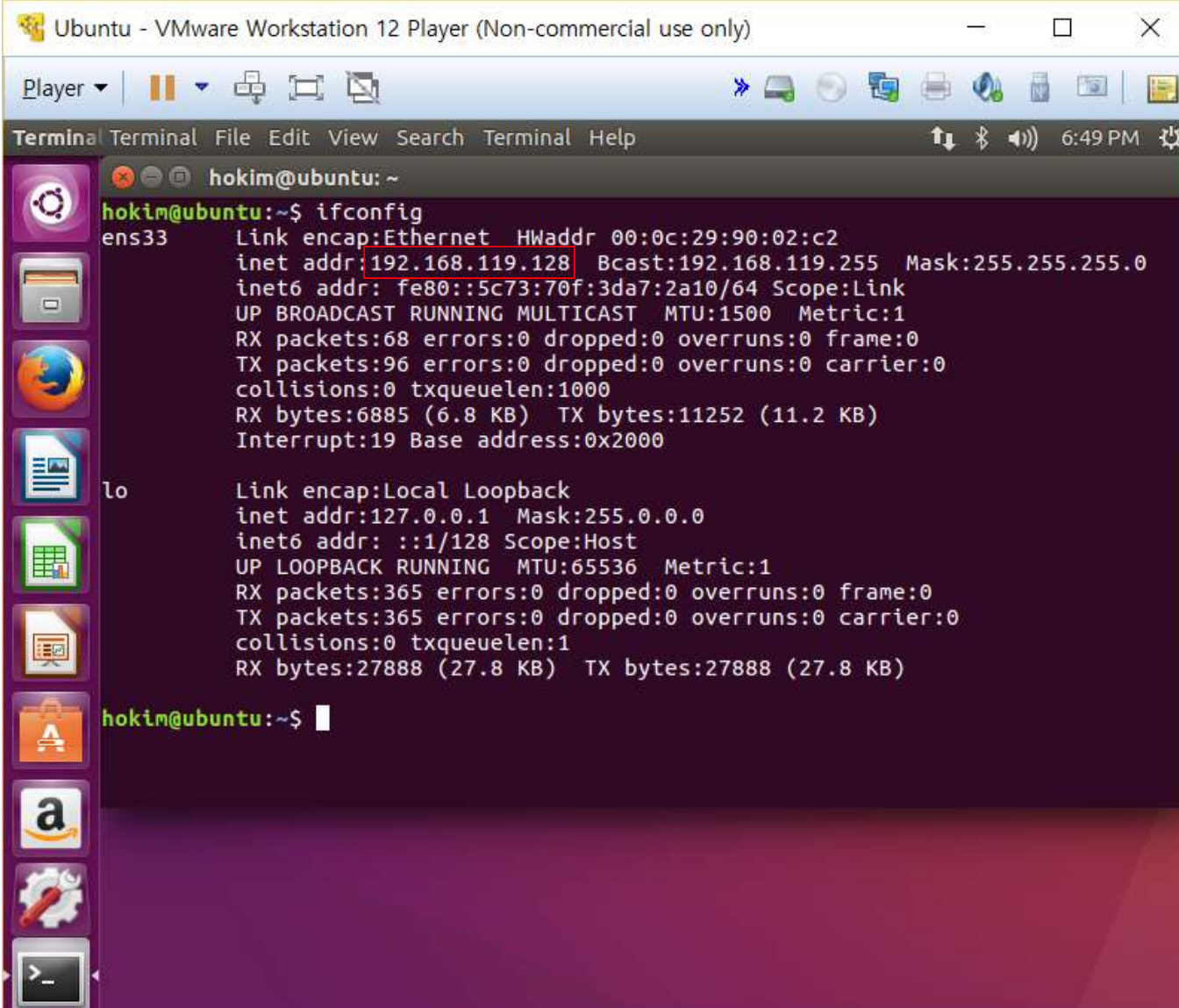
close_sw_design [current_sw_design]
close_hw_design [current_hw_design]
```

```
C:\ hsi -nolog -nojournal -mode batch -source devicetree.tcl
```

```
output : embedded_linux\embedded_linux.tree/*
```



## Ubuntu network IP

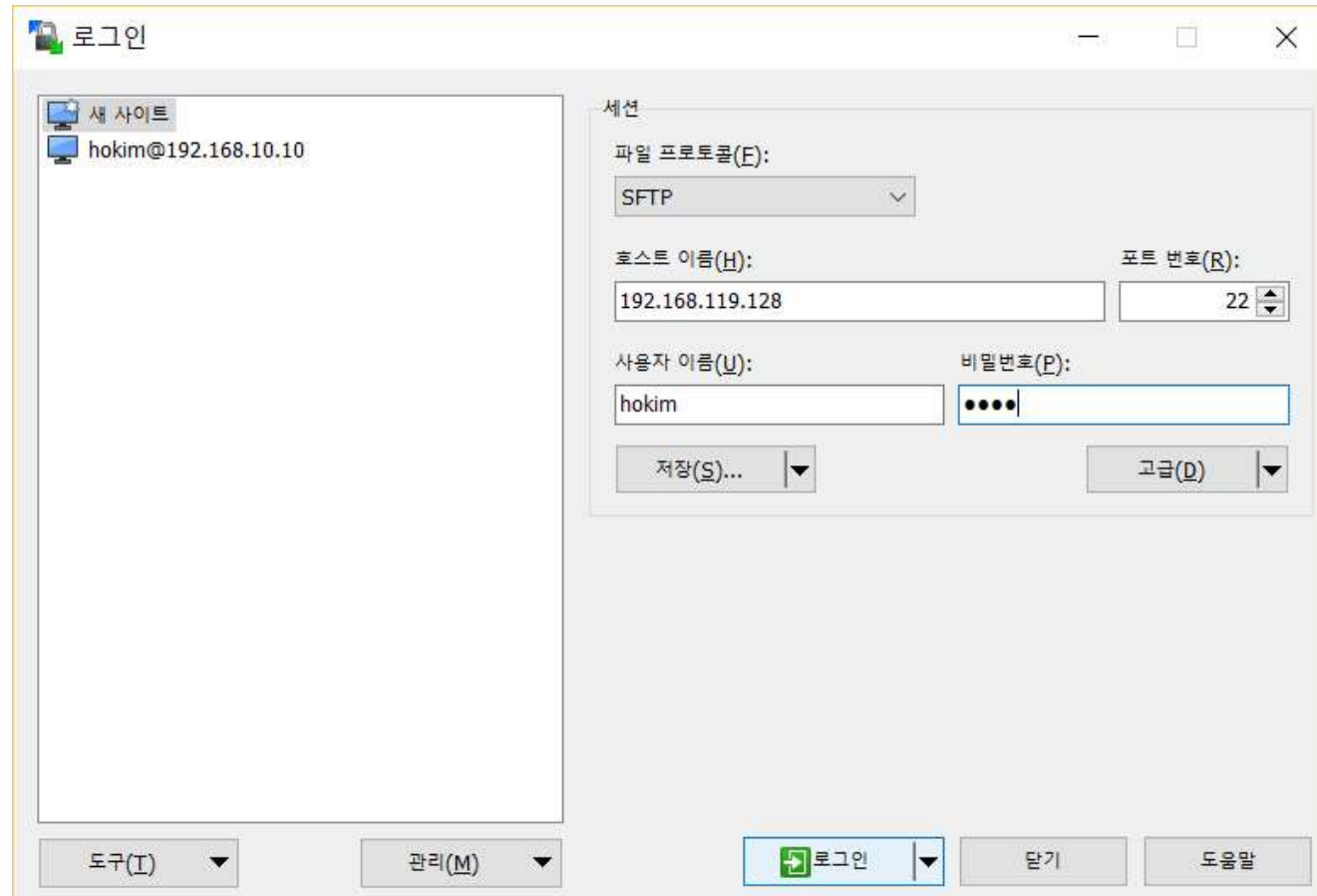


```
hokim@ubuntu: ~
hokim@ubuntu:~$ ifconfig
ens33 Link encap:Ethernet HWaddr 00:0c:29:90:02:c2
 inet addr:192.168.119.128 Bcast:192.168.119.255 Mask:255.255.255.0
 inet6 addr: fe80::5c73:70f:3da7:2a10/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:68 errors:0 dropped:0 overruns:0 frame:0
 TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:6885 (6.8 KB) TX bytes:11252 (11.2 KB)
 Interrupt:19 Base address:0x2000

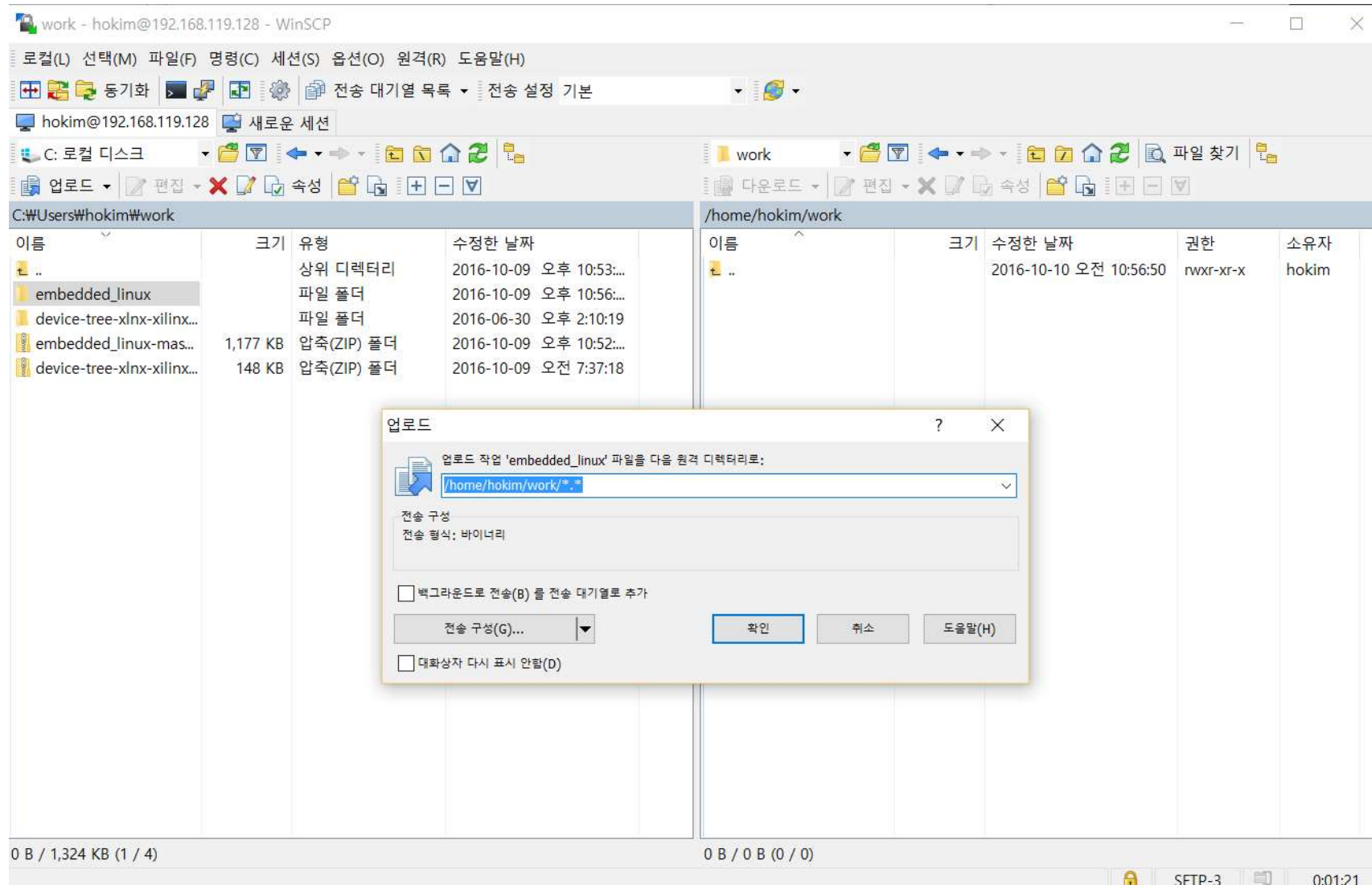
lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 inet6 addr: ::1/128 Scope:Host
 UP LOOPBACK RUNNING MTU:65536 Metric:1
 RX packets:365 errors:0 dropped:0 overruns:0 frame:0
 TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1
 RX bytes:27888 (27.8 KB) TX bytes:27888 (27.8 KB)

hokim@ubuntu:~$
```

embedded\_linux using WinSCP



## embedded\_linux using WinSCP



- **Boot loader for linux**

Load linux kernel(ulmage, devicetree.dtb) from SD to DRAM.

Set ethernet using mac address from i2c eeprom.

Boot ulmage.





In Terminal of Ubuntu

Can use gedit instead of vi

- : deletion

+ : insert

```
$ cd ~/work/embedded_linux
$ mkdir -p dl
$ wget https://github.com/Xilinx/u-boot-xlnx/archive/xilinx-
v2015.4.tar.gz -O dl/u-boot-xlnx-xilinx-v2015.4.tar.gz
$ cd u-boot
$ tar xvzf ../dl/u-boot-xlnx-xilinx-v2015.4.tar.gz
$ cd u-boot-xlnx-xilinx-v2015.4
$ vi board/xilinx/zynq/board.c
```



*board/Xilinx/zynq/board.c*

```
int board_late_init(void)
{
+ #if defined(CONFIG_ZYNQ_GEM_EEPROM_ADDR) && \
+ defined(CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET)
+ unsigned char enetaddr[6];
+
+ if (eeprom_read(CONFIG_ZYNQ_GEM_EEPROM_ADDR,
+ CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET,
+ enetaddr, ARRAY_SIZE(enetaddr)))
+ printf("I2C EEPROM MAC address read failed\n");
+ else
+ eth_setenv_enetaddr("ethaddr", enetaddr);
+ #endif
+
 switch ((zynq_slcr_get_boot_mode()) & ZYNQ_BM_MASK) {
 case ZYNQ_BM_QSPI:
```

**\$ vi common/main.c**

*common/main.c*

```
if (cli_process_fdt(&s))
 cli_secure_boot_cmd(s);

+ setenv("fdt_high", "0x1E000000");
+ setenv("sdboot", "echo Importing environment from SD... && mmcinfo && fatload mmc 0 0x2000000 uEnv.txt &&
env import -t 0x2000000 ${filesize} && boot");

autoboot_command(s);
```

```
$ cp ../zynq-zyboc.dts arch/arm/dts
$ cp ../zynq_zyboc.h include/configs
$ cp ../zynq_zyboc_defconfig configs
```

*arch/arm/dts/zynq-zyboc.dts*

```
/dts-v1/;
#include "zynq-7000.dtsi"

/ {
 model = "Zynq ZYBOC Development Board";
 compatible = "inipro,zynq-zyboc", "xlnx,zynq-7000";

 aliases {
 ethernet0 = &gem0;
 serial0 = &uart1;
 };

 memory {
 device_type = "memory";
 reg = <0x0 0x20000000>;
 };

 chosen {
 bootargs = "earlyprintk";
 linux,stdout-path = &uart1;
 stdout-path = &uart1;
 };
};
```

```
&clkc {
 ps-clk-frequency = <50000000>;
};

&gem0 {
 status = "okay";
 phy-mode = "rgmii-id";
 phy-handle = <ðernet_phy>;

 ethernet_phy: ethernet-phy@0 {
 reg = <0>;
 };
};

&sdhci0 {
 status = "okay";
};

&uart1 {
 status = "okay";
};
```

*include/configs/zynq\_zyboc.h*

```
#ifndef __CONFIG_ZYNQ_ZYBOC_H
#define __CONFIG_ZYNQ_ZYBOC_H

#define CONFIG_SYS_SDRAM_SIZE (512 * 1024 * 1024)

#define CONFIG_ZYNQ_SERIAL_UART1
#define CONFIG_ZYNQ_GEM0
#define CONFIG_ZYNQ_GEM_PHY_ADDR0 0

#define CONFIG_SYS_NO_FLASH

#define CONFIG_ZYNQ_SDHCI0
#define CONFIG_ZYNQ_I2C0
#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN 1
#define CONFIG_CMD_EEPROM
#define CONFIG_ZYNQ_GEM_EEPROM_ADDR 0x50
#define CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET 0xFA
#define CONFIG_ZYNQ_BOOT_FREEBSD

/* Define ZYBO PS Clock Frequency to 50MHz */
#define CONFIG_ZYNQ_PS_CLK_FREQ 50000000UL

#include <configs/zynq-common.h>

#endif /* __CONFIG_ZYNQ_ZYBOC_H */
```

*include/configs/zynq\_zyboc.h*

```
CONFIG_ARM=y
CONFIG_ARCH_ZYNQ=y
CONFIG_TARGET_ZYNQ_ZYBOC=y
CONFIG_DEFAULT_DEVICE_TREE="zynq-zyboc"
CONFIG_SYS_MALLOC_F is not set
CONFIG_SPL=y
CONFIG_FIT=y
CONFIG_FIT_VERBOSE=y
CONFIG_FIT_SIGNATURE=y
CONFIG_CMD_IMLS is not set
CONFIG_CMD_FLASH is not set
CONFIG_CMD_SETEXPR is not set
CONFIG_OF_EMBED=y
```

**\$ vi arch/arm/dts/Makefile**

```
dtb-$(CONFIG_ARCH_ZYNQ) += zynq-zc702.dtb \
zynq-zc706.dtb \
zynq-zed.dtb \
zynq-zybo.dtb \
zynq-microzed.dtb \
zynq-cc108.dtb \
zynq-afx-nand.dtb \
zynq-afx-nor.dtb \
zynq-afx-qspi.dtb \
zynq-cse-nand.dtb \
zynq-cse-nor.dtb \
zynq-cse-qspi.dtb \
zynq-picozed.dtb \
zynq-zc770-xm010.dtb \
zynq-zc770-xm011.dtb \
zynq-zc770-xm012.dtb \
- zynq-zc770-xm013.dtb
+ zynq-zc770-xm013.dtb \
+ zynq-zyboc.dtb
```





```
$ vi arch/arm/mach-zynq/Kconfig
```

```
config TARGET_ZYNQ_ZYBO
 bool "Zynq Zybo Board"
 select ZYNQ_CUSTOM_INIT

config TARGET_ZYNQ_AFX
 bool "Zynq AFX Board"
 select ZYNQ_CUSTOM_INIT

config TARGET_ZYNQ_CSE
 bool "Zynq CSE Board"
 select ZYNQ_CUSTOM_INIT

config TARGET_ZYNQ_CC108
 bool "Zynq CC108 Board"
 select ZYNQ_CUSTOM_INIT

+config TARGET_ZYNQ_ZYBOC
+ bool "Zynq ZyboC Board"
+ select ZYNQ_CUSTOM_INIT
+
+.....
+
config SYS_CONFIG_NAME
 default "zynq_zed" if TARGET_ZYNQ_ZED
 default "zynq_microzed" if TARGET_ZYNQ_MICROZED
 default "zynq_picozed" if TARGET_ZYNQ_PICOZED
 default "zynq_zc70x" if TARGET_ZYNQ_ZC702 || TARGET_ZYNQ_ZC706 \
 || TARGET_ZYNQ_ZC70X
 default "zynq_zc770" if TARGET_ZYNQ_ZC770
 default "zynq_zybo" if TARGET_ZYNQ_ZYBO
 default "zynq_cse" if TARGET_ZYNQ_CSE
 default "zynq_afx" if TARGET_ZYNQ_AFX
 default "zynq_cc108" if TARGET_ZYNQ_CC108
+ default "zynq_zyboc" if TARGET_ZYNQ_ZYBOC
+

endif
```



```
$ make arch=ARM zynq_zyboc_defconfig
$ make arch=ARM CROSS_COMPILE=arm-linux-gnueabihf- CFLAGS="-O2
-mtune=cortex-a9 -mfpu=neon -mfloat-abi=hard" all
$ cp u-boot ~/work/embedded_linux/vivado/u-boot.elf
```



- **Kernel driver** for platform devices

= Kernel source code + Device Tree

Hierarchy of Devices

Provide register address, irq number, so on as property of device.

- **Platform device**

Instead of being dynamically detected, must be statically described in either:

Kernel source code or Device Tree

The devices on I2C buses or SPI buses, or the devices directly part of the system-on-chip.



```
$ cd ~/work/embedded_linux
$ cp vivado/embedded_linux/embedded_linux.tree/system.dts .
$ vi system.dts
```

## *system.dts*

```
&clkc {
 fclk-enable = <0x0>;
 ps-clk-frequency = <50000000>;
};

+&gem0 {
+ phy-handle = <&phy0>;
+ ps7_ethernet_0_mdio: mdio {
+ #address-cells = <0x1>;
+ #size-cells = <0x0>;
+ phy0: phy@0 {
+ compatible = "realtek,RTL8211E";
+ device_type = "ethernet-phy";
+ reg = <0>;
+ };
+ };
+};
```

```
+&i2c0 {
+ eeprom@50 {
+ /* Microchip 24AA02E48 */
+ compatible = "microchip,24c02";
+ reg = <0x50>;
+ pagesize = <8>;
+ };
+};
+/{
+ usb_phy0: phy0 {
+ compatible = "ulpi-phy";
+ #phy-cells = <0>;
+ reg = <0xe0002000 0x1000>;
+ view-port = <0x0170>;
+ drv-vbus;
+ };
+};
+&usb0 {
+ usb-phy = <&usb_phy0>;
+};
```

```
$ dtc -O dtb -I dts -i vivado/embedded_linux/embedded_linux.tree/ -o
devicetree.dtb system.dts
```

```
$ cd ~/work/embedded_linux
$ wget https://github.com/Xilinx/linux-xlnx/archive/xilinx-v2015.4.01.tar.gz -O
dl/linux-xlnx-xilinx-v2015.4.01.tar.gz
$ cd kernel
$ tar xvzf ../dl/linux-xlnx-xilinx-v2015.4.01.tar.gz
$ cd linux-xlnx-xilinx-v2015.4.01
$ cp -r ../drivers/rtl8192cu drivers/net/wireless/
$ vi drivers/net/wireless/Kconfig
```

*drivers/net/wireless/Kconfig*

```
source "drivers/net/wireless/mwifiex/Kconfig"
source "drivers/net/wireless/cw1200/Kconfig"
source "drivers/net/wireless/rsi/Kconfig"
+source "drivers/net/wireless/rtl8192cu/Kconfig"

endif # WLAN
```

```
$ vi drivers/net/wireless/Makefile
```

*drivers/net/wireless/Makefile*

```
obj-$(CONFIG_CW1200) += cw1200/
obj-$(CONFIG_RSI_91X) += rsi/
+obj-$(CONFIG_RTL8192CU) += rtl8192cu/
```

```
$ make ARCH=arm xilinx_zynq_defconfig
```

```
$ make ARCH=arm menuconfig
```





```
hokim@ubuntu: ~/work/embedded_linux/kernel/linux-xlnx-xilinx-v2015.4.01
.config - Linux/arm 4.0.0 Kernel Configuration
> Networking support > Wireless

Wireless
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 []

--- Wireless
[*] cfg80211 - wireless configuration API
[] nl80211 testmode command (NEW)
[] enable developer warnings (NEW)
[] cfg80211 regulatory debugging (NEW)
[] cfg80211 certification onus (NEW)
[*] enable powersave by default (NEW)
[] use statically compiled regulatory rules database (NEW)
[] cfg80211 wireless extensions compatibility (NEW)
[*] Generic IEEE 802.11 Networking Stack (mac80211)
+ (+)

<Select> < Exit > < Help > < Save > < Load >

hokim@ubuntu: ~/work/embedded_linux/kernel/linux-xlnx-xilinx-v2015.4.01
.config - Linux/arm 4.0.0 Kernel Configuration
> Device Drivers > Device Tree and Open Firmware support

Device Tree and Open Firmware support
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 []

[] Device Tree runtime unit tests
[*] Device Tree overlays

<Select> < Exit > < Help > < Save > < Load >
```



```
hokim@ubuntu: ~/work/embedded_linux/kernel/linux-xlnx-xilinx-v2015.4.01
.config - Linux/arm 4.0.0 Kernel Configuration
[...] rivers > Network device support > PHY Device support and infrastructure
PHY Device support and infrastructure
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 []

^(-)
< > Drivers for the Intel LXT PHYs
< > Drivers for the Cicada PHYs
<*> Drivers for the Vitesse PHYs
< > Drivers for SMSC PHYs
< > Drivers for Broadcom PHYs
< > Drivers for Broadcom 7xxx SOCs internal PHYs
< > Driver for Broadcom BCM8706 and BCM8727 PHYs
< > Drivers for ICPlus PHYs
<*> Drivers for Realtek PHYs
< > Drivers for National Semiconductor PHYs
+ (+)

<Select> < Exit > < Help > < Save > < Load >
```

```
hokim@ubuntu: ~/work/embedded_linux/kernel/linux-xlnx-xilinx-v2015.4.01
.config - Linux/arm 4.0.0 Kernel Configuration
> Device Drivers > Network device support > Wireless LAN
Wireless LAN
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 []

^(-)
< > Hermes chipset 802.11b support (Orinoco/Prism2/Symbol) (NEW)
< > Softmac Prism54 support (NEW)
< > Ralink driver support (NEW) ----
< > Realtek rtlwifi family of devices ----
[] TI Wireless LAN support ----
< > ZyDAS ZD1211/ZD1211B USB-wireless support (NEW)
< > Marvell WiFi-Ex Driver (NEW)
< > CW1200 WLAN support (NEW)
< > Redpine Signals Inc 91x WLAN driver support (NEW)
<*> Realtek 8192C USB WiFi

<Select> < Exit > < Help > < Save > < Load >
```





```
hokim@ubuntu: ~/work/embedded_linux/kernel/linux-xlnx-xilinx-v2015.4.01
.config - Linux/arm 4.0.0 Kernel Configuration
> Device Drivers > SPI support

SPI support
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 []

^(-)
< > NXP SC18IS602/602B/603 I2C to SPI bridge
< > Analog Devices AD-FMCOMMS1-EBZ SPI-I2C-bridge driver
<*> Xilinx SPI controller common module
<*> Xilinx Zynq QSPI controller
[] Xilinx Zynq QSPI Dual stacked configuration
< > Xilinx ZynqMP QSPI controller
< > DesignWare SPI controller core support
*** SPI Protocol Masters ***
<*> User mode SPI device driver support
< > Infineon TLE62X0 (for power switching)

<Select> < Exit > < Help > < Save > < Load >
```

```
hokim@ubuntu: ~/work/embedded_linux/kernel/linux-xlnx-xilinx-v2015.4.01
.config - Linux/arm 4.0.0 Kernel Configuration
> Device Drivers > GPIO Support

GPIO Support
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 []

^(-)
< > GPIO driver for 74xx-ICs with MMIO access
-* Generic memory-mapped GPIO controller support (MMIO platfor
< > Synopsys DesignWare APB GPIO driver
< > Emma Mobile GPIO
[] LSI ZEVIO SoC memory mapped GPIOs
[] PrimeCell PL061 GPIO support
< > SMSC SCH311x SuperI/O GPIO
< > GPIO based on SYSCON
<*> Xilinx GPIO support
<*> Xilinx Zynq GPIO support
^(+)
```

```
<Select> < Exit > < Help > < Save > < Load >
```



```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- CFLAGS="-O2
-mtune=cortex-a9 -mfpu=neon -mfloat-abi=hard" -j $(nproc)
UIMAGE_LOADADDR=0x8000 ulmage
$ cp arch/arm/boot/ulmage ../..
```



```
$ cd ~/work/embedded_linux
$ sudo sh ./image.sh
```

## *image.sh*

```
mkdir -p dl

UBUNTU_URL=http://cdimage.ubuntu.com/ubuntu-base/releases/16.04/release
UBUNTU=ubuntu-base-16.04-core-armhf.tar.gz
if [! -f dl/$UBUNTU]; then
 wget $UBUNTU_URL/$UBUNTU -O dl/$UBUNTU
fi

ARCH=armhf
SIZE=3500
mkdir -p img
IMAGE=img/ubuntu-core_${ARCH}_16.04.img
dd if=/dev/zero of=$IMAGE bs=1M count=$SIZE
DEVICE=$(losetup -f)
losetup $DEVICE $IMAGE
parted -s $DEVICE mklabel msdos
parted -s $DEVICE mkpart primary fat16 4MB 128MB
parted -s $DEVICE mkpart primary ext4 128MB 100%
BOOT_DEV=/dev/$(lsblk -lno NAME $DEVICE | sed '2!d')
ROOT_DEV=/dev/$(lsblk -lno NAME $DEVICE | sed '3!d')
mkfs.vfat -v $BOOT_DEV
mkfs.ext4 -F -j $ROOT_DEV
ROOT_DIR=root
mkdir -p $ROOT_DIR
mount $ROOT_DEV $ROOT_DIR
cd $ROOT_DIR
tar xvf ../dl/$UBUNTU
rm -fr boot
cd ..
```





## *image.sh*

```
cat > $ROOT_DIR/etc/fstab << EOF_CAT
/etc/fstab: static file system information.
<file system> <mount point> <type> <options> <dump> <pass>
/dev/mmcb1k0p1 /boot vfat errors=remount-ro 0 0
/dev/mmcb1k0p2 / ext4 errors=remount-ro 0 1
EOF_CAT

cp /etc/resolv.conf $ROOT_DIR/etc/
cp /usr/bin/qemu-arm-static $ROOT_DIR/usr/bin/
chroot $ROOT_DIR << EOF_CHROOT
sed -i 's/^# deb http://ports.ubuntu.com/ubuntu-ports/ xenial universe.*/deb http://ports.ubuntu.com/ubuntu-ports/
xenial universe/' /etc/apt/sources.list
sed -i 's/^# deb http://ports.ubuntu.com/ubuntu-ports/ xenial-updates universe.*/deb http://ports.ubuntu.com/ubuntu-
ports/ xenial-updates universe/' /etc/apt/sources.list
apt-get update
apt-get -y upgrade
DEBIAN_FRONTEND=noninteractive apt-get -y install vim sudo openssh-server udev usbutils u-boot-tools device-tree-
compiler kmod net-tools wpasupplicant parted rfkill lshw wireless-tools gcc g++ cmake git i2c-tools iputils-ping
echo "Asia/Seoul" > /etc/timezone
ln -fs /usr/share/zoneinfo/Asia/Seoul /etc/localtime
locale-gen "en_US.UTF-8"
DEBIAN_FRONTEND=noninteractive dpkg-reconfigure locales
EOF_CHROOT
rm $ROOT_DIR/etc/resolv.conf
rm $ROOT_DIR/usr/bin/qemu-arm-static

mkdir -pv $ROOT_DIR/etc/systemd/system/serial-getty\@ttyPS0.service.d
cat > $ROOT_DIR/etc/systemd/system/serial-getty\@ttyPS0.service.d/autologin.conf << EOF_CAT
[Service]
ExecStart=
ExecStart=-/sbin/agetty --autologin root -s %I 115200,38400,9600 linux
EOF_CAT
umount -l $ROOT_DIR
rmdir $ROOT_DIR
losetup -d $DEVICE
```

using WinSCP

| Ubuntu                                                    | Windows                                    |
|-----------------------------------------------------------|--------------------------------------------|
| ~/work/embedded_linux/<br>devicetree.dtb, ulmage          | C:\Users\Whokim\work\embedded_linux        |
| ~/work/embedded_linux/vivado/<br>u-boot.elf               | C:\Users\Whokim\work\embedded_linux\vivado |
| ~/work/embedded_linux/img/<br>ubuntu-core_armhf_16.04.img | C:\Users\Whokim\work                       |



## *bootbin.tcl*

```
tclsh bootbin.tcl

set project_name embedded_linux

set fileId [open $project_name/boot.bif "w"]
puts $fileId "img: {[bootloader\] $project_name/$project_name.fsbl/executable.elf
$project_name/$project_name.runs/impl_1/system_wrapper.bit u-boot.elf}"
close $fileId

file delete -force boot.bin

exec bootgen -image $project_name/boot.bif -w -o i boot.bin >&@stdout
```

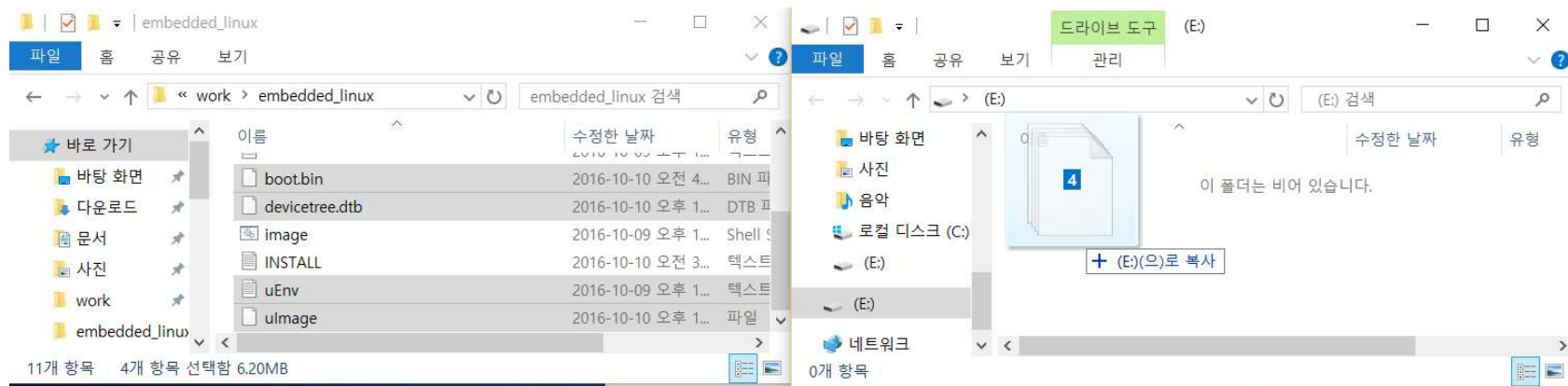
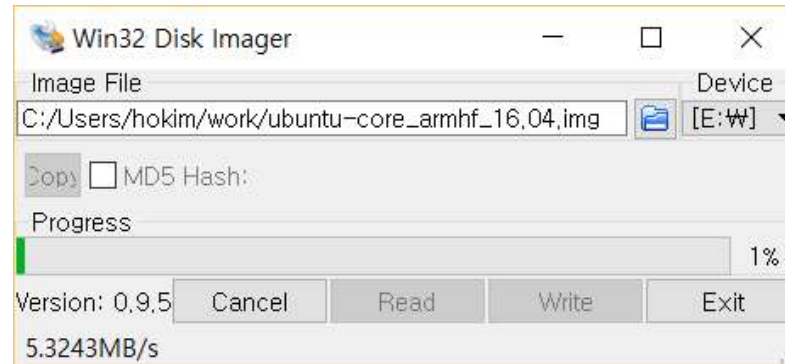
In cmd window for Vivado

```
C:\ cd c:\Users\hokim\work\embedded_linux\vivado
C:\ tclsh bootbin.tcl
C:\ copy boot.bin ..
```

output : boot.bin



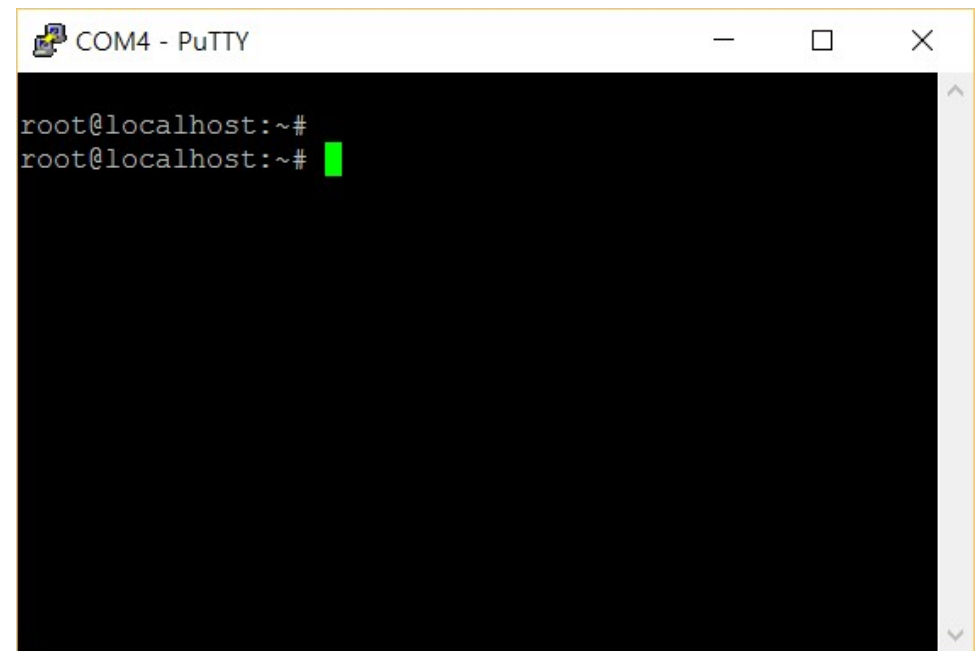
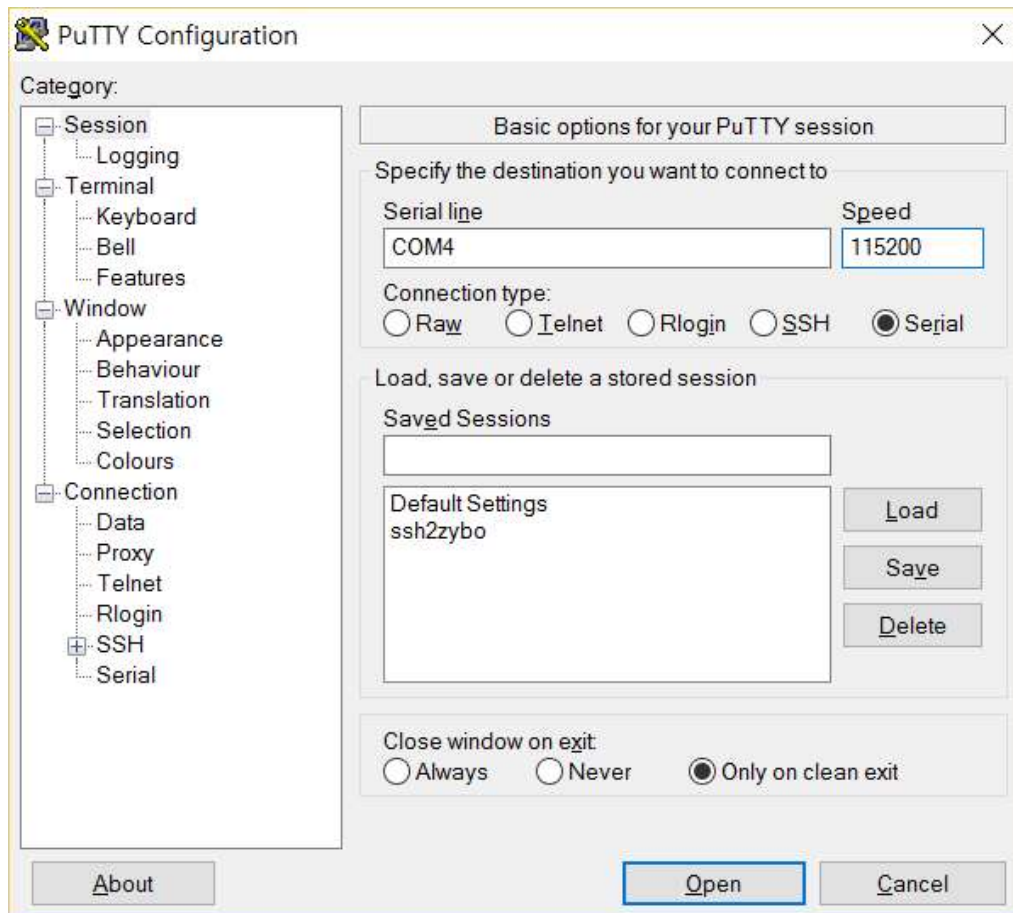
<https://sourceforge.net/projects/win32diskimager/>



Boot zybo with SD card



Log in through UART console using putty





```
groupadd -g 1000 hokim
groupadd -g 1001 admin
useradd -u 1000 -g 1000 -G adm,dialout,cdrom,audio,dip,video,plugdev,admin
-d /home/hokim -m -s /bin/bash hokim
passwd hokim
vi /etc/network/interfaces.d/eth0
```

*/etc/network/interfaces.d/eth0*

```
allow-hotplug eth0
iface eth0 inet static
address 192.168.10.10
netmask 255.255.255.0
```

```
halt
```

Turn off/on zybo



## Log in through ethernet using putty

인터넷 프로토콜 버전 4(TCP/IPv4) 속성

일반

네트워크가 IP 자동 설정 기능을 지원하면 IP 설정이 자동으로 할당되도록 할 수 있습니다. 지원하지 않으면, 네트워크 관리자에게 적절한 IP 설정값을 문의해야 합니다.

☐ 자동으로 IP 주소 받기(O)

☒ 다음 IP 주소 사용(S):

IP 주소(I): 192 . 168 . 10 . 100

서브넷 마스크(U): 255 . 255 . 255 . 0

기본 게이트웨이(D): . . .

☐ 자동으로 DNS 서버 주소 받기(B)

☒ 다음 DNS 서버 주소 사용(E):

기본 설정 DNS 서버(P): . . .

보조 DNS 서버(A): . . .

☐ 끝날 때 설정 유효성 검사(L)

고급(V)...

확인 취소

PuTTY Configuration

Category:

- Session
- Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - Telnet
  - Rlogin
  - SSH
  - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) 192.168.10.10 Port 22

Connection type:  
☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Load Save Delete

Close window on exit:  
☐ Always ☐ Never ☒ Only on clean exit

About Open Cancel

```
$ sudo vi /etc/hostname
```

*/etc/hostname*

```
-localhost.localdomain
+zybo
```

```
$ sudo vi /etc/hosts
```

*/etc/hosts*

```
127.0.0.1 localhost
127.0.1.1 zybo

The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
$ ls /sys/class/net
```

*output:*

```
enx74da38422193 eth0 lo
```



```
$ sudo vi /etc/network/interfaces.d/enx74da38422193
```

*/etc/network/interfaces.d/enx74da38422193*

```
allow-hotplug enx74da38422193
iface enx74da38422193 inet dhcp
 pre-up wpa_supplicant -B -D wext -i enx74da38422193 -c /etc/wpa_supplicant.conf
 post-down killall -q wpa_supplicant
 udhcpc_opts -t7 -T3
```

```
$ sudo vi /etc/wpa_supplicant.conf
```

*/etc/wpa\_supplicant.conf*

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
 ssid="INIPRO"
 key_mgmt=WPA-PSK
 psk="20471047"
}
```

```
$ sudo halt
```

Turn off/on zybo



```
$ sudo vi /etc/network/interfaces.d/enx74da38422193
```

*/etc/network/interfaces.d/enx74da38422193*

```
allow-hotplug enx74da38422193
iface enx74da38422193 inet dhcp
 pre-up wpa_supplicant -B -D wext -i enx74da38422193 -c /etc/wpa_supplicant.conf
 post-down killall -q wpa_supplicant
 udhcpc_opts -t7 -T3
```

```
$ sudo vi /etc/wpa_supplicant.conf
```

*/etc/wpa\_supplicant.conf*

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
 ssid="INIPRO"
 key_mgmt=WPA-PSK
 psk="20471047"
}
```

```
$ sudo -s
echo -e "d\n2\nw" | fdisk /dev/mmcblk0
parted -s /dev/mmcblk0 mkpart primary ext4 128M 100%
halt
```



Turn off / on zybo

Log in through ethernet using putty

```
$ sudo resize2fs /dev/mmcblk0p2
$ df -h
```

*output:*

| Filesystem     | Size | Used | Avail | Use% | Mounted on     |
|----------------|------|------|-------|------|----------------|
| /dev/root      | 7.0G | 632M | 6.0G  | 10%  | /              |
| devtmpfs       | 242M | 0    | 242M  | 0%   | /dev           |
| tmpfs          | 250M | 0    | 250M  | 0%   | /dev/shm       |
| tmpfs          | 250M | 6.5M | 244M  | 3%   | /run           |
| tmpfs          | 5.0M | 0    | 5.0M  | 0%   | /run/lock      |
| tmpfs          | 250M | 0    | 250M  | 0%   | /sys/fs/cgroup |
| /dev/mmcblk0p1 | 118M | 6.3M | 112M  | 6%   | /boot          |



```
$ ifconfig
```

*output:*

```
enx74da38422193 Link encap:Ethernet HWaddr 74:da:38:42:21:93
 inet addr:192.168.0.148 Bcast:192.168.0.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:418 errors:0 dropped:3 overruns:0 frame:0
 TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:77995 (77.9 KB) TX bytes:1772 (1.7 KB)

eth0 Link encap:Ethernet HWaddr d8:80:39:5c:48:82
 inet addr:192.168.10.10 Bcast:192.168.10.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:437 errors:0 dropped:1 overruns:0 frame:0
 TX packets:448 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:33219 (33.2 KB) TX bytes:57605 (57.6 KB)
 Interrupt:145 Base address:0xb000

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 UP LOOPBACK RUNNING MTU:65536 Metric:1
 RX packets:80 errors:0 dropped:0 overruns:0 frame:0
 TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:5920 (5.9 KB) TX bytes:5920 (5.9 KB)
```

