

UEFI & EDK II Training

Platform Build Lab MinnowBoard Max - Windows

tianocore.org

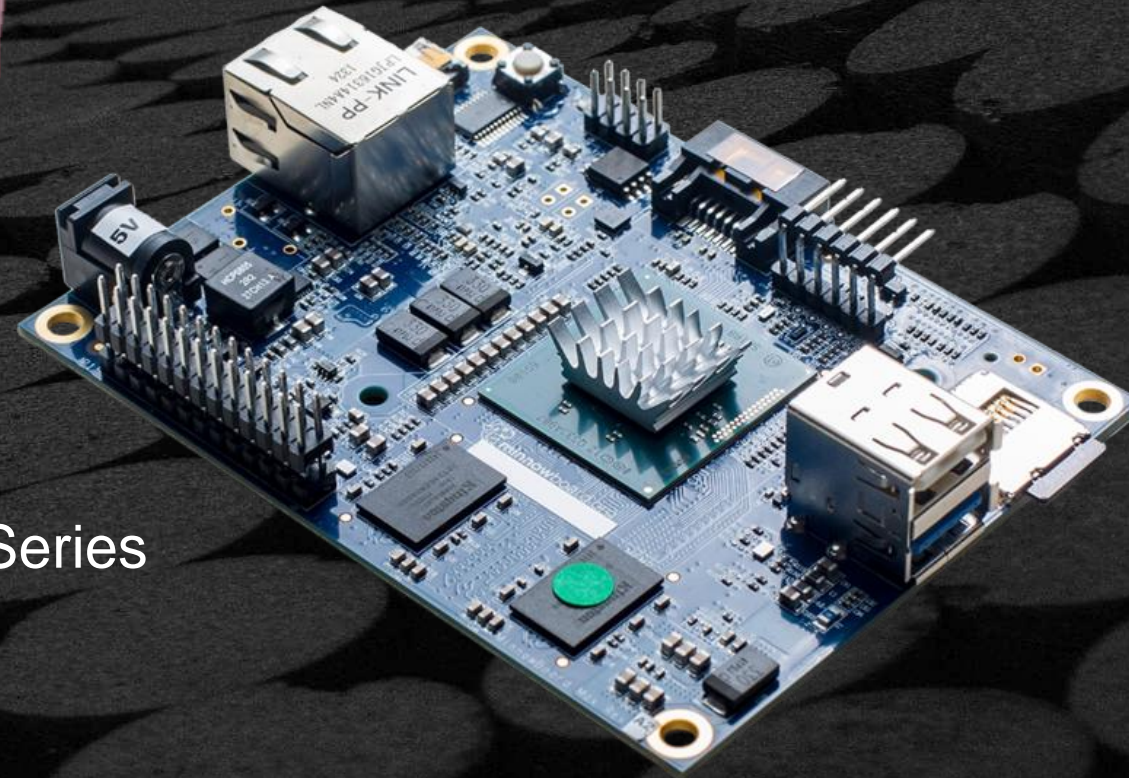
PLATFORM BUILD LABS

- ✿ 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

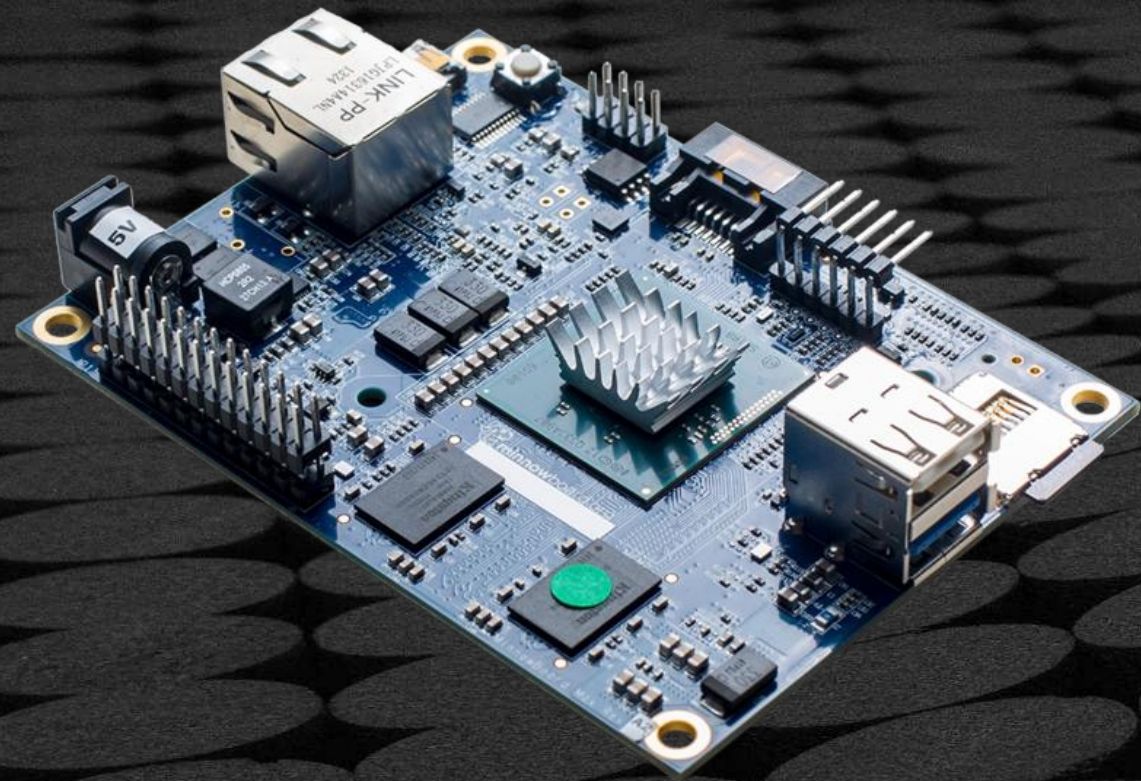
EDK II Platform – MinnowBoard Max/Turbot



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**

Instructions for Lab Materials

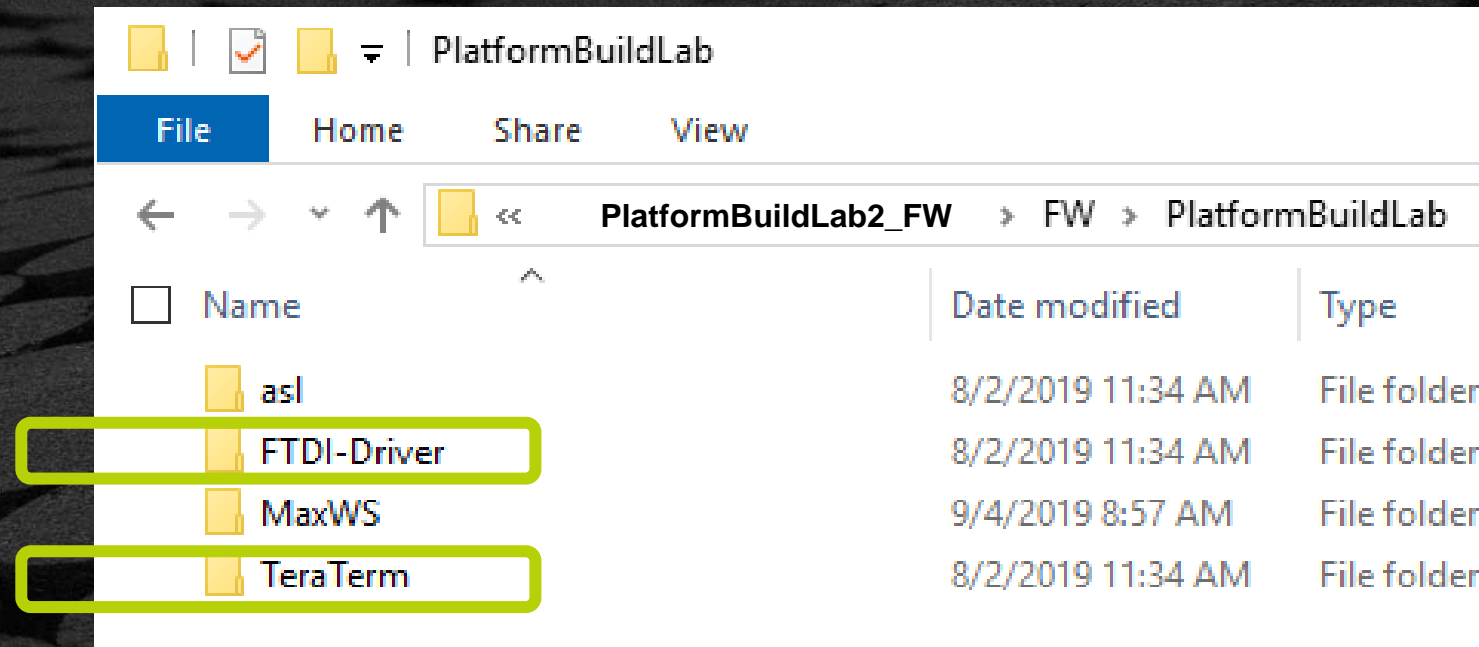
Directory C:\PlatformBuildLab2_FW\FW\PlatformBuildLab

FTDI Driver for Serial UART Cable (COM Port)

<http://www.ftdichip.com/FTDrivers.htm>

TeraTerm (terminal software for COM Port)

<https://en.osdn.jp/projects/ttssh2/releases/>



Note: Download FTDI Driver and TeraTerm as described in the Readme.txt for each directory

Setup MinnowBoard Max Test System

Hardware:

- System Under Test (SUT) – MinnowBoard Max
- USB to 3.3V TTL Cable (6 pin to USB Type A)
- 5V power supply

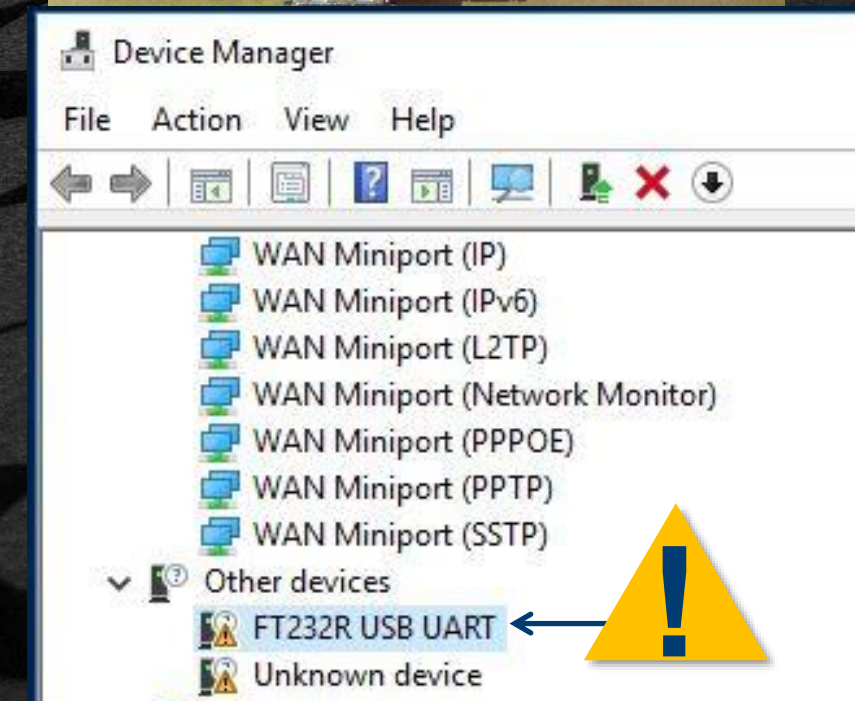
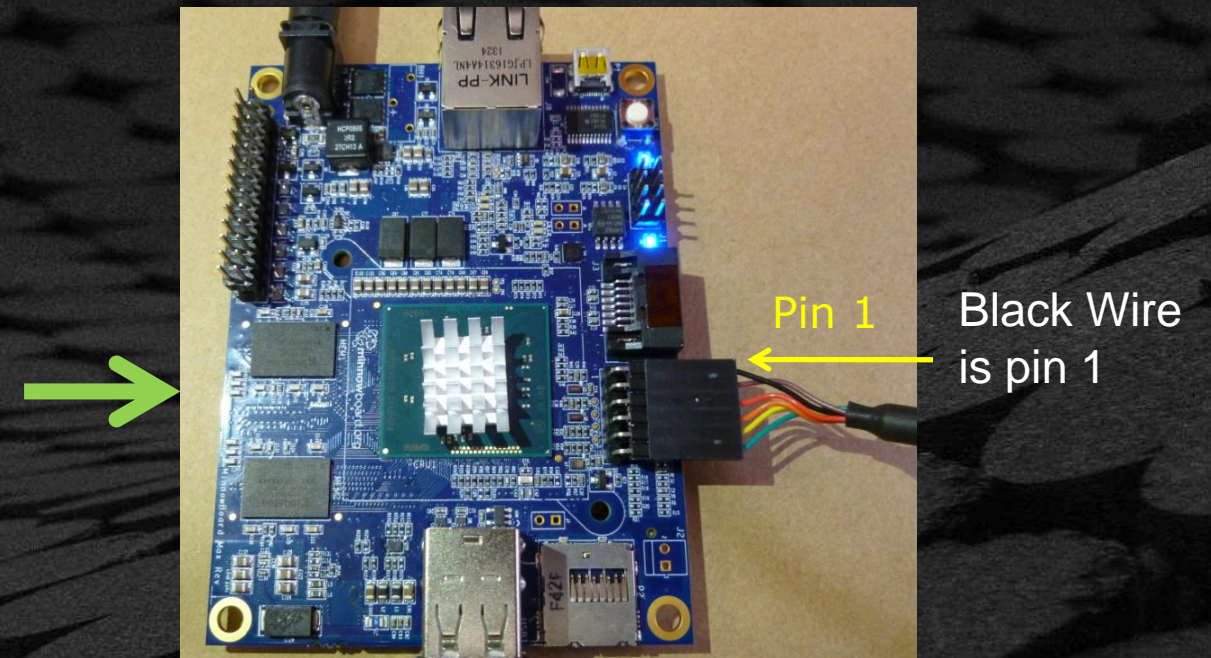
Connect the USB w/ 6 pin header to SUT (Max)

Connect the USB Type A connector to Host (Laptop)


On your Host **Go to the “Device Manager”** in the control panel.

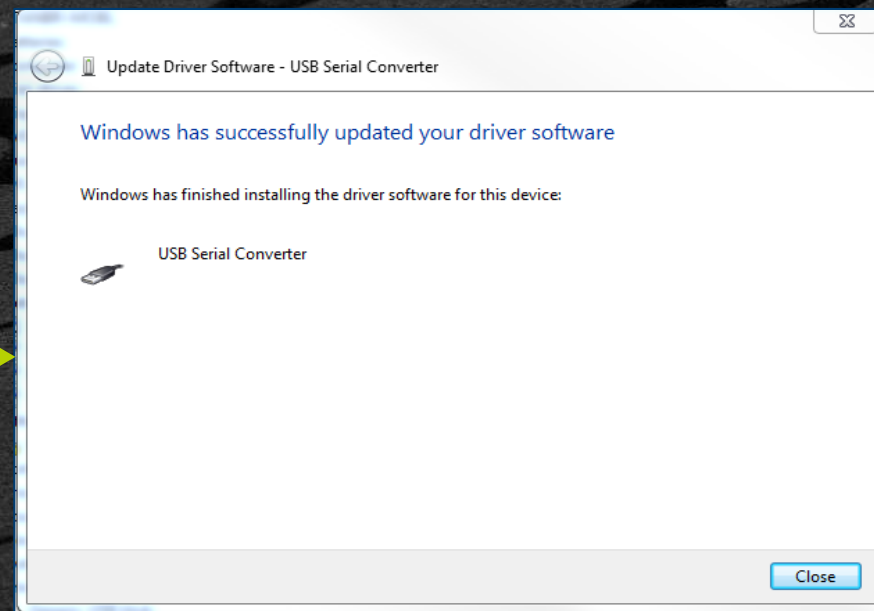
Under the **“Other devices”** category you will see a yellow ⚠ with a warning icon next to it.

You may already have this driver installed if you do not see a ⚠ warning under **“Other devices”**



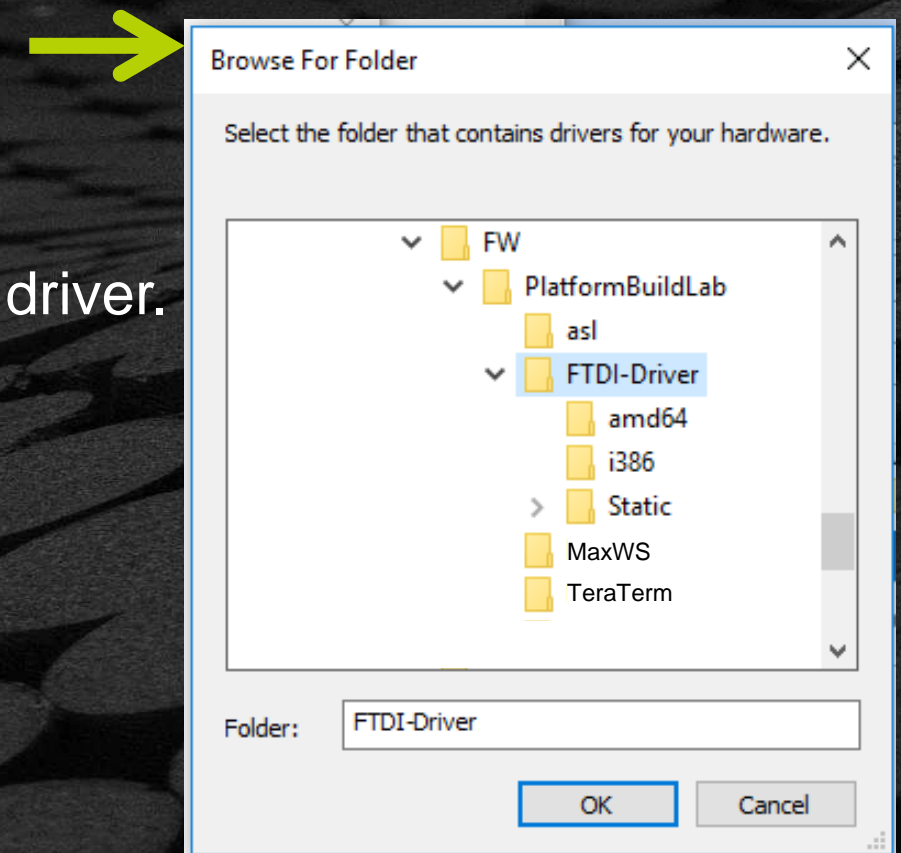
Setup COM port on Host

- Right click yellow  and select "Update Driver Software" from the **Device Manager** menu
- Select "Browse my computer for driver software".
- Click the **Browse** button. – Click on "Include subfolders"
- Browse to the location of the folder you unzipped earlier for the FIDI driver.
- Click on the folder and press **OK**.



- Press **Next**.

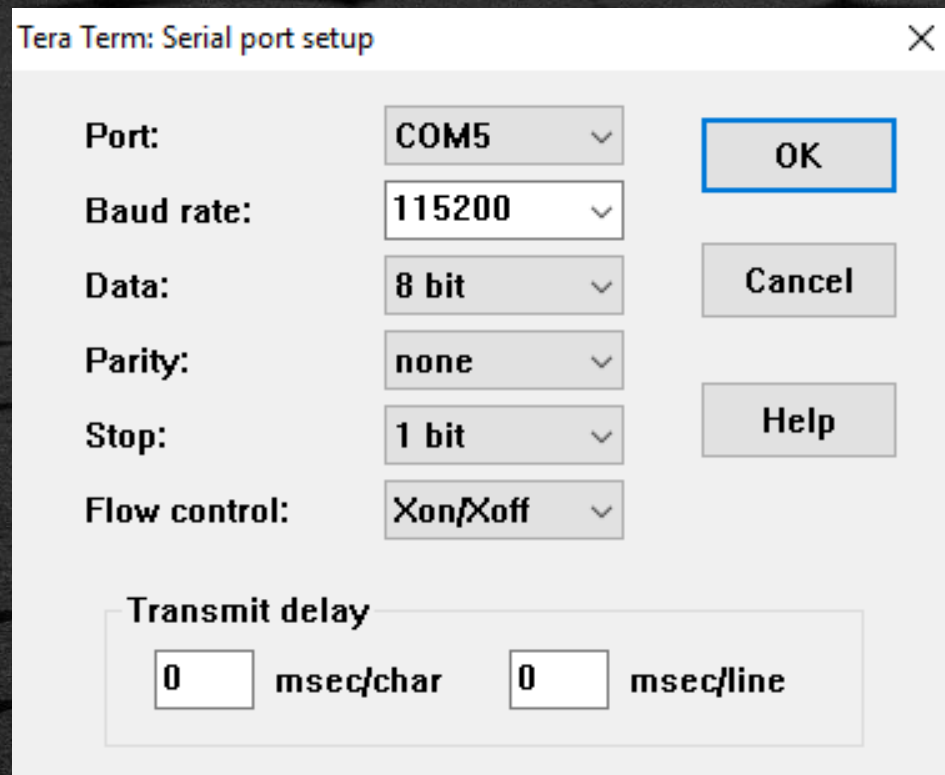
Driver will be installed



Unzip and Install TeraTerm

Open TeraTerm Software

Select the serial port assigned



Tera Term: Serial port setup

Port: COM5

Baud rate: 115200

Data: 8 bit

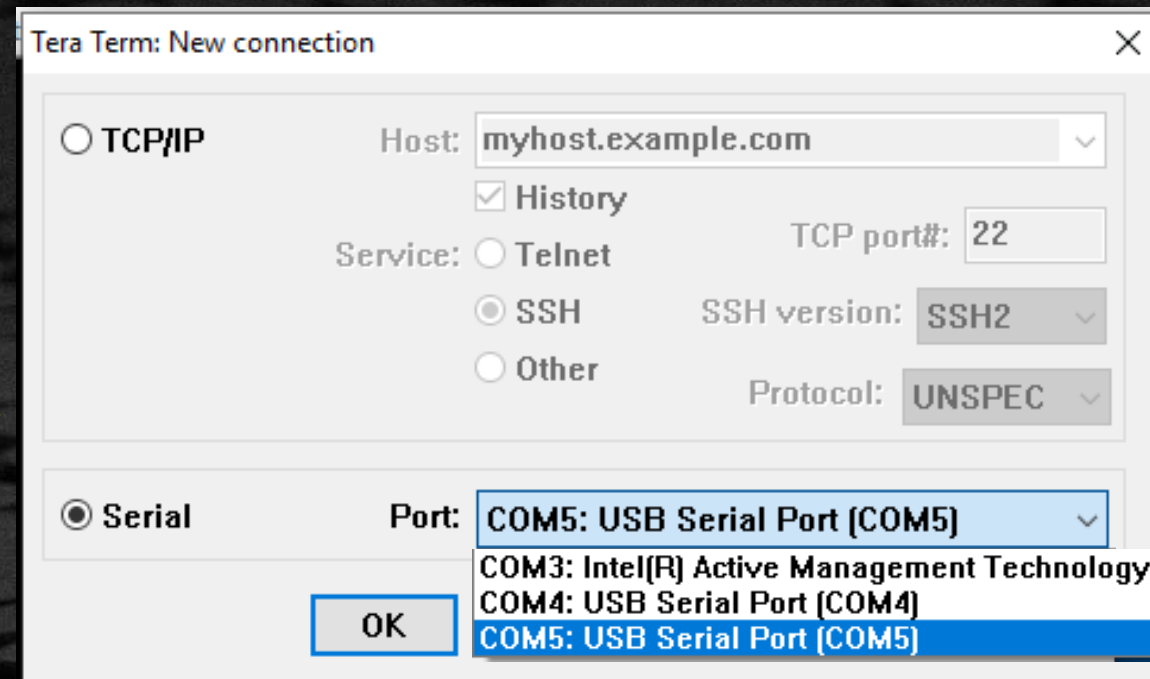
Parity: none

Stop: 1 bit

Flow control: Xon/Xoff

Transmit delay: 0 msec/char 0 msec/line

OK Cancel Help



Tera Term: New connection

☐ TCP/IP

Host: myhost.example.com

☒ History

Service: ☐ Telnet ☒ SSH ☐ Other

TCP port#: 22

SSH version: SSH2

Protocol: UNSPEC

☒ Serial

Port: COM5: USB Serial Port (COM5)

COM3: Intel(R) Active Management Technology - SOL (COM3)

COM4: USB Serial Port (COM4)

COM5: USB Serial Port (COM5)

OK

Choose the correct COM Port number

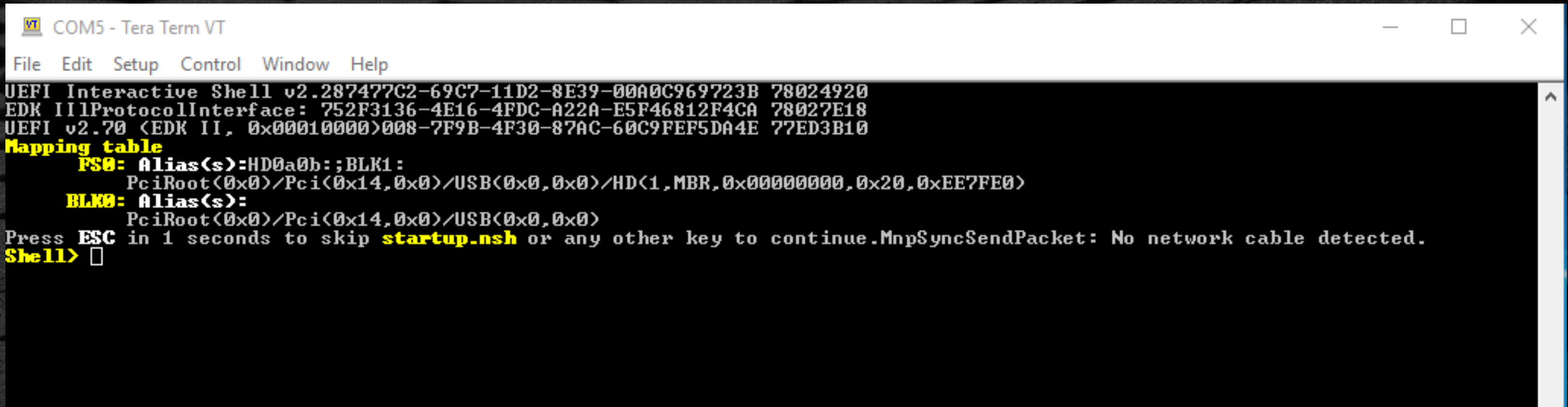
Go to **Setup->Serial Port** and set the following:

- Baud: 115200
- Parity: None
- Data Bits: 8
- Stop Bits: 1
- Flow Control: Xon/Xoff

Power on MinnowBoard Max

Connect the Power supply cable to the MinnowBoard Max

MinnowBoard Max should boot to the UEFI Shell in the TeraTerm window.



```
COM5 - Tera Term VT
File Edit Setup Control Window Help
UEFI Interactive Shell v2.287477C2-69C7-11D2-8E39-00A0C969723B 78024920
EDK IIProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 78027E18
UEFI v2.70 (EDK II, 0x00010000)008-7F9B-4F30-87AC-60C9FEP5DA4E 77ED3B10
Mapping table
  FS0: Alias(s)=HD0a0b:;BLK1:
        PciRoot(0x0)/Pci(0x14,0x0)/USB(0x0,0x0)/HD(1,MBR,0x00000000,0x20,0xEE7FE0)
  BLK0: Alias(s):
        PciRoot(0x0)/Pci(0x14,0x0)/USB(0x0,0x0)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.MnpSyncSendPacket: No network cable detected.
Shell> 
```

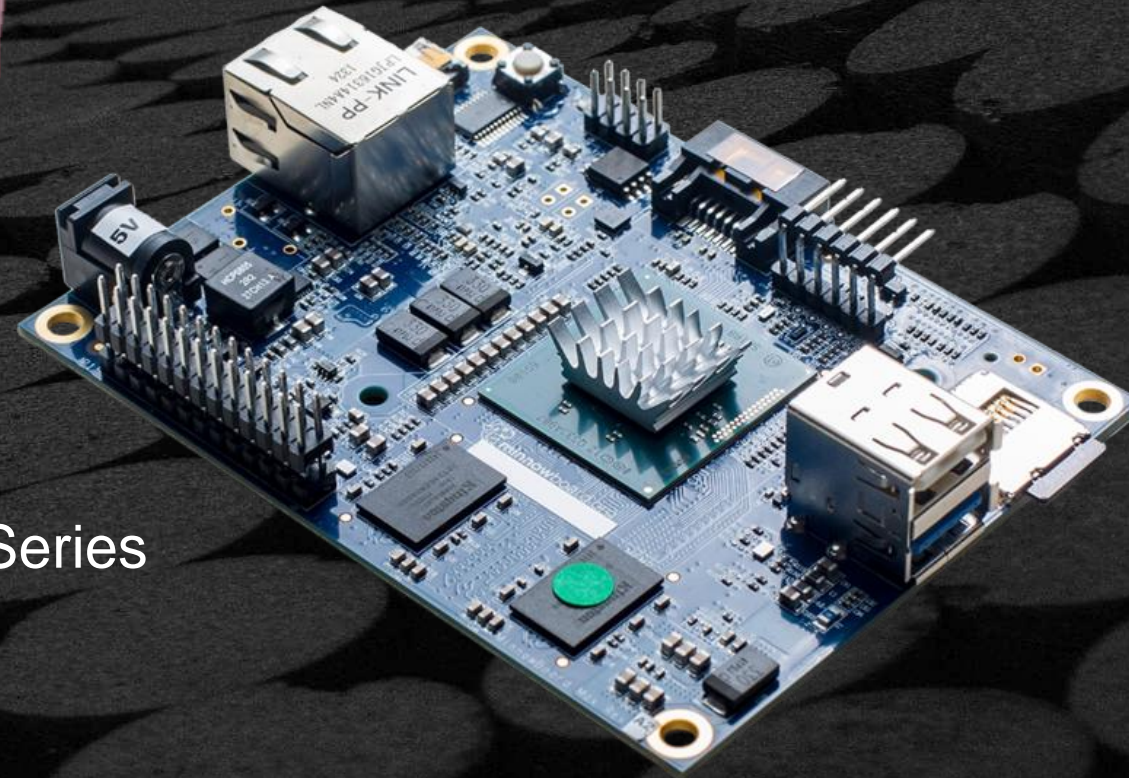

END OF LAB

Return to the Beginning



BUILD MINNOWBOARD TURBOT

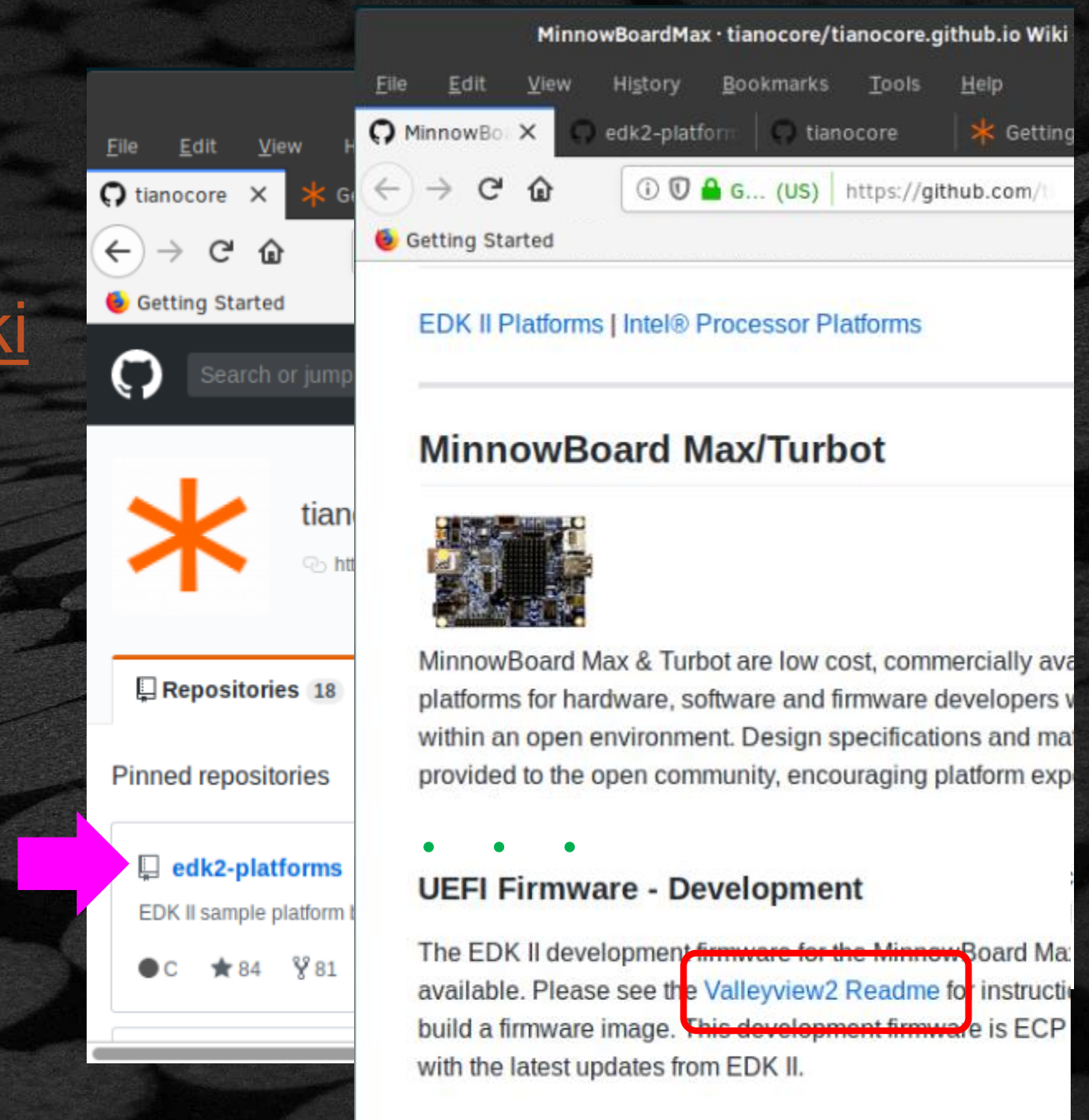
EDK II Platform – MinnowBoard Max/Turbot




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](#)



Download Max Lab Source

Download the PlatformBuildLab2_FW.zip from :  github.com
[PlatformBuildLab2_FW.zip](#)

OR

Use `git clone` to download the PlatformBuildLab2_FW

```
C:/> git clone https://github.com/tianocore-training/PlatformBuildLab2_FW.git
```

Directory PlatformBuildLab2_FW will be created

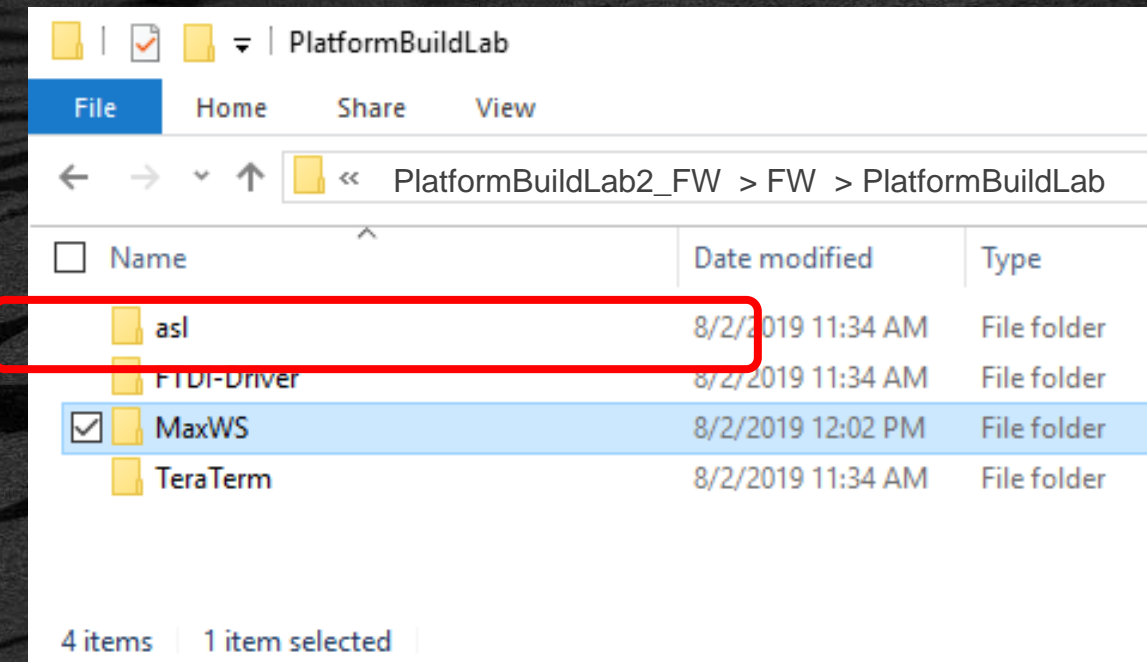
/FW

/PlatformBuildLab

- | | |
|---------------|---------------------------------------|
| - asl | - Asl Compiler |
| - FTDI-Driver | - Serial / USB cable |
| - MaxWS | - MinnowBoard Max Source for the Labs |
| - TeraTerm | - Terminal app |

Directory

C:\PlatformBuildLab2_FW\FW\PlatformBuildLab from
Download or zip



1 Copy \asl Folder to C:\

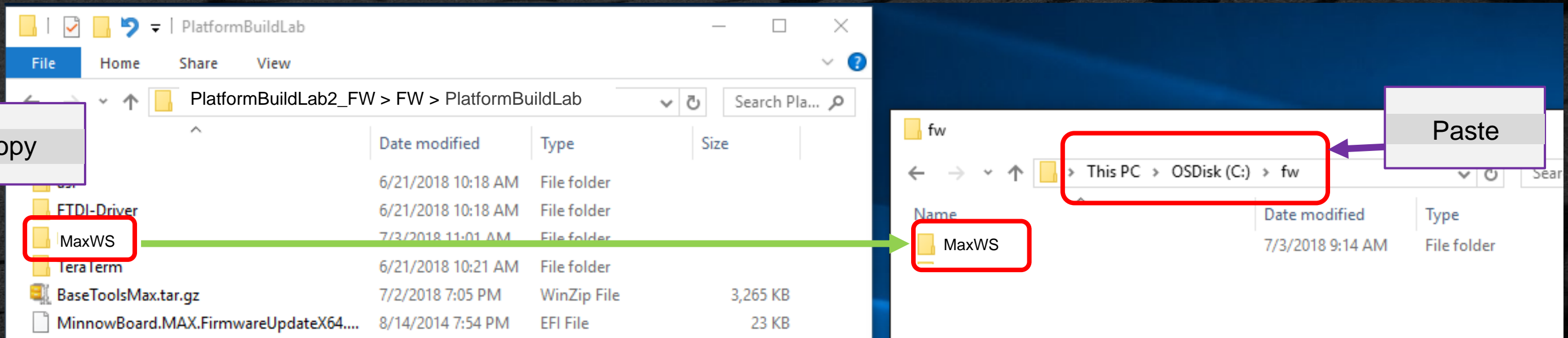
Note: Download Asl compiler described in the Readme.txt

Copy MinnowBoard Max Source

- 2 Open a VS Command prompt
Create a working space source directory under the home directory

```
C:\> mkdir FW
```

From the FW/PlatformBuildLab folder, copy and paste folder “..FW/MaxWS” to C:/FW/MaxWS



Platform Source Directory Structure

```
./MaxWs/  
edk2/ ← Invoke the Build from here  
    (EDK II common packages)  
    BaseTools/  
    edk2-platforms/  
    Platform/Intel/  
        Vlv2TbItDevicePkg / ← Platform DSC here  
    Silicon/Intel/  
        Vlv2DeviceRefCodePkg/  
    edk2-non-os/  
    nasm/**  
    openssl/**
```

** Nasm compiler and ** Openssl may need to be downloaded per the Readme.txt file in each directory

Steps to Build & Install Firmware

- 1 At VS command prompt Cd to directory : C:/FW/MaxWS
- 2 Set up local build environment
- 3 Invoke “Edksetup Rebuild” (build BaseTools)
- 4 Invoke the build process (DEBUG & RELEASE)
- 5 Locate build output (.cap files for BIOS image)
- 6 Flash capsule image onto the platform
- 7 Reset and boot new firmware to UEFI Shell

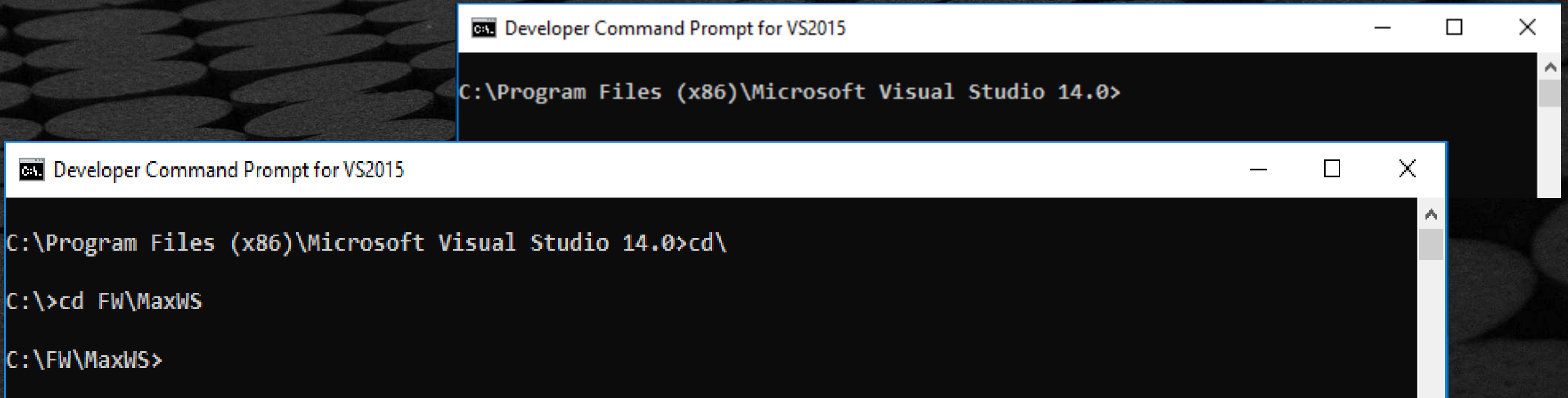
Next slide will follow the above steps

Open a VS Command Prompt

Follow Steps from [here](#) to Pin the Visual Studio Command Prompt to the Windows Task Bar

1 Open a Visual Studio Command Prompt &

```
> cd C:\FW\MaxWS
```



```
Developer Command Prompt for VS2015
C:\Program Files (x86)\Microsoft Visual Studio 14.0>

Developer Command Prompt for VS2015
C:\Program Files (x86)\Microsoft Visual Studio 14.0>cd\
C:\>cd FW\MaxWS
C:\FW\MaxWS>
```


Setup the Build Environment

Run Setenv.bat or type the following: (assumes Python3.7.2 installed)

```
$> setenv.bat
```

```
set WORKSPACE=%CD%
set PACKAGES_PATH=%WORKSPACE%\edk2;%WORKSPACE%\edk2-platforms\Silicon\Intel;%WORKSPACE%\edk2-
platforms\Platform\Intel;%WORKSPACE%\edk2-non-osl\Silicon\Intel

set EDK_TOOLS_PATH=%WORKSPACE%\edk2\BaseTools
path=%path%;%WORKSPACE%\openssl;%USERPROFILE%\AppData\Local\Programs\Python\Python37-32
set NASM_PREFIX=%WORKSPACE%\nasm\
```

Check if Python okay

```
$> python --version
Python 3.7.2
```

Note: Download Nasm compiler and Openssl described in each of the Readme.txt files

3

Invoke Edksetup from edk2 directory

```
$> cd edk2
```

```
$> Edksetup Rebuild
```

Developer Command Prompt for VS2015 - edksetup Rebuild

```
C:\FW\MaxWS\edk2>edksetup Rebuild
```

```
!!! ERROR !!! Cannot find BaseTools Bin Win32!!!
```

```
Please check the directory C:\FW\MaxWS\edk2\BaseTools\Bin\Win32
Or configure EDK_TOOLS_BIN env to point Win32 directory.
```

```
copying ... target.template to C:\FW\MaxWS\edk2\Conf\target.txt
```

```
copying ... tools_def.template to C:\FW\MaxWS\edk2\Conf\tools_def.txt
```

```
copying ... build_rule.template to C:\FW\MaxWS\edk2\Conf\build_rule.txt
```

Developer Command Prompt for VS2015 - edksetup Rebuild

```
SimpleFileParsing.c
```

```
cl.exe -c /nologo /Zi /c /O2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPR
```

```
2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPRECATED /D _CRT_NONSTD
```

```
I C:\FW\MaxWS\edk2\BaseTools\Source\C\Include\Ia32 -I C:\C\Com
```

```
seTools\Source\C\Include -I C:\FW\MaxWS\edk2\BaseTools\SLzFindMt.c
```

```
StringFuncs.c -FoStringFuncs.obj
```

```
StringFuncs.c
```

```
cl.exe -c /nologo /Zi /c /O2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPR
```

```
2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPRECATED /D _CRT_NONSTD
```

```
I C:\FW\MaxWS\edk2\BaseTools\Source\C\Include\Ia32 -I C:
```

```
seTools\Source\C\Include -I C:\FW\MaxWS\edk2\BaseTools\SBin\Win32\LzmaCompress.exe LzmaCompress.obj Sdk\C\Alloc.obj Sdk\C\LzFin
```

```
TianoCompress.c -FoTianoCompress.obj
```

```
TianoCompress.c
```

```
cl.exe -c /nologo /Zi /c /O2 /MT /W4 /WX /D _CR
```

```
2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPRECATED /D _CRT_NONSTD
```

```
I C:\FW\MaxWS\edk2\BaseTools\Source\C\Include\Ia32 -I C:
```

```
seTools\Source\C\Include -I C:\FW\MaxWS\edk2\BaseTools\S
```

```
PcdValueCommon.c -FoPcdValueCommon.obj
```

```
PcdValueCommon.c
```

```
lib.exe /nologo /out:C:\FW\MaxWS\edk2\BaseTools\
```

```
c32.obj Decompress.obj EfiCompress.obj EfiUtilityMsgs.obexecute command "nmake all" in directory C:\FW\MaxWS\edk2\BaseTools\Sour
```

```
th.obj ParseGuidedSectionTools.obj ParseInf.obj PeCoffLo
```

```
cdValueCommon.obj
```

```
execute command "nmake all" in directory C:\FW\MaxWS\edk
```

```
#####
```

```
# Build executables
```

```
#####
```

Developer Command Prompt for VS2015

```
cl.exe -c /nologo /Zi /c /O2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPR
```

```
WS\edk2\BaseTools\Source\C\Include -I C:\FW\MaxWS\edk2\BaseTools\Source\
```

```
C\Common Sdk\C\Bra86.c -FoSdk\C\Bra86.obj
```

```
Bra86.c
```

```
cl.exe -c /nologo /Zi /c /O2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPR
```

```
WS\edk2\BaseTools\Source\C\Include -I C:\FW\MaxWS\edk2\BaseTools\Source\
```

```
C\Common Sdk\C\LzFindMt.c -FoSdk\C\LzFindMt.obj
```

```
LzFindMt.c
```

```
cl.exe -c /nologo /Zi /c /O2 /MT /W4 /WX /D _CRT_SECURE_NO_DEPR
```

```
WS\edk2\BaseTools\Source\C\Include -I C:\FW\MaxWS\edk2\BaseTools\Source\
```

```
C\Common Sdk\C\Threads.c -FoSdk\C\Threads.obj
```

```
Threads.c
```

```
link.exe /nologo /debug /OPT:REF /OPT:ICF=10 /incremental:no /no
```

```
le.obj Sdk\C\7zStream.obj Sdk\C\Bra86.obj Sdk\C\LzFindMt.obj Sdk\C\Threa
```

```
copy LzmaF86Compress.bat C:\FW\MaxWS\edk2\BaseTools\Bin\Win32\Lz
```

```
1 file(s) copied.
```

```
#####
```

```
# Install to C:\FW\MaxWS\edk2\BaseTools\Lib\Win32
```

```
# Install to C:\FW\MaxWS\edk2\BaseTools\Bin\Win32
```

```
#####
```

```
execute command "nmake all" in directory C:\FW\MaxWS\edk2\BaseTools\Sour
```

```
th.obj ParseGuidedSectionTools.obj ParseInf.obj PeCoffLo
```

```
cdValueCommon.obj
```

```
execute command "nmake all" in directory C:\FW\MaxWS\edk
```

```
!!! WARNING !!! No CYGWIN_HOME set, gcc build may not be used !!!
```

```
#####
```

```
C:\FW\MaxWS\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

Build Process for DEBUG Target

4

From the edk2 directory invoke the “build” command to build MinnowBoard Max
Note: Use the Your VS TAG below with “-t” option

```
$> build -a IA32 -a X64 -t VS2015x86 -p Vlv2TbлтDevicePkg\PlatformPkgX64.dsc -y Vlv.report -v
```

```
Developer Command Prompt for VS2015 - build -a IA32 -a X64 -n 5 -t VS2015x86 -b DEBUG -p Vlv2TbлтDevicePkg\PlatformPkgX64.dsc -v
PREBUILD      = python c:\fw\maxws\edk2-platforms\Platform\Intel\Vlv2TbлтDevicePkg\PreBuild.p
-n 5 -t VS2015x86 -b DEBUG -p Vlv2TbлтDevicePkg\PlatformPkgX64.dsc -v --conf=c:\fw\maxws\edk2\c
POSTBUILD     = python c:\fw\maxws\edk2-platforms\Platform\Intel\Vlv2TbлтDevicePkg\Feature/Ca
psule/GenCapsuleAll_EDKII_TEST.py -a IA32 -a X64 -n 5 -t VS2015x86 -b DEBUG -p Vlv2TbлтDevicePkg
dsc -v --conf=c:\fw\maxws\edk2\conf all

- Prebuild Start -

PreBuild: RelativePathc:\fw\maxws\Build
PreBuild: mkdir Build
PreBuild: RelativePathc:\fw\maxws\Build\Vlv2TbлтDevicePkgX64
PreBuild: mkdir Build\Vlv2TbлтDevicePkgX64
PreBuild: RelativePathc:\fw\maxws\Build\Vlv2TbлтDevicePkgX64\DEBUG_VS2015x86
PreBuild: mkdir Build\Vlv2TbлтDevicePkgX64\DEBUG_VS2015x86
PreBuild: python C:\FW\MaxWS\edk2-platforms\Platform\Intel\Tools\GenBiosId\GenBiosId.py -i c:\fw
2TbлтDevicePkgX64\DEBUG_VS2015x86\BiosId.env -o c:\fw\maxws\Build\Vlv2TbлтDevicePkgX64\DEBUG_VS2
n -ot c:\fw\maxws\Build\Vlv2TbлтDevicePkgX64\DEBUG_VS2015x86\BiosId.txt
PreBuild:
PreBuild:
End of Pre-BUILD
Press Enter key to continue
```

Press Enter to
Continue the build

Note: RC.EXE Resource Compiler See [Link](#):

www.tianocore.org

Errors

Examine Build Parameters

```
build -a IA32 -a X64 -t VS2015x86  
-p Vlv2Tb1tDevicePkg\PlatformPkgX64.dsc -y Vlv.report -v
```

TARGET	= DEBUG
TARGET_ARCH	= IA32 X64
TOOL_CHAIN_TAG	= VS2015x86
ACTIVE_PLATFORM	= Vlv2Tb1tDevicePkg /PlatformPkgX64
Report file created	= Vlv.report

Build Mode

CPU Architecture

VS Tool Chain

Platform DSC file

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 : v1v2Tb1tDevicePkg/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
// . . .
```

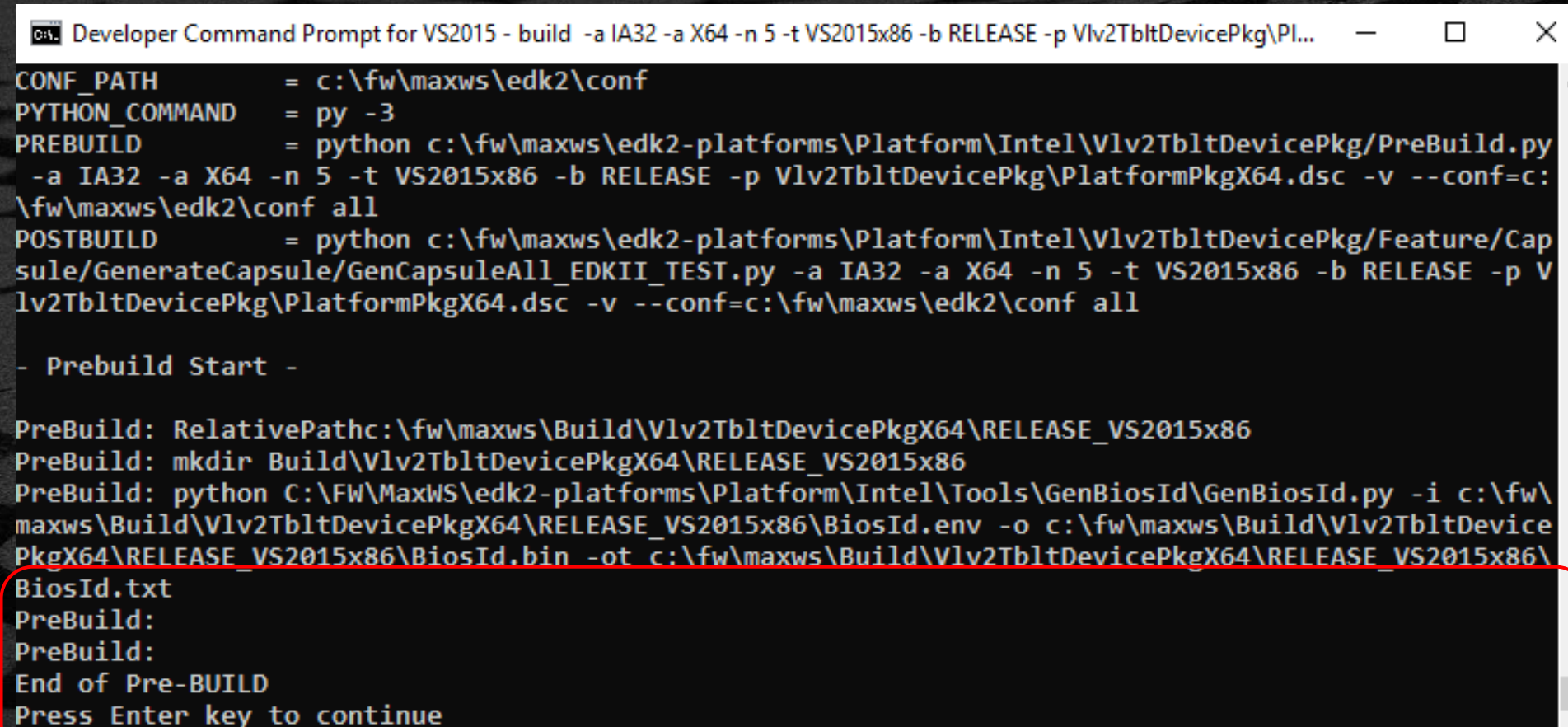

Build Process for RELEASE Target

4

From the VS Command Prompt ...

Note: Use the Your VS TAG below with “-t” option

```
> build -a IA32 -a X64 -t VS2015x86 -b RELEASE -p Vlv2TbлтDevicePkg\PlatformPkgX64.dsc -v
```



```
CONF_PATH      = c:\fw\maxws\edk2\conf
PYTHON_COMMAND = py -3
PREBUILD       = python c:\fw\maxws\edk2-platforms\Platform\Intel\Vlv2TbлтDevicePkg/PreBuild.py
                 -a IA32 -a X64 -n 5 -t VS2015x86 -b RELEASE -p Vlv2TbлтDevicePkg\PlatformPkgX64.dsc -v --conf=c:
                 \fw\maxws\edk2\conf all
POSTBUILD      = python c:\fw\maxws\edk2-platforms\Platform\Intel\Vlv2TbлтDevicePkg/Feature/Cap
                 sule/GenerateCapsule/GenCapsuleAll_EDKII_TEST.py -a IA32 -a X64 -n 5 -t VS2015x86 -b RELEASE -p V
                 lv2TbлтDevicePkg\PlatformPkgX64.dsc -v --conf=c:\fw\maxws\edk2\conf all

- Prebuild Start -

PreBuild: RelativePathc:\fw\maxws\Build\Vlv2TbлтDevicePkgX64\RELEASE_VS2015x86
PreBuild: mkdir Build\Vlv2TbлтDevicePkgX64\RELEASE_VS2015x86
PreBuild: python C:\FW\MaxWS\edk2-platforms\Platform\Intel\Tools\GenBiosId\GenBiosId.py -i c:\fw\
maxws\Build\Vlv2TbлтDevicePkgX64\RELEASE_VS2015x86\BiosId.env -o c:\fw\maxws\Build\Vlv2TbлтDevice
PkgX64\RELEASE_VS2015x86\BiosId.bin -ot c:\fw\maxws\Build\Vlv2TbлтDevicePkgX64\RELEASE_VS2015x86\
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 images

```

C:\> Developer Command Prompt for VS2015

GenCapsuleAll_New_Root: RelativePathc:\fw\maxdev\ws\Build\Vlv2TbltDevicePkgX64\Capsules\firmware.m
1
GenCapsuleAll_New_Root: remove Build\Vlv2TbltDevicePkgX64\Capsules\firmware.metainfo.xml
GenCapsuleAll_New_Root: RelativePathc:\fw\maxdev\ws\Build\Vlv2TbltDevicePkgX64\Capsules\firmware.bi
GenCapsuleAll_New_Root: remove Build\Vlv2TbltDevicePkgX64\Capsules\firmware.bin
GenCapsuleAll_New_Root:
*** Capsule update files in directory :
*** c:\fw\maxdev\ws\Build\Vlv2TbltDevicePkgX64\CapsulesTestCert_X64_DEBUG_VS2015x86
End of Post-BUILD

- Postbuild Done -

```

The platform post build process will create capsule images to use with a capsule update process

The script displays the location of the final .cap files

6 Flash the binary image


1. Access Max .CAP files from build folder

-/Build/Vlv2TbltDevicePkgX64/Capsules/TestCert_X64_DEBUG_VS2015x86
- *.cap
- RELEASE/Capsules/TestCert_X64_RELEASE_VS2015x86

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:"



```

EDK II F0C0C0C0INTERFACE: F52F3130 7E10 7FDC B22B E3F40012F7C8 70E07710
UEFI v2.50 (EDK II, 0x00010000)008-7F9B-4F30-87AC-60C9FEF5DA4E 76AE0A70
Mapping table
FS0: Alias(s):HD8b0b0b::BLK1:
      PciRoot(0x0)/Pci(0x14,0x0)/USB(0x1,0x0)/USB(0x1,0x0)/HD(1,MBR,0x00427D1E,0x40,0x1EAF00)
BLK0: Alias(s):
      PciRoot(0x0)/Pci(0x14,0x0)/USB(0x1,0x0)/USB(0x1,0x0)
Press ESC in 4 seconds to skip startup.nsh or any other key to continue.
Shell> fs0:
  
```

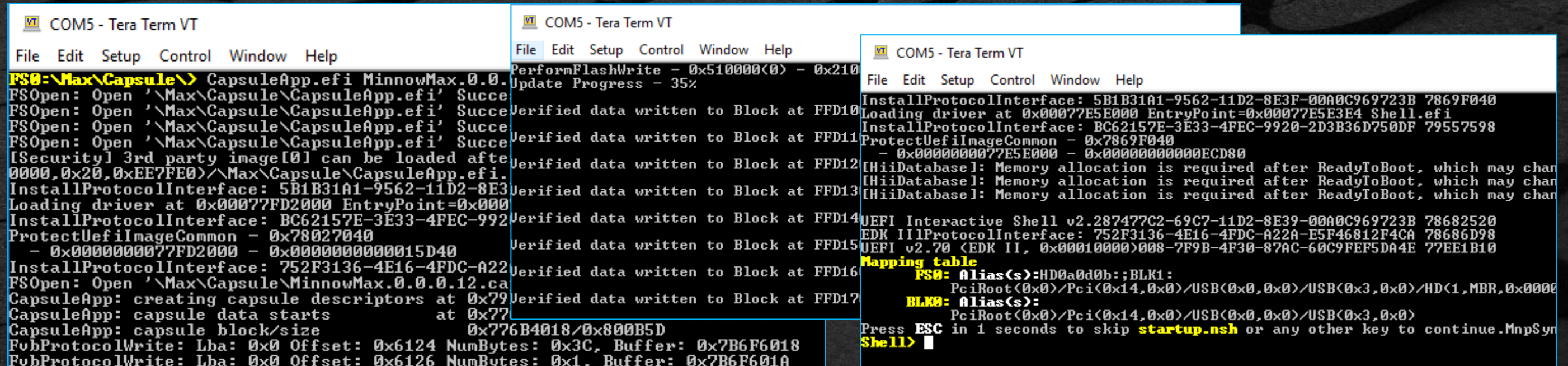

6

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



```

COM5 - Tera Term VT
File Edit Setup Control Window Help
FS0:\Max\Capsule\> CapsuleApp.efi MinnowMax.0.0.0.12.cap
FSOpen: Open '\Max\Capsule\CapsuleApp.efi' Success
FSOpen: Open '\Max\Capsule\CapsuleApp.efi' Success
FSOpen: Open '\Max\Capsule\CapsuleApp.efi' Success
FSOpen: Open '\Max\Capsule\CapsuleApp.efi' Success
[Security] 3rd party image[0] can be loaded after
0000,0x20,0xEE7FE0)\Max\Capsule\CapsuleApp.efi.
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 7869F040
Loading driver at 0x00077FD2000 EntryPoint=0x000
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 79557598
ProtectUefiImageCommon - 0x7869F040
- 0x0000000077FD2000 - 0x00000000000015D40
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 78686D98
FSOpen: Open '\Max\Capsule\MinnowMax.0.0.0.12.cap' Success
CapsuleApp: creating capsule descriptors at 0x79
CapsuleApp: capsule data starts at 0x77
CapsuleApp: capsule block/size 0x776B4018/0x800B5D
FvbProtocolWrite: Lba: 0x0 Offset: 0x6124 NumBytes: 0x3C, Buffer: 0x7B6F6018
FvbProtocolWrite: Lba: 0x0 Offset: 0x6126 NumBytes: 0x1, Buffer: 0x7B6F601A

COM5 - Tera Term VT
File Edit Setup Control Window Help
PerformFlashWrite - 0x510000(0) - 0x210
Update Progress - 35%
Verified data written to Block at FFD10
Verified data written to Block at FFD11
Verified data written to Block at FFD12
Verified data written to Block at FFD13
Verified data written to Block at FFD14
Verified data written to Block at FFD15
Verified data written to Block at FFD16
Verified data written to Block at FFD17

COM5 - Tera Term VT
File Edit Setup Control Window Help
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 7869F040
Loading driver at 0x00077E5E000 EntryPoint=0x00077E5E3E4 Shell.efi
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 79557598
ProtectUefiImageCommon - 0x7869F040
- 0x0000000077E5E000 - 0x000000000000ECD80
[HiiDatabase]: Memory allocation is required after ReadyToBoot, which may change
[HiiDatabase]: Memory allocation is required after ReadyToBoot, which may change
[HiiDatabase]: Memory allocation is required after ReadyToBoot, which may change
UEFI Interactive Shell v2.287477C2-69C7-11D2-8E39-00A0C969723B 78682520
EDK IIProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 78686D98
UEFI v2.70 (EDK II, 0x00010000)008-7F9B-4F30-87AC-60C9FEF5DA4E 77EE1B10
Mapping table
FS0: Alias(s):HD0a0d0b:;BLK1:
PciRoot(0x0)/Pci(0x14,0x0)/USB(0x0,0x0)/USB(0x3,0x0)/HD(1,MBR,0x00000000)
BLK0: Alias(s):
PciRoot(0x0)/Pci(0x14,0x0)/USB(0x0,0x0)/USB(0x3,0x0)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.MnpSyn
Shell>

```


Capsule Update with External Monitor

Logo with a progress bar will display update process progress



Verify After Firmware Update

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
```

```
Minnowboard Turbot D0 PLATFORM  
Intel(R) Atom(TM) CPU E3826 @ 1.46 GHz 1.46 GHz  
MNW2MAXW.X64.0200.D01.1908051725 2048 MB RAM  
  
WARNING: Test key detected.  
  
Select Language <Standard English>  
This is the option one  
adjusts to change the  
language for the current  
system  
  
> Device Manager  
> Boot Manager  
> Boot Maintenance Manager  
  
Continue  
Reset
```

<Enter>=Select Entry

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](#)

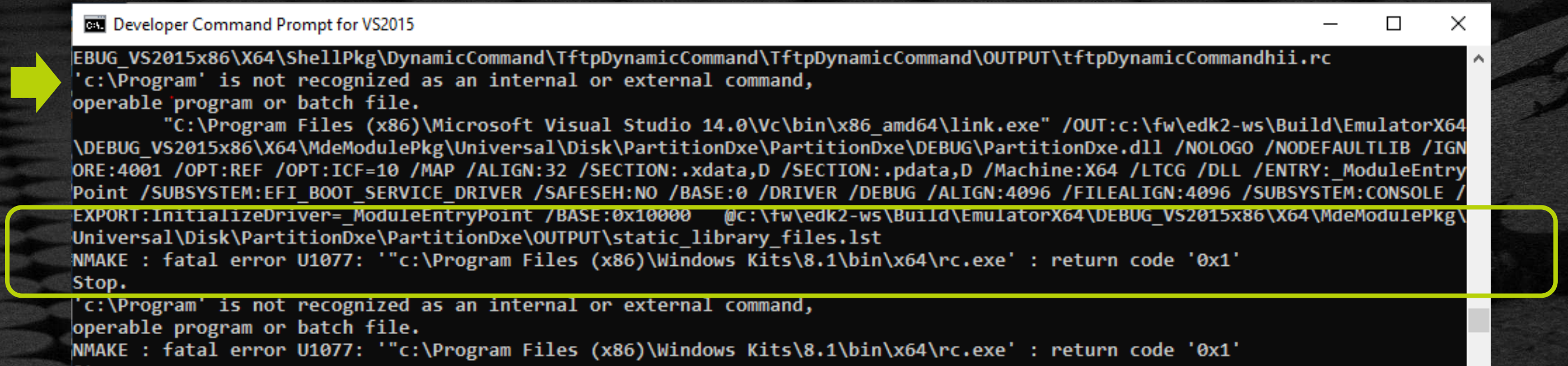


BACKUP

BUILD ERRORS

Build Error- RC.exe

Error message:



```

Developer Command Prompt for VS2015

EBUG_VS2015x86\X64\ShellPkg\DynamicCommand\TftpDynamicCommand\TftpDynamicCommand\OUTPUT\tftpDynamicCommandhii.rc
'c:\Program' is not recognized as an internal or external command,
operable program or batch file.
"C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\x86_amd64\link.exe" /OUT:c:\fw\edk2-ws\Build\EmulatorX64
\DEBUG_VS2015x86\X64\MdeModulePkg\Universal\Disk\PartitionDxe\PartitionDxe\DEBUG\PartitionDxe.dll /NOLOGO /NODEFAULTLIB /IGN
ORE:4001 /OPT:REF /OPT:ICF=10 /MAP /ALIGN:32 /SECTION:.xdata,D /SECTION:.pdata,D /Machine:X64 /LTCG /DLL /ENTRY:_ModuleEntry
Point /SUBSYSTEM:EFI_BOOT_SERVICE_DRIVER /SAFESEH:NO /BASE:0 /DRIVER /DEBUG /ALIGN:4096 /FILEALIGN:4096 /SUBSYSTEM:CONSOLE /
EXPORT:InitializeDriver=_ModuleEntryPoint /BASE:0x10000 @c:\fw\edk2-ws\Build\EmulatorX64\DEBUG_VS2015x86\X64\MdeModulePkg\
Universal\Disk\PartitionDxe\PartitionDxe\OUTPUT\static_library_files.lst
NMAKE : fatal error U1077: '"c:\Program Files (x86)\Windows Kits\8.1\bin\x64\rc.exe' : return code '0x1'
Stop.
'c:\Program' is not recognized as an internal or external command,
operable program or batch file.
NMAKE : fatal error U1077: '"c:\Program Files (x86)\Windows Kits\8.1\bin\x64\rc.exe' : return code '0x1'

```

Find where the RC.EXE is located on your VS Installation:

Example (VS 2015): The RC.exe is located on this machine:

C:\Program Files (x86)\Windows Kits\8.1\bin\x64

Edit Conf\tools_def.txt

Build Error- RC.exe Cont.

Edit `Conf\tools_def.txt`

Search for your installation of Visual Studio (2013, 2015, 2017) “RC.EXE”

Probably in path `C:\Program Files (x86)\Windows Kits\`

Update according to the path for where the RC.EXE is found

```
# Microsoft Visual Studio 2013 Professional Edition
DEFINE WINSDK8_BIN      = c:\Program Files\Windows Kits\8.1\bin\x86\
DEFINE WINSDK8x86_BIN   = c:\Program Files (x86)\Windows Kits\8.1\bin\x64

# Microsoft Visual Studio 2015 Professional Edition
DEFINE WINSDK81_BIN     = c:\Program Files\Windows Kits\8.1\bin\x86\
DEFINE WINSDK81x86_BIN  = c:\Program Files (x86)\Windows Kits\8.1\bin\x64

# Microsoft Visual Studio 2017 Professional Edition
DEFINE WINSDK10_BIN     = C:\Program Files (x86)\Windows Kits\10\bin\x86
```

Paths on your
machine



Build Error: fatal error C1041:

Build Error from fatal error C1041: cannot open program database

This Error is usually because the location you are building is being shared by another application in Windows. Example: Syncplicity may cause this

Error Message:

```
k:\fw\edk2\MdePkg\Library\BaseLib\LinkedList.c : fatal error C1041: cannot open program
database
'k:\fw\edk2\build\nt32ia32\debug_vs2013x86\ia32\mdepkg\library\baselib\baselib\vc120.pdb'; if
multiple CL.EXE write to the same .PDB file, please use /FS
NMAKE : fatal error U1077: '"C:\Program Files (x86)\Microsoft Visual Studio
12.0\Vc\bin\cl.exe"' : return code '0x2'
Stop.
```

Solution: Try using a Workspace that is not shared