UEFI & EDK II Training

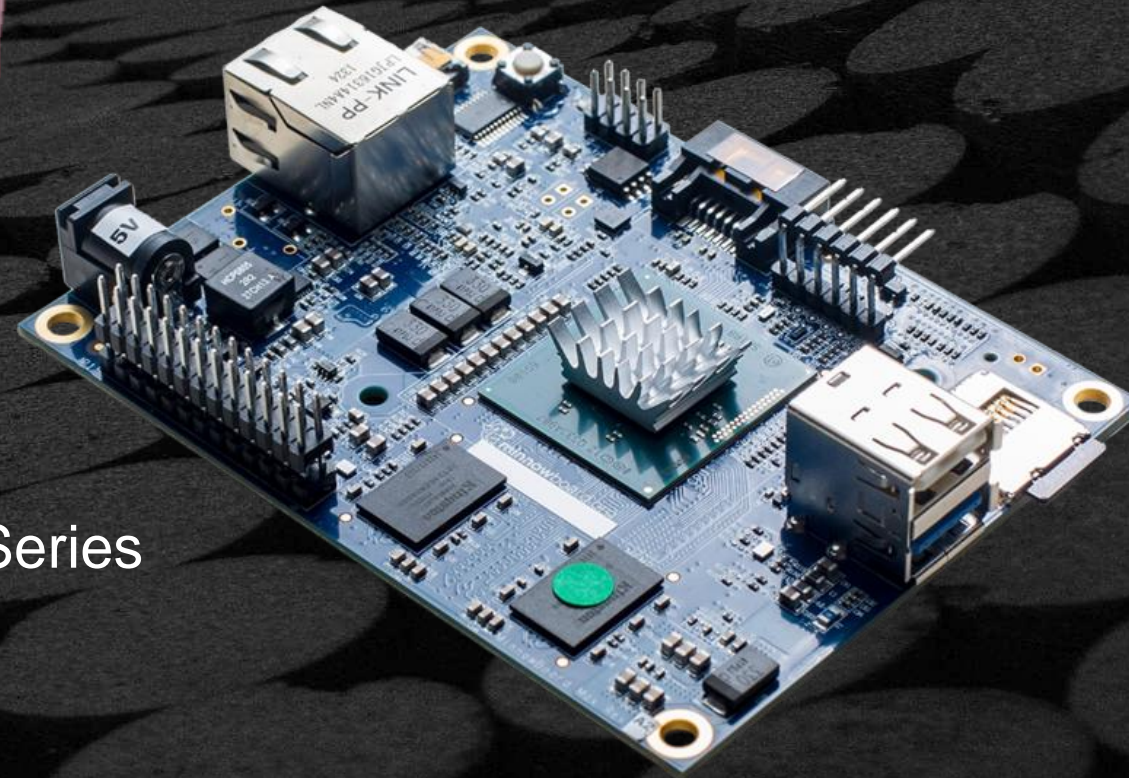## Platform Build Lab MinnowBoard Max - Linux

tianocore.org

PLATFORM BUILD LABS

Lab Setup and Build for MinnowBoard Max/Turbot

- Hardware Setup for MinnowBoard Max/Turbot
- Build a EDK II Platform using MinnowBoard Max/Turbot

PLATFORM HW SETUP
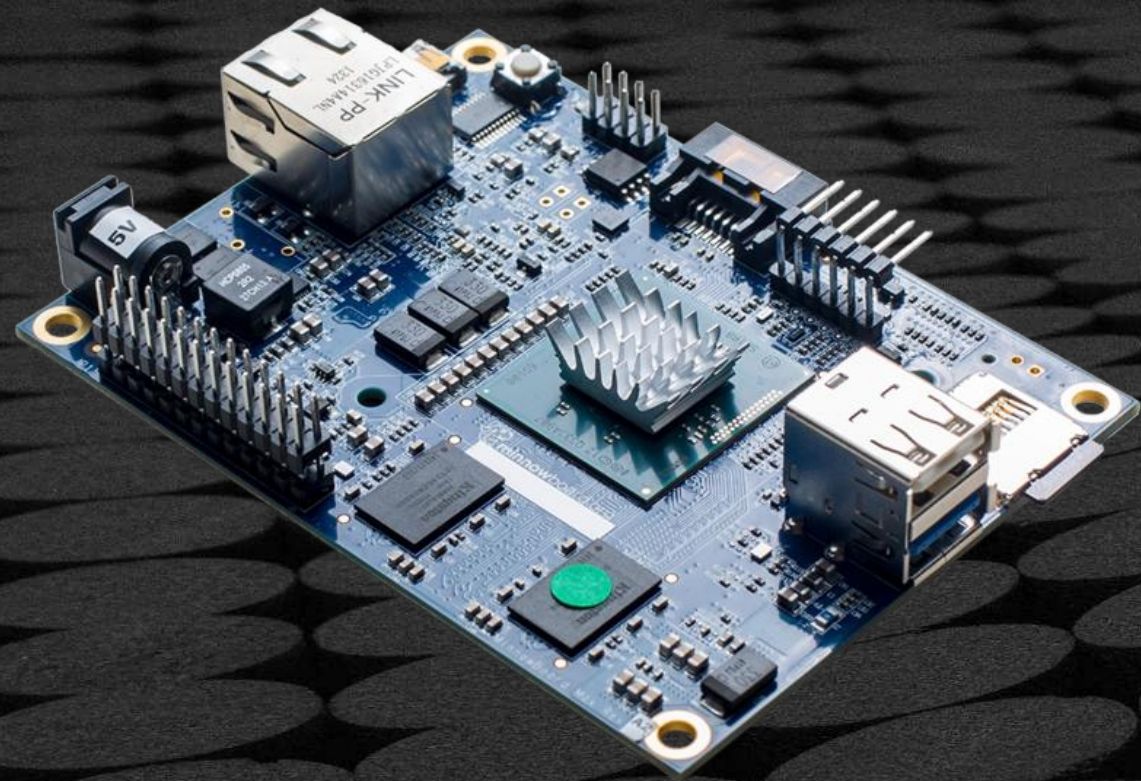
Setup hardware for the MinnowBoard Max/Turbot

EDK II Platform – MinnowBoard Max/Turbot

minnowboard.org

Turbot

Intel Atom® processor E3800 Series
(Formerly Bay Trail-I)

MinnowBoard Max/Turbot Workshop Lab Hardware

FTDI USB Cable

5V** Power Supply

USB thumb drive

**Warning do not use any other power supply than 5V or the board will Fry
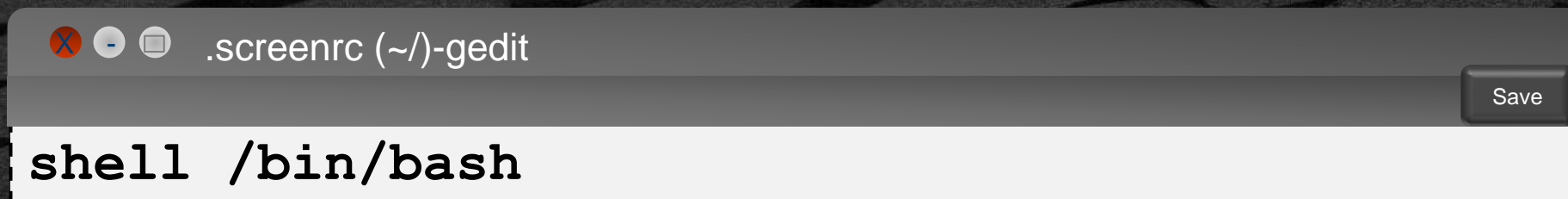
tianocore

Install "Screen" on Ubuntu

- skip for Clear Linux* Project

Terminal prompt (Cnt-Alt-T)

```
bash$ sudo apt-get install screen
bash$ cd $Home
bash$ gedit ~.screenrc
```

While in screen

Cnt-A then **D** goes back to Terminal

Inside the editor, type
"shell /bin/bash" then save

```
bash$ screen –r
(returns to screen)
```

.screenrc (~/)-gedit

Save

`shell /bin/bash`

type

Save

Skip if using Clear Linux* Project, Screen is part of install
There may be other serial terminal applications that are supported.
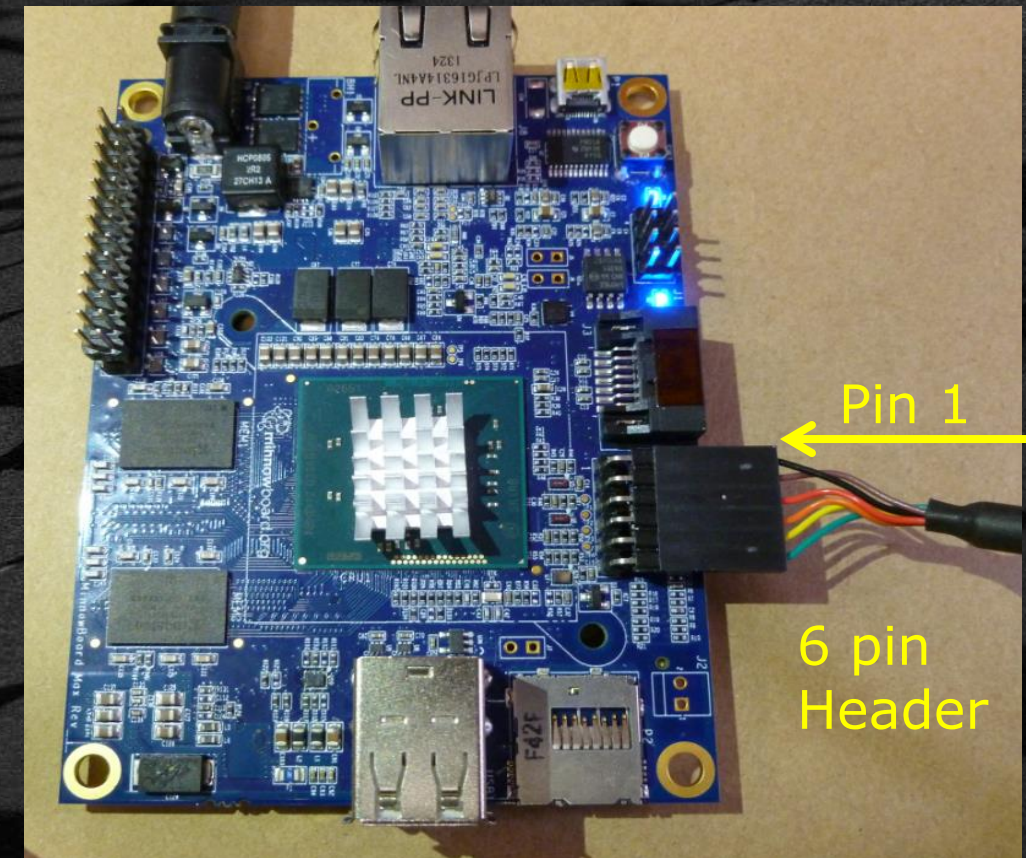
Hardware:

- System Under Test (SUT) – MinnowBoard Max /Turbot

- USB to 3.3V TTL Cable (6 pin to USB Type A)

- 5V** power supply

Connect the USB w/ 6 pin header to SUT

    - black wire(pin 1) is closest to the SATA connector
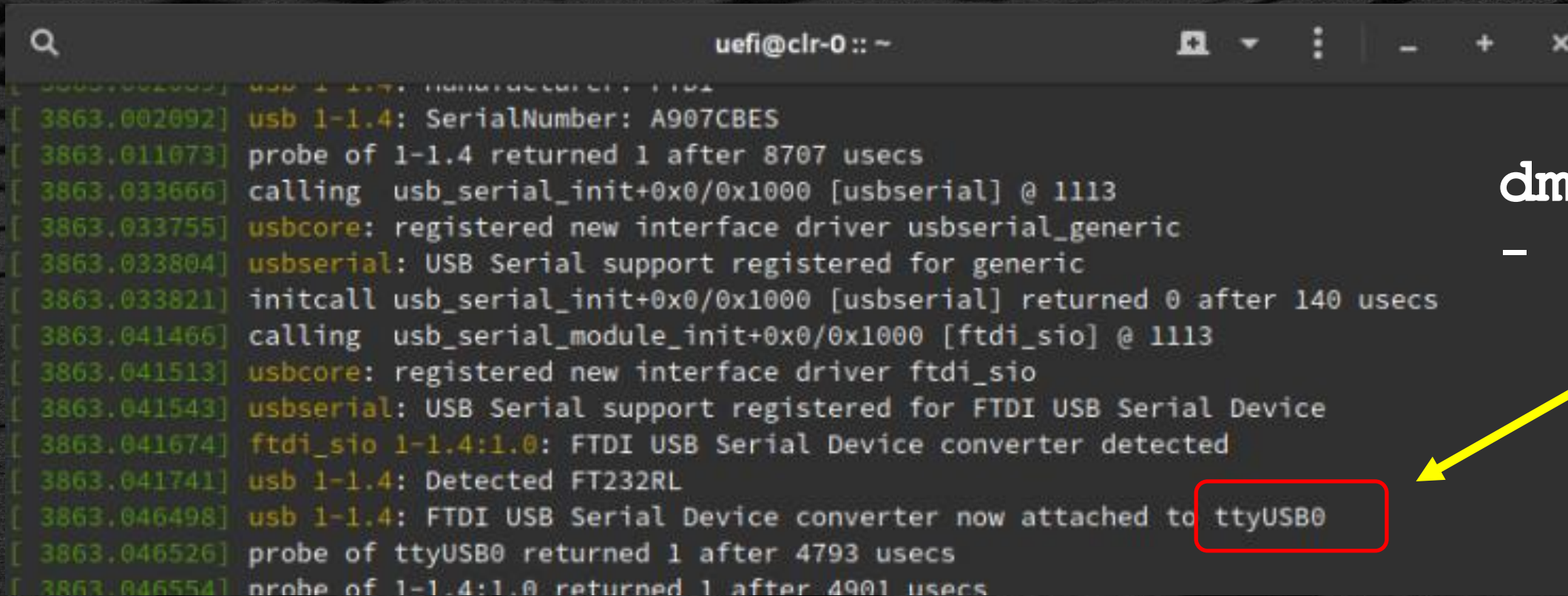
Connect the USB Type A connector to Host

Pin 1

6 pin Header

**Warning do not use any other power supply than 5V or the board will Fry

Setup MinnowBoard Max Test System

Open Terminal Prompt (Cnt-Alt-T)

```
bash$ sudo dmesg                          (to check which USB port is assigned)
bash$ sudo chmod 666 /dev/ttyUSBn         (where n is the FTDI number)
```


**dmesg command
- ttyUSB0**

Power on MinnowBoard Max

Connect the Power supply cable to the MinnowBoard Max

```
bash$ screen /dev/ttyUSBn 115200
```

MinnowBoard Max should boot to the UEFI Shell in the Terminal – Screen .

While in screen
 Cnt-A then **D** goes back to terminal

 bash$ screen –r
(returns to screen)

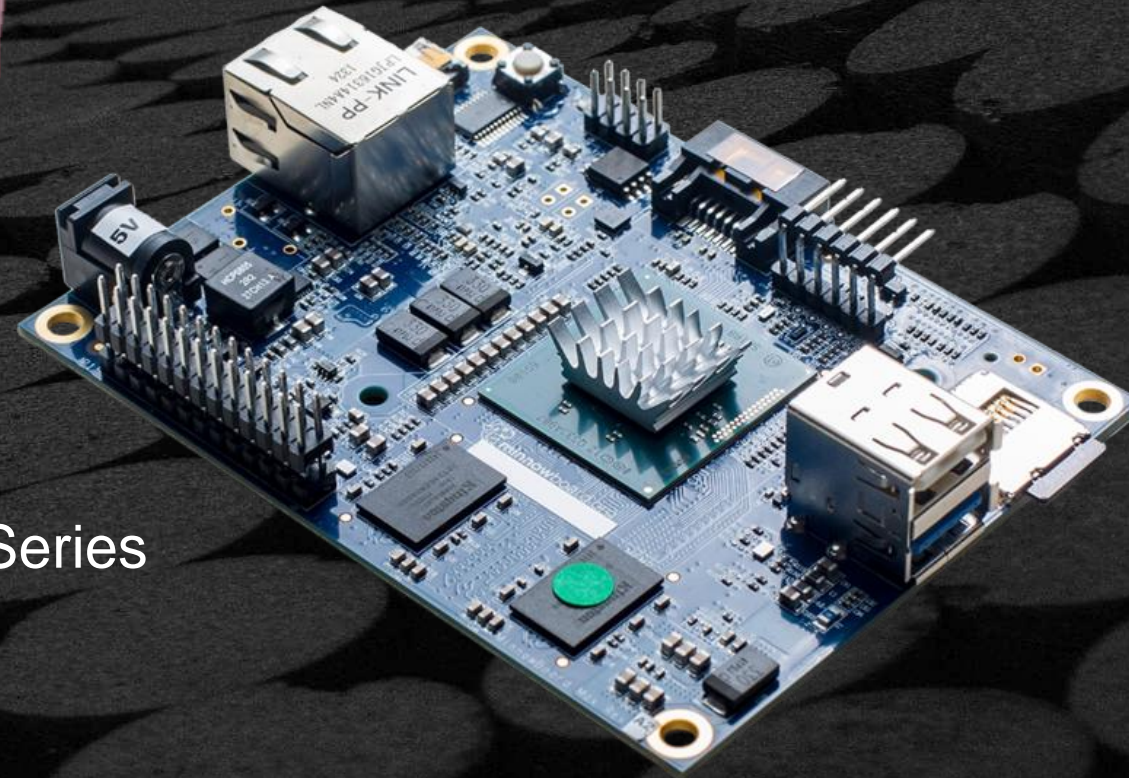Note: Cnt-H for Backspace

Cnt-A then Z to exit screen

9

END OF LAB

Return to the Beginning   or   **>**   to continue

BUILD MINNOWBOARD MAX / TURBOT

EDK II Platform – MinnowBoard Max/Turbot

minnowboard.org

Turbot

Intel® Atom processor E3800 Series
(Formerly Bay Trail-I)

Where to get Open Source MinnowBoard Max

- Open Source Max Development Wiki
- How to Download & Build:
  Readme.md
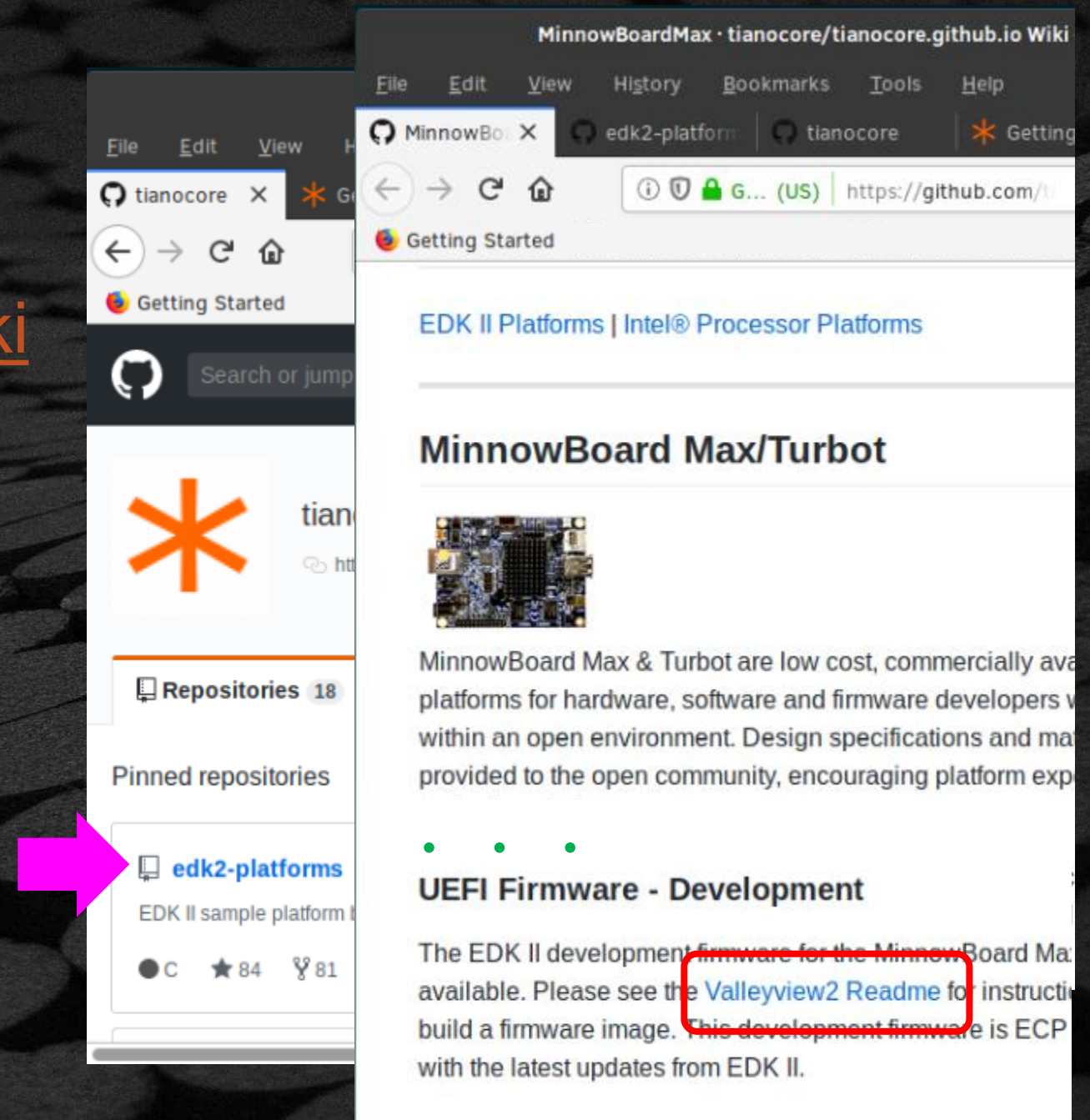
Down Load Max Lab Source

Download the PlatformBuildLab2_FW.zip from : github.com
PlatformBuildLab2_FW.zip
OR
Use `git clone` to download the PlatformBuildLab2_FW

```
bash$ cd $HOME
bash$ git clone https://github.com/tianocore-training/PlatformBuildLab2_FW.git
```

Directory PlatformBuildLab2_FW will be created

```
FW
- PlatformBuildLab
  - MaxWS                     - Minnowboard Max Source for the Labs
  - . . .
```

Linux setup for MinnowBoard Max Lab

ubuntu

Lab Setup Requirements – Ubuntu 16.04

```
bash$ sudo apt-get install build-essential uuid-dev iasl git gcc-5 nasm
bash$ sudo apt-get install screen
bash$ sudo apt-get install gcab
```

Clear Linux*
Project

Lab Setup Requirements – Clear Linux* Project

```
bash$ sudo swupd bundle-add devpkg-util-linux
bash$ sudo swupd bundle-add devpkg-gcab
```

Copy MinnowBoard Max Source

Open a terminal prompt (Alt-Cnt-T)

Create a working space source directory under the home directory

```
bash$ mkdir ~src
```

From the FW/PlatformBuildLab folder, copy and paste folder "../FW/MaxWS" to ~src

Copy

Paste

tianocore
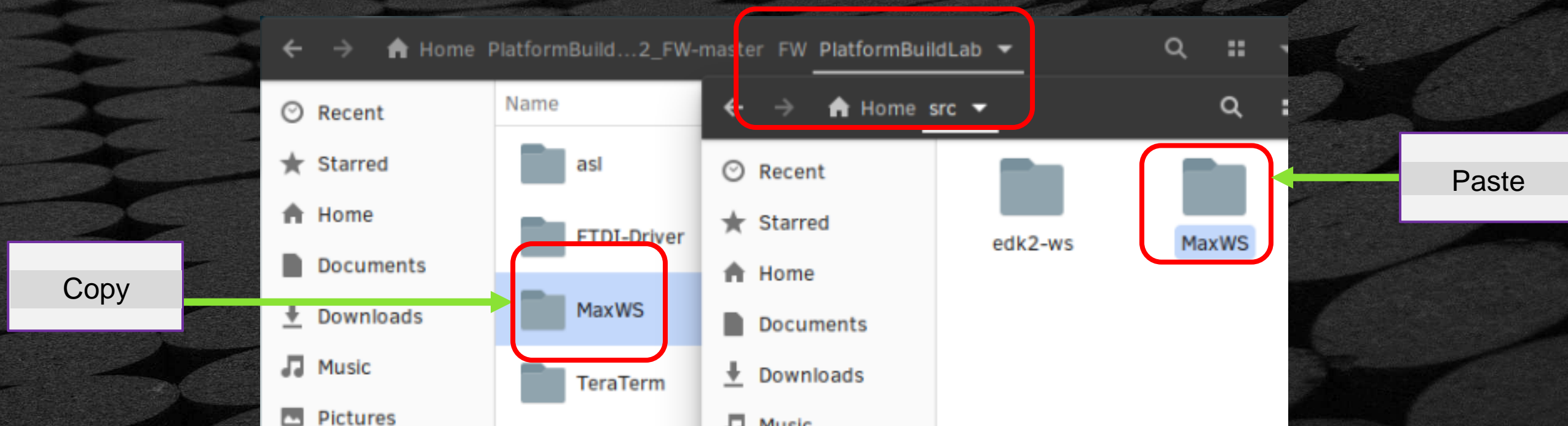
Rename or `mv` the directory "~src/MaxWS/edk2/BaseTools"

```
bash$ cd ~/src/MaxWS/edk2
bash$ mv BaseTools BaseToolsX
bash$ tar -xf BaseTools.tar.xz
```

Extract the file `BaseTools.tar.xz` to ~src/MaxWS/edk2
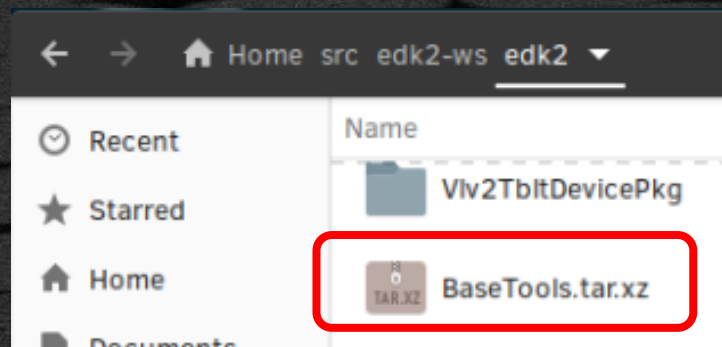
Double click
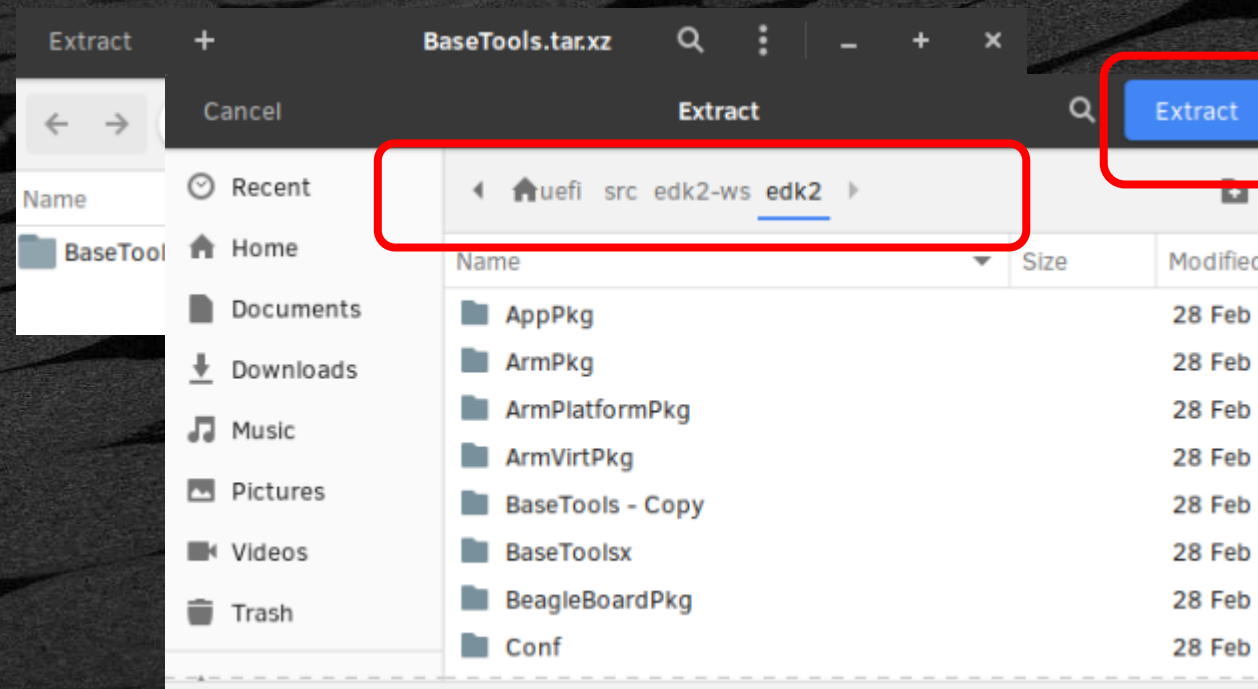
Select Extract directory
$HOME/src/MaxWS/edk2

```
./MaxWs/
    edk2/
        (EDK II common packages)
        BaseTools/
    edk2-platforms/
        Platform/Intel/
            Vlv2TbltDevicePkg /
        Silicon/Intel/
            Vlv2DeviceRefCodePkg/
    edk2-non-osi/
```

Invoke the Build from here

Platform DSC here

Steps to Build & Install Firmware

1. Open Terminal prompt & Cd ~/src/MaxWS/
2. Set up local build environment, edksetup.sh
3. Build `BaseTools`
4. Invoke the build process
5. Locate build output (.cap files for BIOS image)
6. Flash capsule image onto the platform
7. Reset and verify the new firmware

Next slide will follow the above steps

Setup Local Environment from Terminal

1 Terminal prompt (Cnt-Alt-T) & CD to work space directory

2 Set up Local environment (`PACKAGES_PATH`)

```
bash$ cd ~src/MaxWS

bash$ export WORKSPACE=$PWD
bash$ export PACKAGES_PATH=$WORKSPACE/edk2:\
    $WORKSPACE/edk2-platforms/Silicon/Intel:\
    $WORKSPACE/edk2-platforms/Platform/Intel:\
    $WORKSPACE/edk2-non-osi/Silicon/Intel

bash$ cd edk2/
bash$ chmod +x edksetup.sh

bash$ . edksetup.sh
```
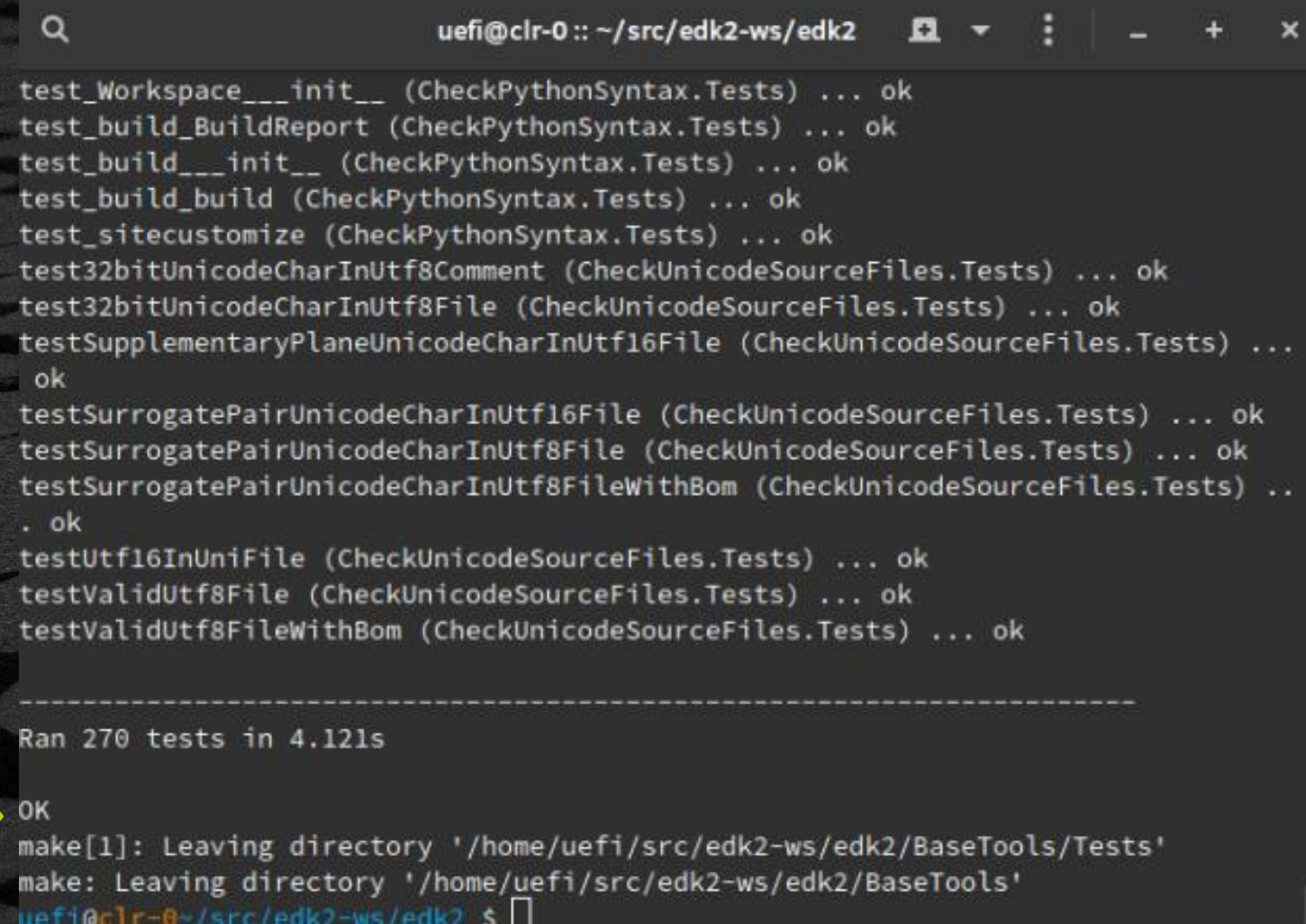
3

Run Make

```
bash$ cd ~/src/MaxWS/edk2
bash$ make -C BaseTools/
```

Make sure the tests pass OK

```
test_Workspace___init__ (CheckPythonSyntax.Tests) ... ok
test_build_BuildReport (CheckPythonSyntax.Tests) ... ok
test_build___init__ (CheckPythonSyntax.Tests) ... ok
test_build_build (CheckPythonSyntax.Tests) ... ok
test_sitecustomize (CheckPythonSyntax.Tests) ... ok
test32bitUnicodeCharInUtf8Comment (CheckUnicodeSourceFiles.Tests) ... ok
test32bitUnicodeCharInUtf8File (CheckUnicodeSourceFiles.Tests) ... ok
testSupplementaryPlaneUnicodeCharInUtf16File (CheckUnicodeSourceFiles.Tests) ...
 ok
testSurrogatePairUnicodeCharInUtf16File (CheckUnicodeSourceFiles.Tests) ... ok
testSurrogatePairUnicodeCharInUtf8File (CheckUnicodeSourceFiles.Tests) ... ok
testSurrogatePairUnicodeCharInUtf8FileWithBom (CheckUnicodeSourceFiles.Tests) ..
. ok
testUtf16InUniFile (CheckUnicodeSourceFiles.Tests) ... ok
testValidUtf8File (CheckUnicodeSourceFiles.Tests) ... ok
testValidUtf8FileWithBom (CheckUnicodeSourceFiles.Tests) ... ok

----------------------------------------------------------------------
Ran 270 tests in 4.121s

OK
make[1]: Leaving directory '/home/uefi/src/edk2-ws/edk2/BaseTools/Tests'
make: Leaving directory '/home/uefi/src/edk2-ws/edk2/BaseTools'
uefi@clr-0~/src/edk2-ws/edk2 $
```

uefi@clr-0 :: ~/src/edk2-ws/edk2

Platform Pre & Post Build Scripts

Many Platforms have a bash, bat or Python script file to pre or post process the EDK II build process

For MinnowBoard Max :
Pre build processing:
Python script Vlv2TbltDevicePkg/PreBuild.py – determines date and creates BiosId.bin in build output directory

Post build processing:
Python script Vlv2TbltDevicePkg/Feature/Capsule/GenerateCapsule/GenCapsuleAll.py – creates .CAP files for updating

4

From the edk2/ directory invoke the "build" command to build MinnowBoard Max

```
bash$ build -a IA32 -a X64 -t GCC5 -p Vlv2TbltDevicePkg/PlatformPkgX64.dsc -y Vlv.report -v
```



```
uefi@clr-0 :: ~/src/MaxWs/edk2

uefi@clr-0~/src/MaxWs/edk2 $ build -a IA32 -a X64 -n 5 -t GCC5 -b DEBUG -p Vlv2TbltDevicePkg/
PlatformPkgX64.dsc -y Vlv.report
Build environment: Linux-5.2.9-823.native-x86_64-with-glibc2.29
Build start time: 10:12:47, Aug.21 2019

WORKSPACE       = /home/uefi/src/MaxWs
PACKAGES_PATH   = /home/uefi/src/MaxWs/edk2:/home/uefi/src/MaxWs/edk2-platforms/Silicon/Inte
l:/home/uefi/src/MaxWs/edk2-platforms/Platform/Intel:/home/uefi/src/MaxWs/edk2-non-osi/Silico
n/Intel
```

```
PreBuild: python /home/uefi/src/MaxWs/edk2-platforms/Platform/Intel/Tools/GenBiosId/GenBiosId
.py -i /home/uefi/src/MaxWs/Build/Vlv2TbltDevicePkgX64/DEBUG_GCC5/BiosId.env -o /home/uefi/sr
c/MaxWs/Build/Vlv2TbltDevicePkgX64/DEBUG_GCC5/BiosId.bin -ot /home/uefi/src/MaxWs/Build/Vlv2T
bltDevicePkgX64/DEBUG_GCC5/BiosId.txt
PreBuild:
End of Pre-BUILD
Press Enter key to continue
```

Press Enter to Continue the build

Examine Build Parameters

```
build -a IA32 -a X64 -t GCC5 -p
      Vlv2TbltDevicePkg\PlatformPkgX64.dsc -y Vlv.report -v
```

`TARGET`	`= DEBUG`	Build Mode
`TARGET_ARCH`	`= IA32 X64`	CPU Architecture
`TOOL_CHAIN_TAG`	`= GCC5`	GCC Tool Chain
`ACTIVE_PLATFORM`	`= Vlv2TbltDevicePkg /PlatformPkgX64`	Platform DSC file
`Report file created`	`= Vlv.report`	PCDs, Libs, etc.

Platform Build and PCD Parameters

Platform Parameters

Many Platform Parameters are defined in a top .DSC file that controls PCD and build switches

For MinnowBoard Max : `Vlv2TbltDevicePkg/PlatformPkgConfig.dsc`
Example:

```
#
# TRUE is ENABLE. FALSE is DISABLE.
#
  //  . . .
DEFINE SECURE_BOOT_ENABLE = TRUE
DEFINE USER_IDENTIFICATION_ENABLE = FALSE
DEFINE VARIABLE_INFO_ENABLE = FALSE
DEFINE S3_ENABLE = TRUE
DEFINE CAPSULE_ENABLE = TRUE
DEFINE CAPSULE_RESET_ENABLE = TRUE
  // . . .
```

Build Process for RELEASE Target

4

From the edk2/ directory invoke the "build" command to build MinnowBoard Max

```
$bash build -a IA32 -a X64 -t GCC5 -b RELEASE -p Vlv2TbltDevicePkg/PlatformPkgX64.dsc -v
```



```
uefi@clr-O :: ~/src/MaxWs/edk2

g/PreBuild.py -a IA32 -a X64 -n 5 -t GCC5 -b RELEASE -p Vlv2TbltDevicePkg/PlatformPkgX64.dsc
-y Vlv-R.report -v --conf=/home/uefi/src/MaxWs/edk2/Conf all
POSTBUILD        = python /home/uefi/src/MaxWs/edk2-platforms/Platform/Intel/Vlv2TbltDevicePk
g/Feature/Capsule/GenerateCapsule/GenCapsuleAll_EDKII_TEST.py -a IA32 -a X64 -n 5 -t GCC5 -b
RELEASE -p Vlv2TbltDevicePkg/PlatformPkgX64.dsc -y Vlv-R.report -v --conf=/home/uefi/src/MaxW
s/edk2/Conf all

- Prebuild Start -

PreBuild: RelativePath/home/uefi/src/MaxWs/Build/Vlv2TbltDevicePkgX64/RELEASE_GCC5
PreBuild: mkdir Build/Vlv2TbltDevicePkgX64/RELEASE_GCC5
PreBuild: python /home/uefi/src/MaxWs/edk2-platforms/Platform/Intel/Tools/GenBiosId/GenBiosId
.py -i /home/uefi/src/MaxWs/Build/Vlv2TbltDevicePkgX64/RELEASE_GCC5/BiosId.env -o /home/uefi/
src/MaxWs/Build/Vlv2TbltDevicePkgX64/RELEASE_GCC5/BiosId.bin -ot /home/uefi/src/MaxWs/Build/V
lv2TbltDevicePkgX64/RELEASE_GCC5/BiosId.txt
PreBuild:
PreBuild:
End of Pre-BUILD
Press Enter key to continue
```

Press Enter to Continue the build

DEBUG & RELEASE Differences

Slower boot because the time it takes to display debug info

Larger image because of debug code & embedded info

Uses the serial port for debug string output

Contains detailed debug strings that show
the boot process and various ASSERT/TRACE errors

5 Locate the build .Cap image

The platform post build process will create capsule images from the multiple firmware volumes generated by the EDK II build process

tianocore

6 Flash the binary image

1. Access Max .CAP files from build folder
   - `. . ./Build/Vlv2TbltDevicePkgX64/Capsules/TestCert_X64_DEBUG_GCC5`
   - `*.cap`
   - `RELEASE   . . ./Capsules/TestCert_X64_RELEASE_GCC5`

2. Copy .cap files to a USB Thumb drive

3. Copy `CapsuleApp.efi` to a USB thumb drive

4. Boot into the UEFI Shell on Max then type "`FS0:`"
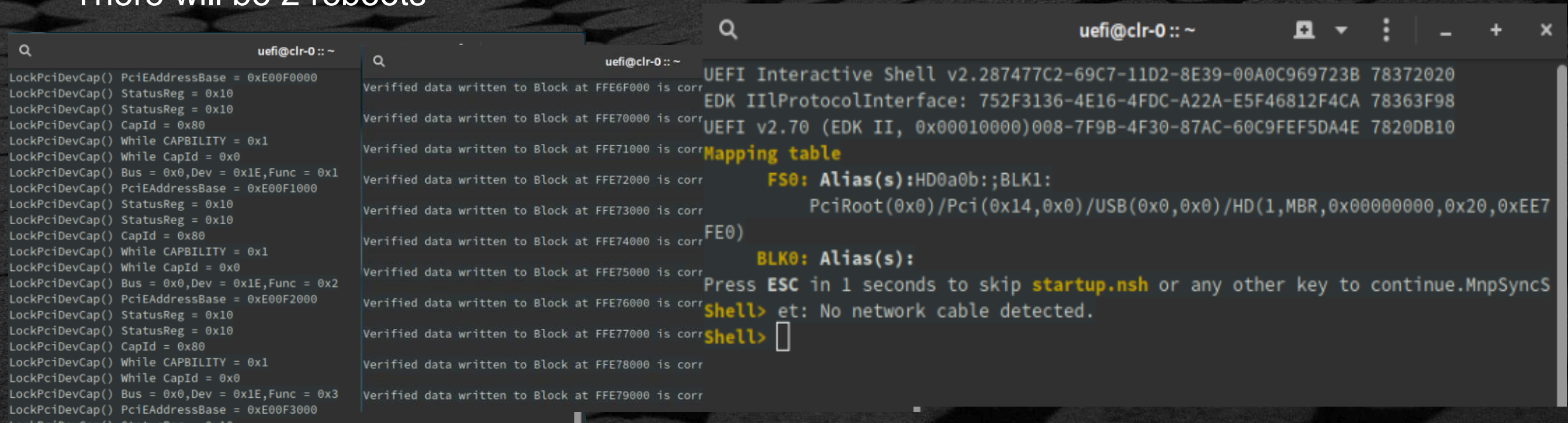
using "screen"
```
bash$ screen /dev/ttyUSB0 115200
```

```
            PciRoot(0x0)/Pci(0x14,0x0)/USB(0x0,0x0)/HD(1,MBR,0x00000000,0x20,0xEE7
FE0)
     BLK0: Alias(s):
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.MnpSyncS
Shell> et: No network cable detected.
Shell> fs0:
fs0:\>
```

6   Run CapsuleApp.efi utility with MinnowMax. . . cap file

(Note the "*TAB*" Key will fill out the command line for you)

```
FS0:\> CapsuleApp.efi MinnowMax.0.0.0.12.cap
```

System will start the Capsule update process

There will be 2 reboots

Capsule update with external Monitor

Logo with a progress bar will display update process progress

7 Reboot and Verify

- Verify that the Firmware was updated by checking the Date

- At the shell prompt type "exit"

- The EDK II front page will show the BIOS ID with Date/time stamp

```
Shell>
Shell> exit
```

SUMMARY

- Hardware Setup for MinnowBoard Max/Turbot
- Build a EDK II Platform using MinnowBoard Max/Turbot

Questions?

Return to Main Training Page

Return to Training Table of contents for next presentation link

www.tianocore.org