Sitara Linux Software Developer's Guide

v4.01

Texas Instruments

Sitara Linux Software Developer's Guide

Supports Sitara Linux SDK v4.01

The SDK Migration Guide and Archived Software Developer's Guide are located in Reference Documentation.

Welcome to the Sitara Linux Software Developer's Guide

Thank you for choosing the Sitara AMx Evaluation Module (EVM). Please *bookmark* this page and refer back to it as needed. It is designed to quickly provide the information you need most while evaluating the AMx EVM.

We are always striving to improve this product. Please let us know if you have ideas or suggestions.

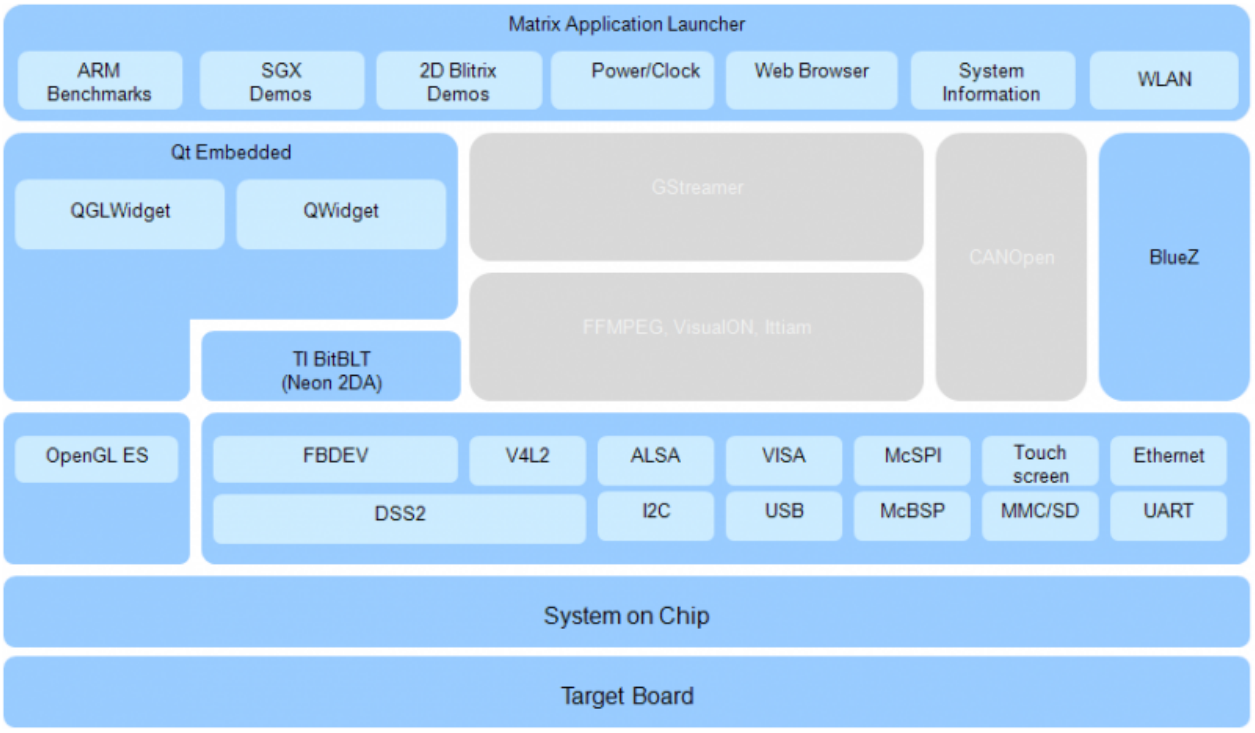
Supported Platforms

The following ARM MPU microprocessors are supported with this SDK version 4.01:

AM1810
AM180x
AM35x
AM37x

Software Overview

The following software stack illustrates at a high level the various components provided with the Sitara Linux SDK. **NOTE - Availability and usage of certain applications are platform dependent and clarified in the associated chapters within this Software Developer's Guide.**



EVM Hardware Overview

Details of the EVM associated with the Sitara Linux SDK are located in the Hardware User's Guide which is provided below and in the /docs directory of Sitara Linux SDK.

Platform	Document	EVM Provider
AM37x	AM37x EVM Hardware User's Guide [1]	Mistral
AM35x	Hardware User's Guide	Logic
AM180x	Hardware User's Guide	Logic
AM1810	Hardware User's Guide	Spectrum Digital
AM17x	Hardware User's Guide	

Configuring Your Development Environment

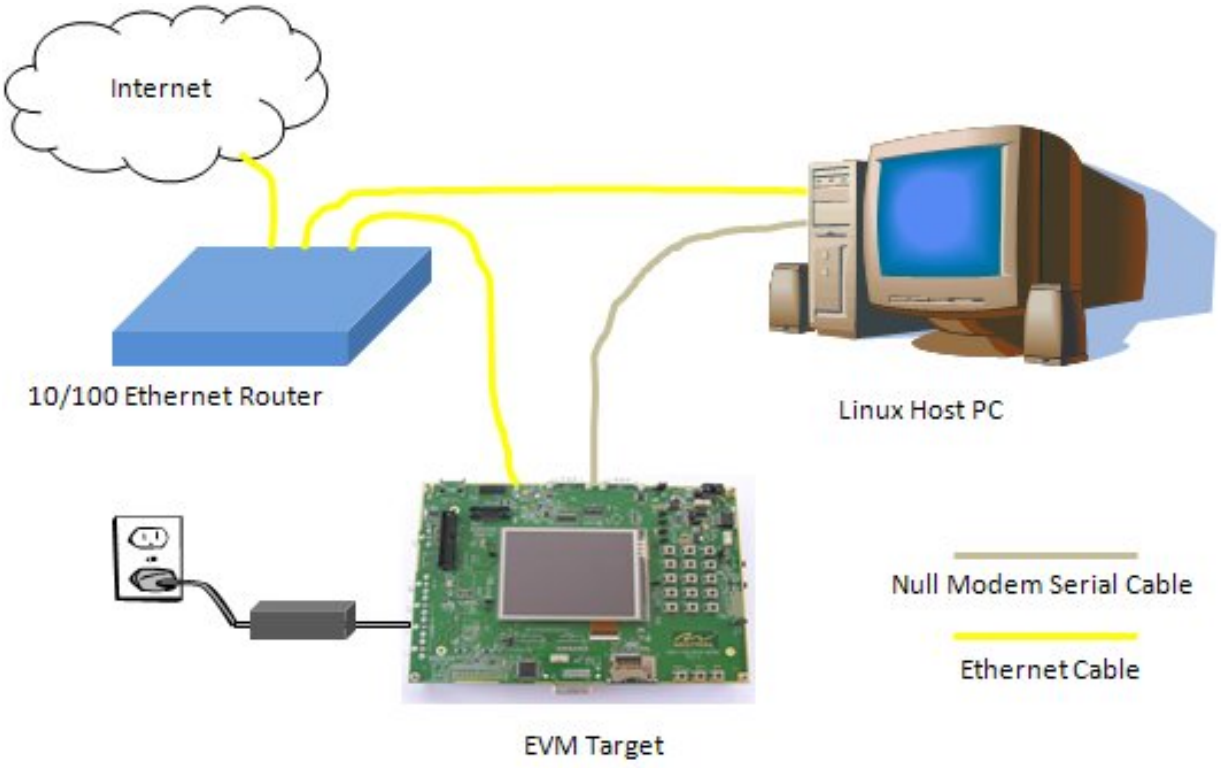
Before you can start using the Sitara Linux SDK you must have a Linux development environment. Linux development environments vary, so we have recommended the following for ease of use and consistency when working together. The next few steps will assist you in the setup & configuration of your Linux host based on the physical setup shown below.

Why Do I Need a Router

Routers offer convenience when working within a Linux development environment. This is especially true when working with a Network File System, discussed in more detail below. Using a router by default provides DHCP capabilities which in return ensures static IPs for your PC and EVM. This is important since you do not want the IP addresses changing on you without warning.

Why Do I Need a Null Modem Serial Cable

The common interface to the EVM or target when working within a Linux development environment is the Serial Terminal interface. Use of a serial terminal interface requires a null modem serial cable to be connected between the EVM and the Linux host machine. There are a number of different serial terminal clients that can then be installed and run on the Linux host machine. We'll talk more about this in the steps below.



1. Configure a Linux Host - If you already have a Linux host machine, go to Step 2. If you do not have a Linux host machine, you can configure a Linux host machine on your Windows PC using a virtual machine.

- → Build a Ubuntu based Linux host with **VMware** on WinXP (preferred)
- Build a Ubuntu based Linux host with **VirtualBox** on WinXP

2. Install the SDK & Toolchain - With a Linux host machine in place, you now need to install the toolchain used to build the SDK and then you need to install the SDK itself.

NOTE - If behind a company firewall, be sure to set the necessary proxy settings within Ubuntu 10.04 LTS

Install the CodeSourcery Lite Toolchain, **then**

Run the → Sitara SDK Installer

3. Run the Setup Script - Once the SDK has been installed a → Setup.sh Script has been provided to guide you through the remaining configuration including:

- Installation of required host packages
- Target FileSystem installation
- NFS setup
- TFTP setup
- Minicom setup
- uboot setup
- Load uboot script

After completing the setup.sh the EVM will be configured as you selected for your development environment.

Code Composer Studio v5 User's Guide

→ Code Composer Studio v5 (CCSv5) is provided with the Sitara Linux SDK. CCSv5 is an Integrated Development Environment (IDE) based on Eclipse and provides a graphical environment you can use to experiment with the example applications provided or the develop your own.

Matrix Application Launcher User's Guide

This → Matrix User's Guide provides an overview and details of the graphical user interface (GUI) and text user interface (TUI) implementations of the application launchers used by the EVM.

Developing with the Example Applications

There are a numbers of Example Applications provided within the Sitara SDK. Below are the User's Guides associated with the Example Applications.

- → Cryptography User's Guide
- → Matrix User's Guide
- USB Profiler User's Guide
- Power Manager User's Guide
- Multimedia User's Guide
- Camera User's Guide

WLAN & Bluetooth User's Guide

Applies to: AM37x EVM, AM180x EVM

AM37x EVM

If using the AM37x EVM, please refer to this AM37x Wireless LAN & Bluetooth User's Guide for more information on how to enable the WL1271 daughtercard which is connected to the EVM.

AM180x EVM

If using the AM180x EVM, please refer to this AM18x Wireless LAN & Bluetooth User's Guide for more information on how to enable the WL1271daughtercard which is connected to the EVM.

Profibus User's Guide

Applies to: AM1810 EVM only

This Profibus User's Guide (coming soon) provides an overview and details on how to use the Profibus support provided in the AM1810 EVM & SDK.

Rebuilding & Flashing the Bootloaders (X-loader & Uboot)

This information is provided by the PSP User's Guide and referenced here for your convenience.

X-Loader

- Compiling the AM35x/AM37x X-loader
- Compiling the AM18x/17x X-loader

Uboot

- Compiling and Flashing the AM35x/AM37x Uboot
- Compiling and Flashing the AM18x/17x Uboot

Rebuilding the Kernel

This information is provided by the PSP User's Guide and referenced here for your convenience.

- Compiling the AM35x/AM37x Linux Kernel
- Compiling the AM17x/AM18x Linux Kernel

Creating a SD Card

The SD card provided with the EVM may be duplicated by following the procedures outlined in How to Make a 3 Partition SD Card.

Pin Mux Utility User's Guide

Applies to: AM37x/DM37x, AM389x/C6A816x, OMAP35x

This → Pin Mux Utility User's Guide provides an overview and details on how to use the Pin Mux Utility for your development purposes.

For AM18x support only, please refer to this → Pin Setup Tool for AM18xx User's Guide.

Flash Tool User's Guide

Applies to: AM37x, OMAP3

This → Flash Tool User's Guide provides an overview and details on how to use the Flash Tool for your development purposes.

For AM18x support only, please refer to this → Flash Tool User's Guide

Use the procedure here to flash all system binaries to NAND flash from u-boot.

Error Correction (ECC) User's Guide

Applies to: AM37x, AM35x (PSP v03.00.01.06)

ECC support is now available in the xloader, uboot, kernel and Flash tool for microprocessors that have 4/8b ECC memory devices, like those mentioned above.

Details on how to utilize ECC within each of these components is provided in the → ECC User's Guide.

Other How Tos

This section provides how-to articles on additional setups and configurations you may find useful in your development.

- Developing with 3D Graphics
- How to Connect to an EVM via Telnet
- How to Setup a Samba Server
- Understanding the Boot Sequence
- How to Move Files From Host to Target
- How to Flash the Linux System to NAND from U-boot
- → How to use a Mouse instead of the Touchscreen with Matrix
- How to enable DVI display

DVI display can be enabled in two ways

1. Using bootargs

```
 Eg: omapdss.def_disp="dvi" omapfb.mode=dvi:720x480@60
```

2. Using SYSFS attributes

```
echo 0 > /sys/devices/platform/omapdss/display0/enabled
echo "" > /sys/devices/platform/omapdss/manager0/display
fbset -fb /dev/fb0 -xres 720 -yres 480
echo "dvi" > /sys/devices/platform/omapdss/manager0/display
echo 1 > /sys/devices/platform/omapdss/display2/enabled
kill <matrix_gui PID>
Restart without enabling rotation: matrix_gui -qws
/usr/share/matrix/html/menu_main.html
```

Reference Documentation

SDK Migration Guide

- Sitara Linux SDK Migration Guide

Archived - Software Developer's Guide

- Sitara SDK 4.00 - Software Developer's Guide (archived) [2]

Release Notes

- Sitara SDK 4.01 Release Notes
- Sitara SDK 4.00 Release Notes
- WLAN/BT Release Notes & Downloads

Graphics

- Graphics SDK Getting Started Guide

PSP Documentation

- AM35x-OMAP35x-PSP 03.00.01.06 Feature Performance Guide
- AM35x-OMAP35x-PSP 03.00.01.06 Release Notes
- AM35x-OMAP35x-PSP 03.00.01.06 UserGuide

Others

- Linux Glossary

Software Updates

We are continually improving the quality and content of the software we provide in the EVM. Updates to the SDK may be obtained at Software Updates [3] as they become available.

Technical Support

- E2E Support Forums [4] - an active community of TIers and other customer like you already using the AM37x EVM. You may find your question has already been answer with a quick Search of the Forums. If not, a quick post will likely provide you the answers you need. Support@ti.com - a support email list you may submit your question to.
- support@ti.com [5]

Feedback

We are always striving to improve this product. Please let us know if you have ideas or suggestions by clicking "Leave a Comment..." below.

References

[1] http://www.mistralsolutions.com/assets/downloads/AM37x_EVM.php
[2] http://processors.wiki.ti.com/index.php?title=Sitara_Linux_Software_Developer%E2%80%99s_Guide&oldid=43627
[3] http://www.ti.com/sitara
[4] http://e2e.ti.com/support/dsp/sitara_arm174_microprocessors/default.aspx
[5] mailto:support@ti.com

How to Build a Ubuntu Linux host under VMware

Introduction

This guide demonstrates how to get a virtual Ubuntu Linux machine running with VMware under Windows XP. Please use only the Ubuntu 10.04 release as this is what is called an LTS (Long Term Support). There are SDK scripts that will be checking for this release identity.

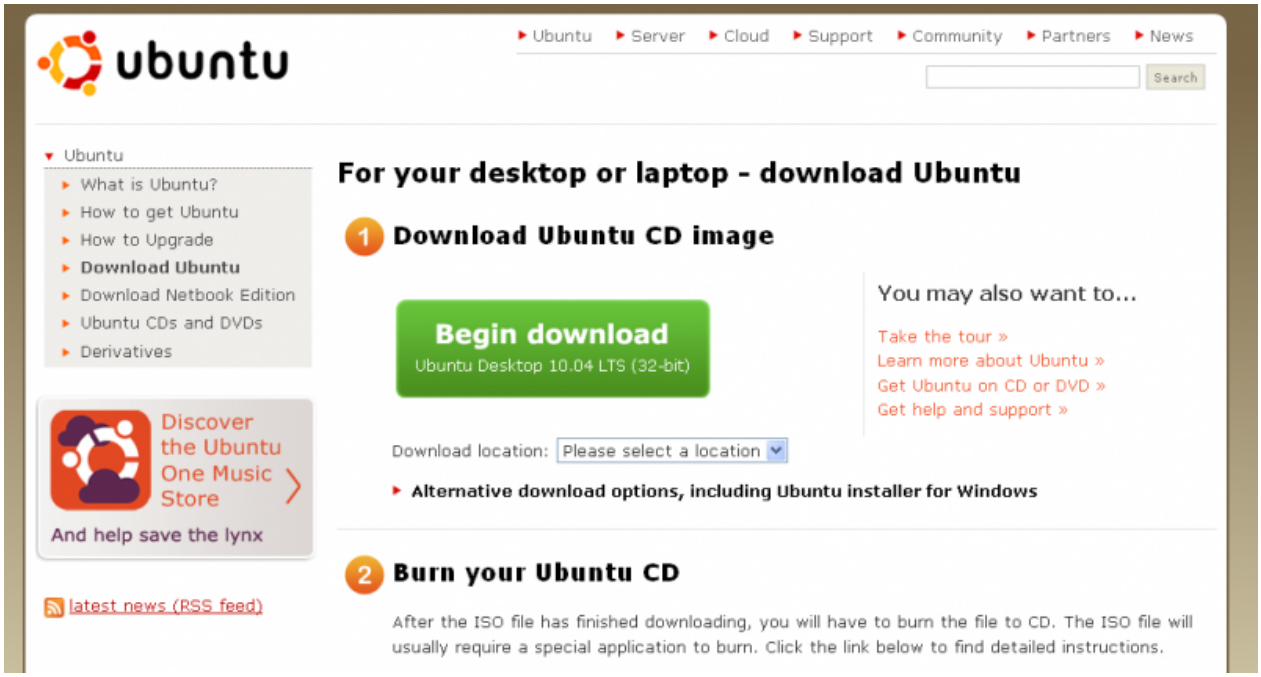
Requirements:

- Windows XP host with internet connection, at least 1G of RAM and 40G of free hard drive space.

The instructions here are for setting up a 40G virtual machine. The entire 40G is not taken at once, but as the machine is used and software is installed, the machine can grow and take up as much as 40G.

Download the Ubuntu 10.04 LTS ISO image

Get the Ubuntu 10.04 LTS CD ISO image from: http://www.ubuntu.com [1]



Click download and the follow instructions to download and save the ISO image (CD image).

Download VMware and install

Get VMware from: http://www.vmware.com [2]

Vmware Player is a free download from the website and enables the user to create an entire virtual machine from scratch using just the ISO image downloaded from Ubuntu. It is necessary to sign up for an account at VMware in order to get to the download areas. The general steps to getting VMware are as follows:

- Login to the vmware website
- Select VMware Player from the products menu
- Follow the steps to download VMware Player

NOTE - We have tested with v3.0.1 and v3.1.2 with no known issues. As of Oct 26, 2010, v3.1.2 is the latest version.

- Run the executable to install VMware
- Accept license and all default settings.

After Vmware is installed the Windows host will have two new (virtual) network adapters. These can be seen in the Windows host by looking under Control Panel --> Network Connection. No additional setup is needed for these new adapters.

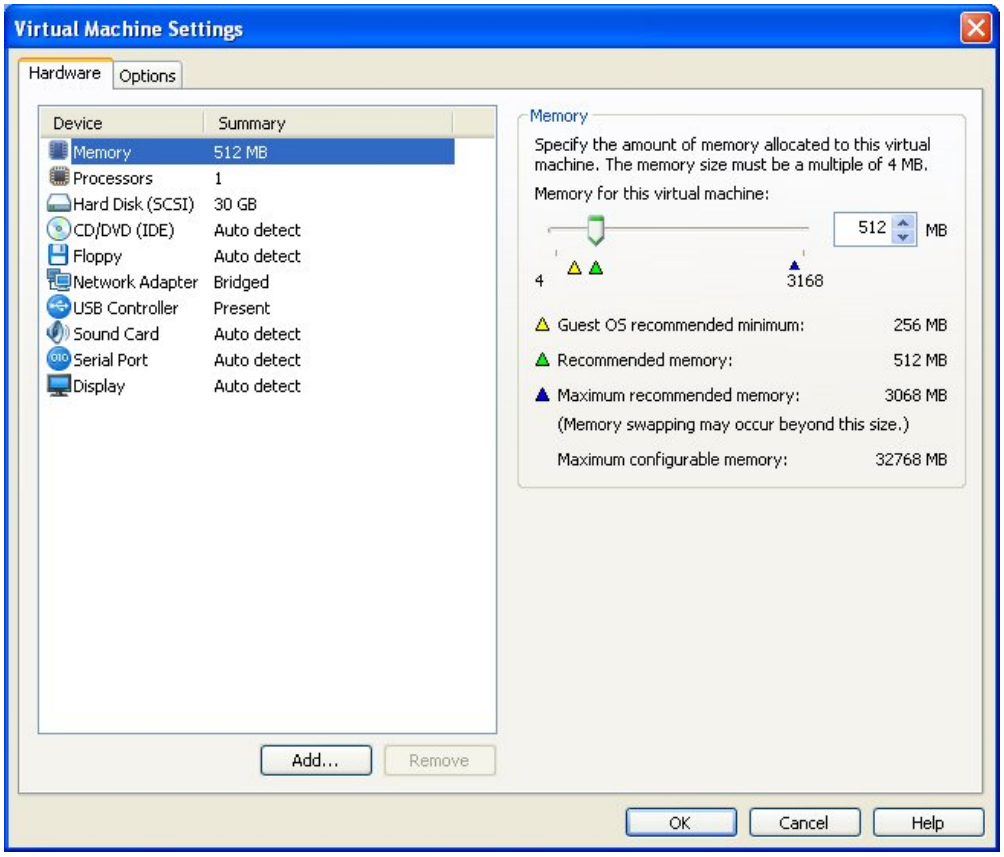
Create a New Virtual Machine with VMware

Before starting a new installation it is assumed that the Windows host has a proper internet connection to a DHCP server and that the Windows host has enough hard drive space for the new virtual machine.

The following steps are performed with VMware 3.0.1. The exact steps with other versions may vary slightly

- Start VMware.
- From the File menu select "Create a New Virtual Machine..."
- Choose to install the operating system later. Click "Next".
- Select Linux as the "Guest Operating System" and then choose Ubuntu as the "Version". Click "next".
- Provide a "Virtual machine name" and "Location" where the machine will be stored on the Windows host. The defaults are fine here. Click "Next".
- For "Maximum disk size (GB)" it is good to start with 40G if possible. This means that it will take up 40G on the Windows host. Make sure that the Windows host has at least this much before proceeding. It is also a good practice to tell VMware to split the virtual disk into 2G files. This will makes the image easier to copy and transport if necessary. Click "Next".
- Click "Finish" to complete the creation of the virtual machine.

The machine name will now be listed under the home page of VMware. It is necessary to modify some machine settings before playing the machine for the first time. Select the machine in the home page. Under the "VM" menu select "Settings...".



Click on CD/DVD and change the connection to "Use ISO image file". Click on "Browse..." and select the Ubuntu ISO image file that was previously downloaded. Click on Network Adapter and change the Network connection to "Bridged" and then check the box to "Replicate physical network connection".

Adding a serial port to the virtual machine

If you plan to use a serial terminal application, a serial port must be added to the virtual machine. This port must be a physical serial port which exists on the host PC. Click on "Add..." and select "Serial Port". Click "Next". Choose "Use physical serial port on host". Click "Next". Click Finish. Click "Ok".

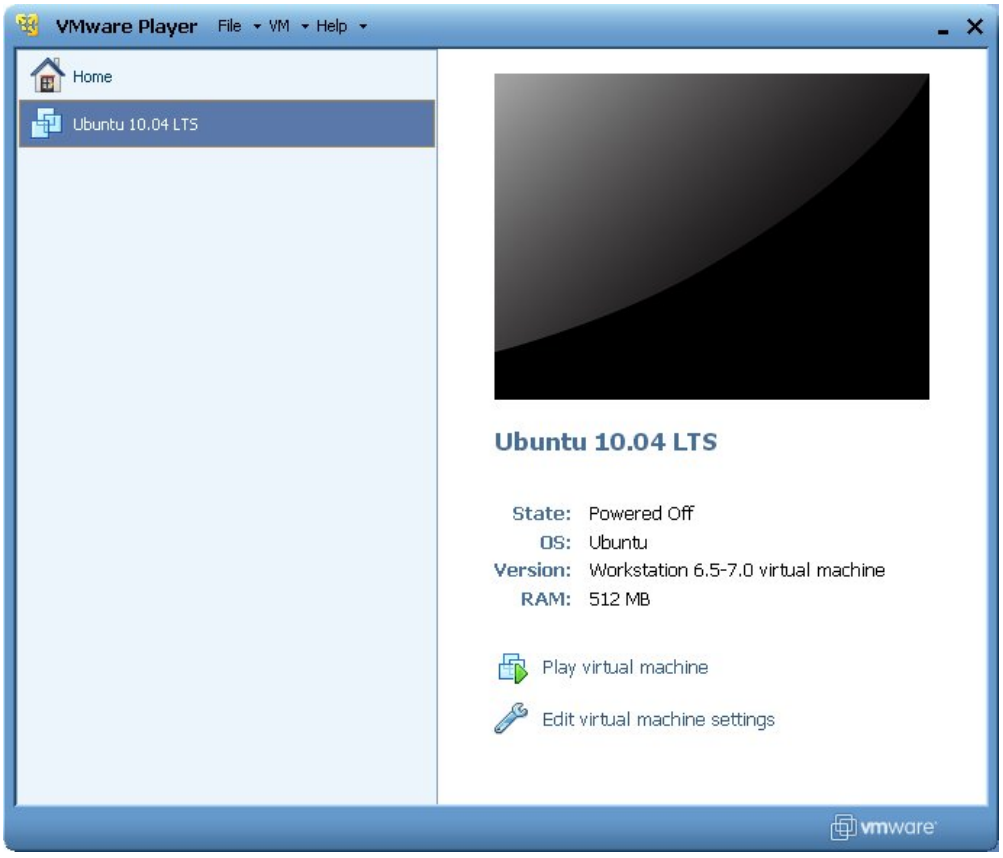
Since this is a physical port on the host PC, it cannot be used by the host PC and the virtual machine at the same time. When the virtual machine is started, the serial port will be unavailable for use by the host PC. If the serial port is being used at the time that the virtual machine is started, the virtual machine will not be able to access the serial port after it is booted up. So if you want the virtual machine to gain control of the physical serial port of the host PC, there can not be any application like hyperterminal or teraterm running on the host PC at the time that the virtual machine is started.

Further instructions for using the serial port with minicom inside of Ubuntu are here [3].

Minicom is the preferred application for use with the Sitara SDK. And the installation and setup of minicom is done automattically by the Sitara SDK installer.

Now click on "Play virtual machine". Since this is the first time starting the machine and the Ubuntu ISO image is in the virtual CD drive, the Ubuntu OS will install itself in the virtual machine.

Click through the Ubuntu installation, making the appropriate choices as you go. When prompted for a login name make the login name **user**. This will help with SDK installation scripts.



The full installation will take 20-30 minutes. When it completes the machine will reboot. The machine will now prompt for the login (**user**) and password.

After the machine reboots into Ubuntu it is helpful to take the Ubuntu ISO out of the virtual CD drive. Click on the VM menu and select "Settings...". Click on CD/DVD and change the connection from "Use ISO image file" to "Use physical drive". The actual drive letter can be selected from the drop down list. If there is no physical drive on the host machine, the CD/DVD device can be simply removed from the machine.

Install VMware Tools

VMware tools is a very useful addition to VMware. It allows you to resize the VMware screen and also allows cut-and-paste of text from the Ubuntu machine to and from the Windows host.

Click on the VM menu. Select "Install VMware Tools". Vmware will automatically mount a drive with the VMware Tools tarball. There will be instructions at the bottom of the screen. Follow the instructions to install VMware Tools. Select all of defaults during installation of VMware Tools.

Confirming a Valid Network Connection

After logging into the machine for the first time, bring up a terminal window. This can be found under the Applications menu in Ubuntu. Applications --> Accessories --> Terminal. Type **pwd** in this terminal. This should return **/home/user**. Now type **ifconfig**. This should return information about the network connection. Under **eth0** the IP address should be similar (but not the same) as the IP address owned by the Windows host.

```
user@Ubuntu1004:~$ pwd
/home/user
user@Ubuntu1004:~$ ifconfig
eth0 Link encap:Ethernet HWaddr 00:0c:29:da:a8:6e
inet addr:128.247.107.65 Bcast:128.247.107.255 Mask:255.255.254.0
inet6 addr: fe80::20c:29ff:feda:a86e/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:759 errors:0 dropped:0 overruns:0 frame:0
TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:62873 (62.8 KB) TX bytes:4937 (4.9 KB)
Interrupt:19 Base address:0x2024

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:16436 Metric:1
RX packets:12 errors:0 dropped:0 overruns:0 frame:0
TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:720 (720.0 B) TX bytes:720 (720.0 B)
user@Ubuntu1004:~$
```

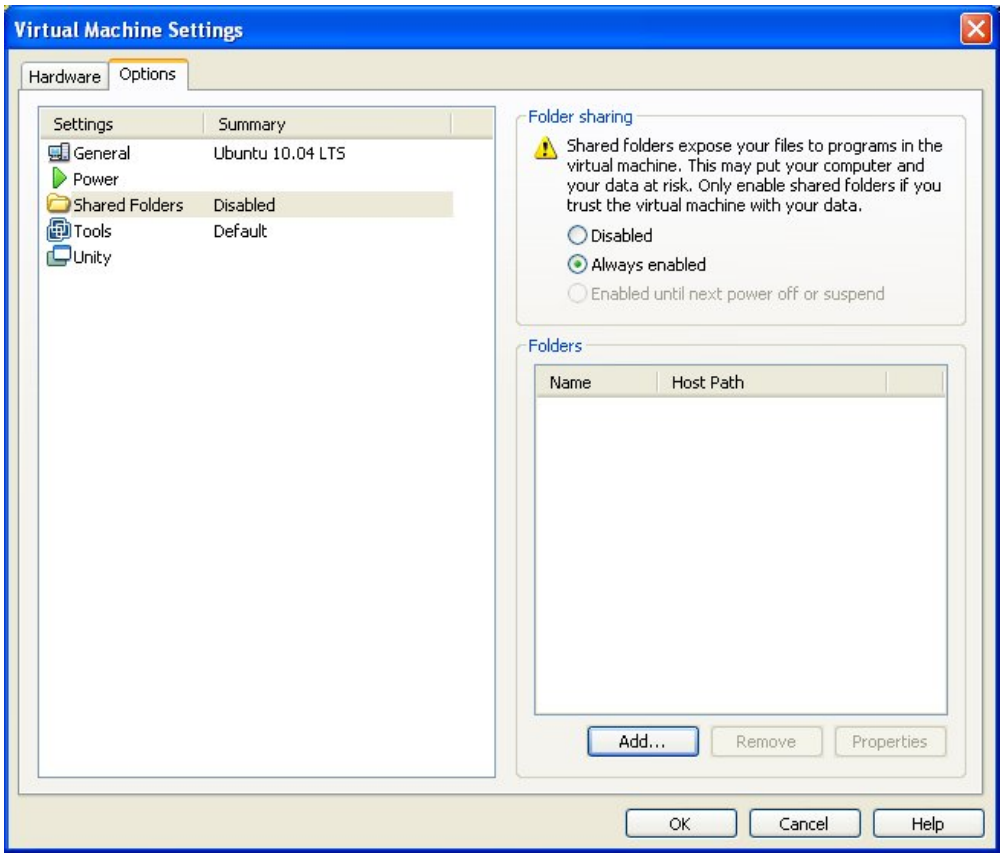
How to Read a USB SD Card Reader in VMware

When a USB card reader with an SD card is inserted into the USB slot of the host machine, the virtual machine will automatically detect the drive and mount partitions from the SD card. If this does not happen automatically it can be done manually by clicking the VM menu and selecting Removable Devices and then selecting the card reader from the sub-menu under Removable Devices. From this sub-menu it is possible to connect or disconnect the USB card reader.

How to Set up Shared Folder in VMWare

The following steps show how to enable Shared Folders within VMware which allows you to easily share files between Ubuntu 10.04 running on VMware and your Windows host.

1. **Under Virtual Machine Settings, the Options tab, select *Shared Folders* and *Always enabled*.**



2. **Click Add...and the following dialog should display.**

3. Browse the Windows folder your want to Share. And provide a Name for that folder. This Name is what Ubuntu 10.04 LTS will mount to.

4. **Ensure Enable this share is checked (should be default).** And click Finish.

5. **Start your virtual machine and log into Ubuntu 10.04 LTS.** Create a Desktop short-cut to the Shared Folder you just set up.

- Right click on your Ubuntu 10.04 LTS desktop
- Click *Create Launcher...*
- Change Type to *Location*
- Enter a *Name*
- Next to Location:, enter */mnt/hgfs/**shared_folder_name***

NOTE - shared_folder_name is what ever name your provide in Settings

After clicking OK, you should have a desktop shortcut to your Shared Folder.

Configuring a Proxy in Ubuntu

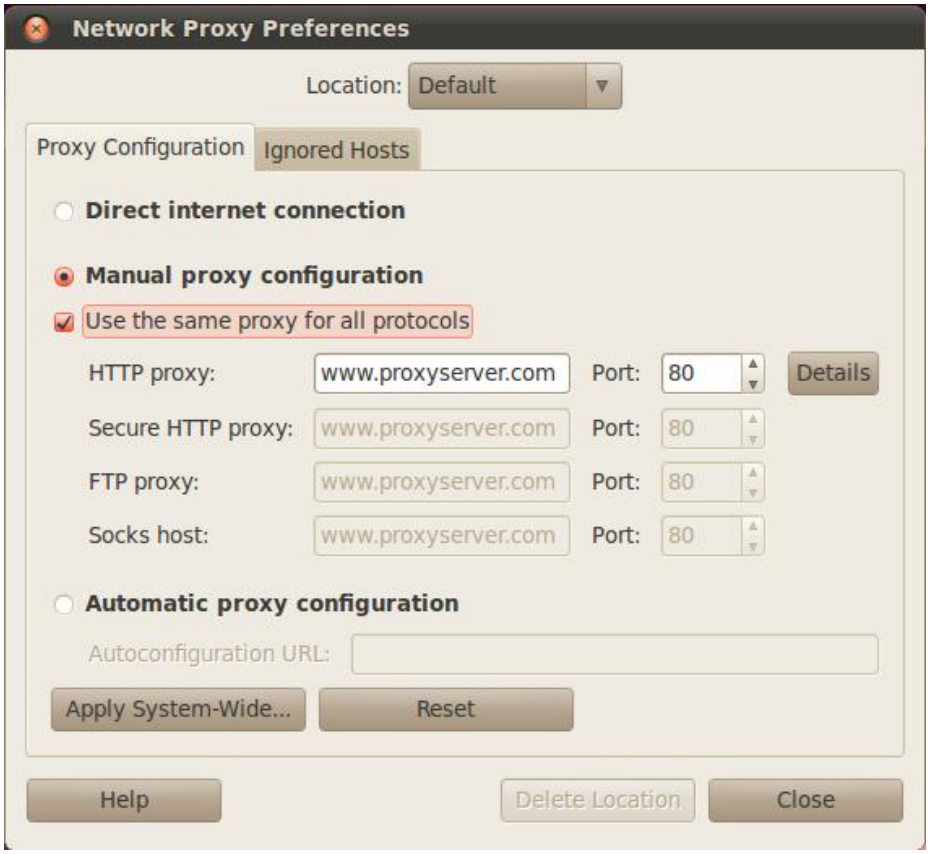
If your network is behind a firewall you will need to configure the network proxy for Ubuntu in order to successfully download the applications required to complete your development environment or to browse the Internet on your Linux Host machine.

To configure the network proxy in Ubuntu go to System-Preferences and click Network Proxy as seen below.

As seen in the image below, click Manual Proxy Configuration. Specify the HTTP proxy server used by your company. You may find this information under your Windows OS inside the Internet Explorer Network Connections. Be sure to specify the port.

NOTE - wwww.proxyserver.com is not a valid HTTP proxy. It is shown as an example only. You need use the HTTP proxy server & Port used by your company.

Make sure you check "Use the same proxy for all protocols". Also be sure to click "Apply System-Wide". Then Close



Finally you will be asked to entered your password in order to set your network proxy information. This is typically the same password you used to log into Ubuntu on boot.

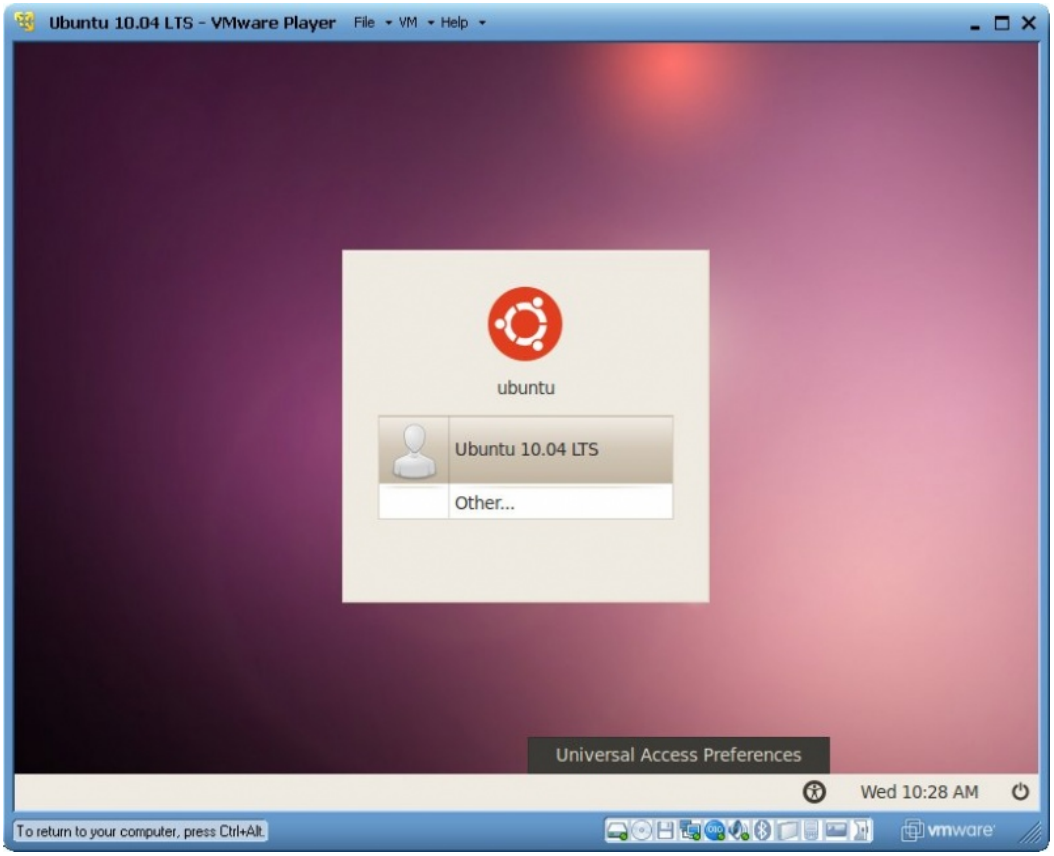
You should now be able to browse the Internet using Firefox within Ubuntu.

Keyboard does not Work in Ubuntu 10.04 LTS

If your physical keyboard does not work when you first start VMWare running Ubuntu 10.04 LTS, then you may want to try the following steps to resolve the issue. This assumes your mouse works properly.

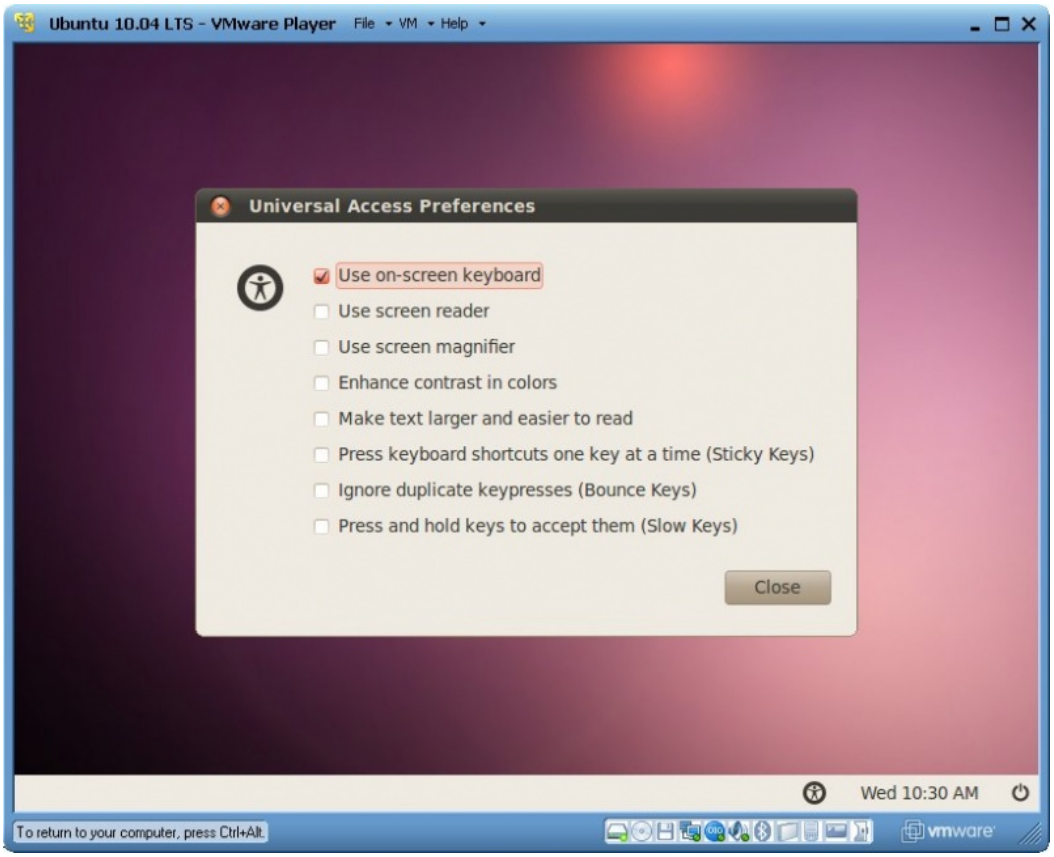
NOTE - The following makes use of the on-screen keyboard provided in Ubuntu 10.04 LTS. Once you are able to log into Ubuntu 10.04 LTS using the on-screen keyboard, issues with the physical keyboard should be resolved.

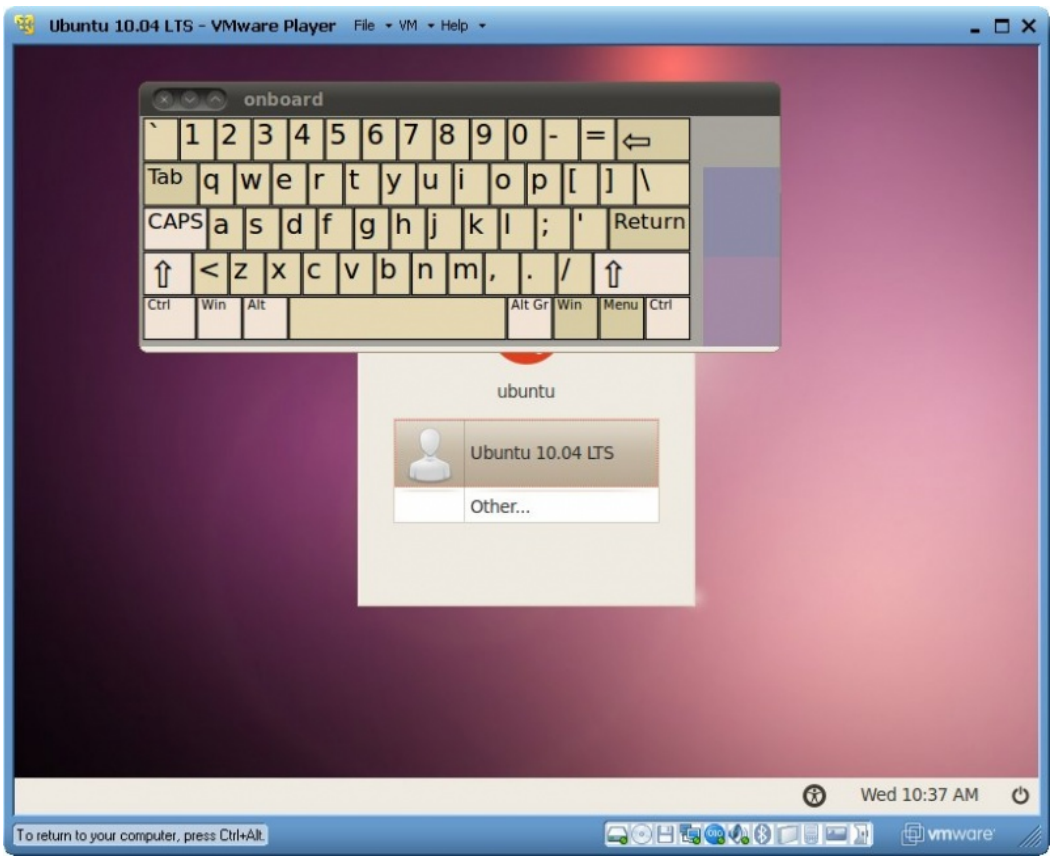
1. Open the *Universal Access Preferences* by left clicking on the icon with the man in the circle. This will display the *Universal Access Preferences* button. Click on *Universal Access Preferences*.



2. Select the first option to *Use on-screen keyboard*. Once you *Close*, the on-screen keyboard should display allowing you to log into Ubuntu 10.04 LTS.

NOTE - If the on-screen keyboard does not display, *Shutdown* Ubuntu 10.04 LTS and restart your Ubuntu 10.04 LTS virtual machine again in VMWare. Once Ubuntu 10.04 starts again the on-screen keyboard should be displayed.





3. Modify /etc/default/console-setup using gedit to enable the physical keyboard

```
username@ubuntu:~$ sudo gedit /etc/default/console-setup
```

NOTE - You will need sudo access and therefore must enter your password to access this file.

At the bottom change the following lines:

```
<original>
XKBMODEL=”SKIP”
XKBLAYOUT=”us”
XKBVARIANT=”U.S. English”
XKBOPTIONS=""

<new changes>
XKBMODEL=”pc105”
XKBLAYOUT=”us”
XKBVARIANT=""
XKBOPTIONS=""
```

The physical keyboard should now work from here out. You may also now disable the on-screen keybaord under *Universal Access Preferences*.

References

[1] http://www.ubuntu.com
[2] http://www.vmware.com
[3] http://processors.wiki.ti.com/index.php/Setting_up_Minicom_in_Ubuntu

Installing CodeSourcery Toolchain

How to Obtain the CodeSourcery Lite Edition

This section walks through the steps required to install the Code Sourcery Lite toolchain for ARM GNU/Linux EABI processors. The specific toolchain release is 2009q1-203.

NOTE: Download the "IA32 GNU/Linux TAR" tarball. It is significantly smaller and takes less time to download.

Download CS Lite 2009q1-203 [1]

When installing please remember the location you installed this toolchain. If your EVM comes with a SDK Installer it will check for a valid toolchain.

How to Obtain the CodeSourcery Commercial Edition

CodeSourcery also provides commercial edition that contains an IDE, additional tools and support from CodeSourcery. See CodeSourcery offerings for Davinci™, OMAP™ and Sitara™ processors here [2].

A DVD containing both CodeSourcery G++ Lite and Evaluation version of commercial tools are shipped with the AM37x Evaluation Module [3].

NOTE - When using the arm_2009q1_203_arm_none_linux_gnueabi.bin instead of the TAR, the toolchain path is /home/user/CodeSourcery/Sourcery_G++_Lite (assumes default). To set path for bash shell, edit .bashrc file by adding the lin:

```
export PATH=/home/user/CodeSourcery/Sourcery_G++_Lite/bin:$PATH
```

References

[1] http://www.codesourcery.com/gnu_toolchains/arm/portal/release858
[2] http://www.codesourcery.com/sgpp/partners/omap
[3] http://focus.ti.com/docs/toolsw/folders/print/tmdxevm3715.html

Sitara SDK Installer

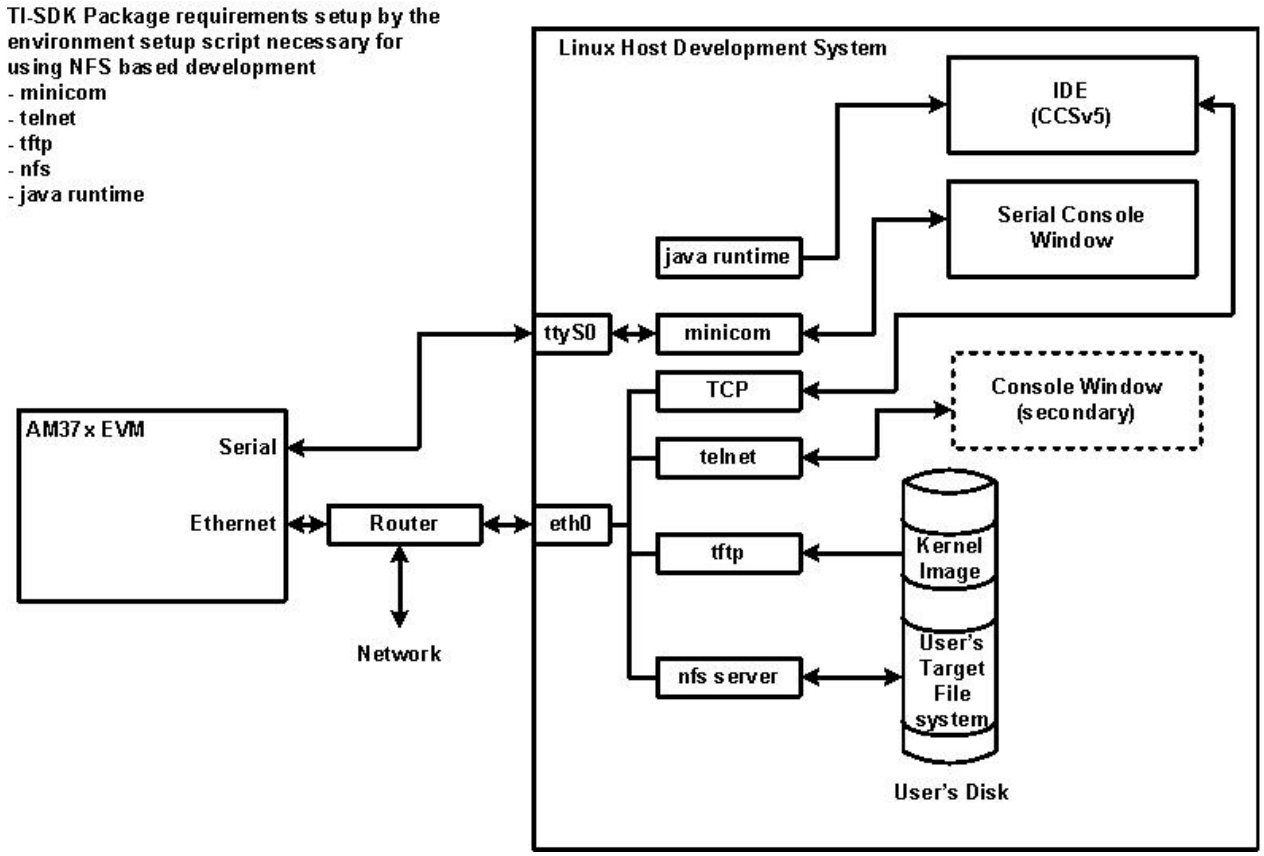
Overview

The SDK Installer (ti-sdk-amxx-vx.x.x.x) will install the necessary components to start your development on the AMxx microprocessor. The SDK consists of source for the Matrix App launcher starting point application, a development filesystem, a target filesystem, an IDE (CCSv5), Pin Mux Utility, Flash Tool applications, the PSP and documentation.

Please note the user must first have installed the Code Sourcery tools necessary to compile and link. Please refer to Installing CodeSourcery [1] for more information on how to install the tools.

The ti-sdk was built and tested against a specific Linux Distribution name and version, Ubuntu 10.04. Note this **does not** prevent the user from installing the SDK on other Linux distributions.

To assist the user in getting a to a point of starting development the installer contains an Environment Setup Script [2] that will run with the Linux Distribution specified for the ti-sdk. This particular script sets up several functions needed by the SDK such as the Java runtime for the CCSv5 IDE. While it is written for the current distribution the user can modify the script to fit their particular distribution. Please see the picture for a block diagram of the development environment setup for ti-sdk. Please also note that this script is specifically tied to Ubuntu 10.04 as will make specific package installations that are only known to work on this release.



SDK Installer Execution Steps

1. **Confirm**

User is to confirm if loading the ti-sdk is ok. This is important to note in the user is trying to over-install on an existing directory and has made changes to the directory.

2. **Toolchain Verification**

The Code Sourcery toolchain is required and the installer will perform a verification that the correct package is installed based on user input. An example is given below.

3. **Directory Install Location**

The user will be prompted for a location on where to put the ti-sdk. An example is given below.

4. **Installation of software**

The software is installed.

5. **Environment Setup Script**

If the user is running the supported Linux distribution they will have the option of running the environment script that will set download from Ubuntu several packages that will enable tftp, nfs, serial console and the IDE. This is shown below. This script is run after the installer completes and should only be run once. If the user decides to re-install the environment setup script does not need to be run again.

How to Run the SDK Installer

The SDK Installer is found on the START_HERE partition of the SD card in the EVM box. Run the SDK Installer by double clicking on it within your Linux host PC.

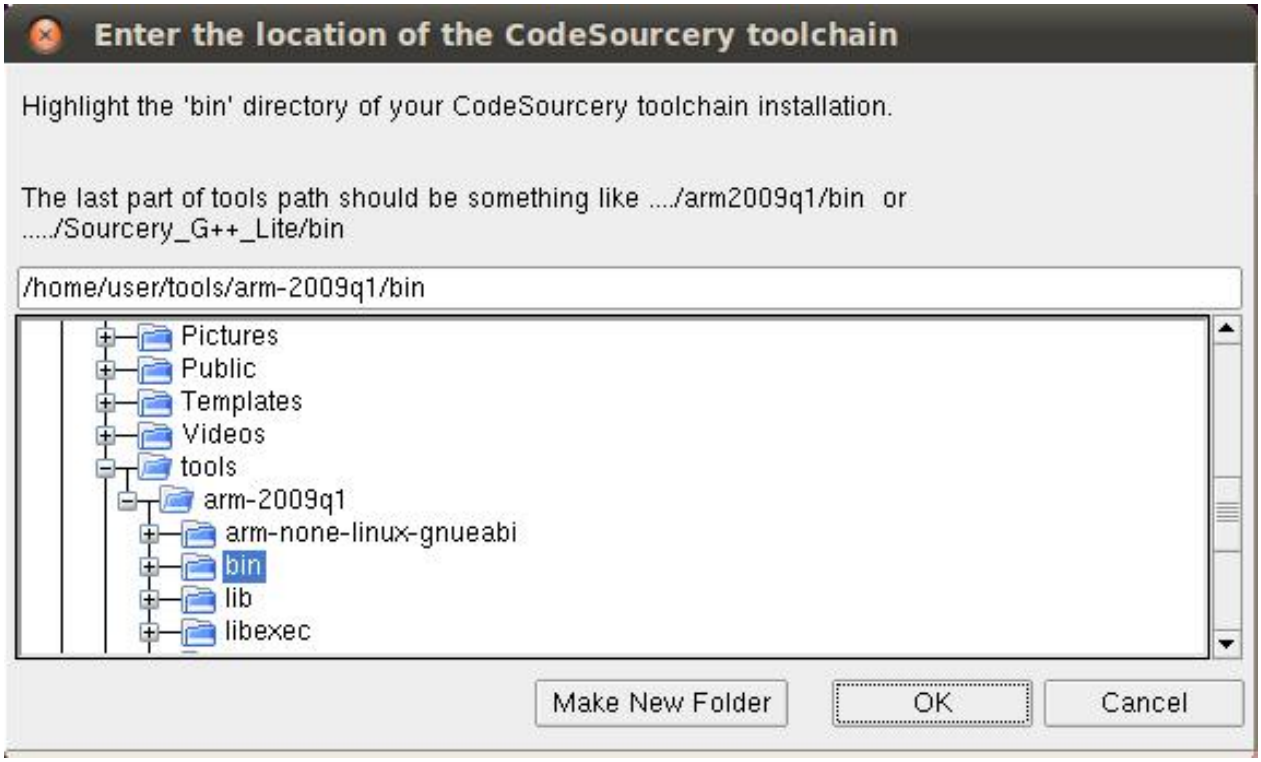
Alternatively, bring up a terminal window and change directories to the where the START_HERE partition is mounted and run the SDK Installer with the following command:

```
[prompt]:~$  ./ti-sdk-amxx-evm-x.x.x.x-Linux-x86-Install
```

Toolchain Verification

The installer will ask the user to verify the version of the Code Sourcery toolchain. The location that the user inputs will be used by the installer to setup paths used by the included software projects to build the projects.
Below are some of the series of panes that the user will see and use to determine the installed toolchain.
This first pane is provided information on the version of the toolchain required and where to find it if the user does not have this particular version.

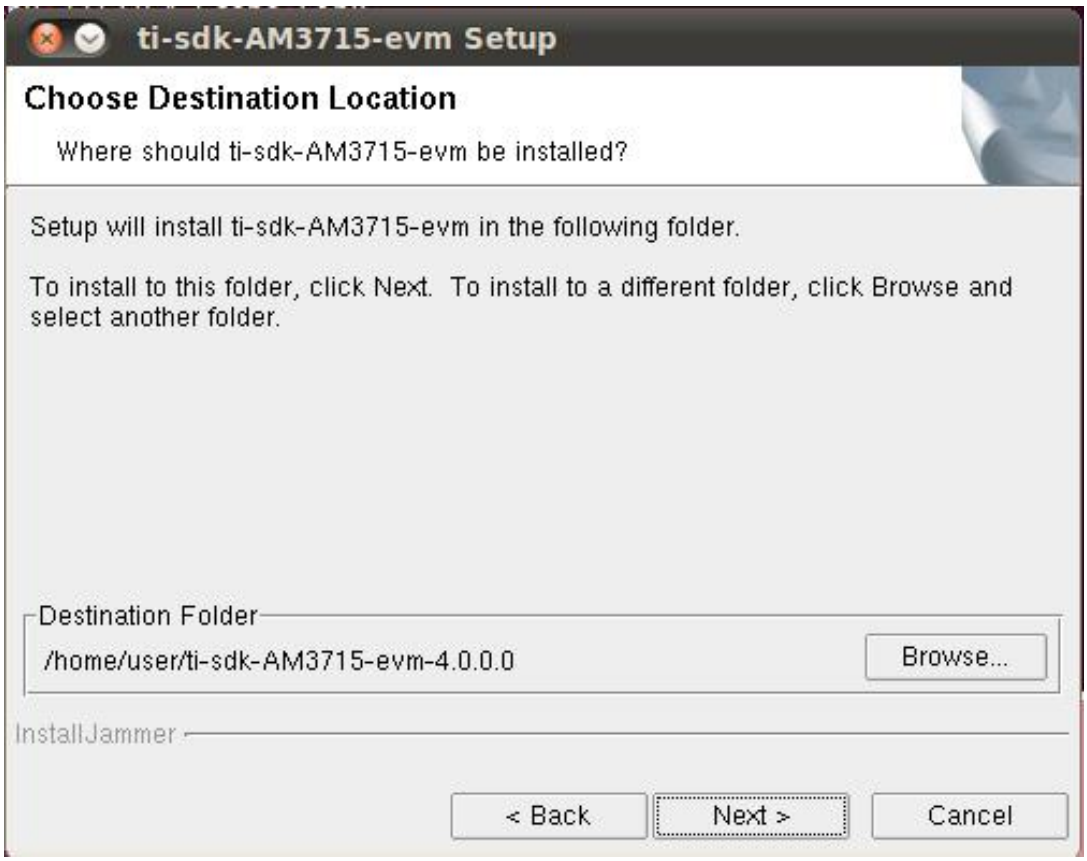
This next pane the user indicates where their toolchain is installed. This path is then installed in the user's .profile and other files used by the included projects and makefiles.



This is the message returned to the user if the user has found the correct version of the tools.

Where to install the ti-sdk package

The default selection of where to install is the user's home directory. In this particular example the name of the user is user.



Code Composer Studio v5 IDE Installer

If you checked the "Install Code Composer" checkbox in the SDK Installer, the CCSv5 installer will get kicked off prior to exiting from the SDK installer. The wiki page Running_the_CCSv5_installer [3] provides information about the CCSv5 installer.

References

[1] http://processors.wiki.ti.com/index.php/Installing_CodeSourcery_Lite
[2] http://processors.wiki.ti.com/index.php/SDK_Setup_Script
[3] http://processors.wiki.ti.com/index.php/Code_Composer_Studio_v5_Preview_Edition_Users_Guide#Running_the_CCSv5_installer

SDK Setup Script

Overview

One of the last panes of the SDK Installer recommends the environment Setup Script (setup.sh)

```
--------------------------------------------------------------------------------
TISDK setup script

This script will set up your development host for sdk development.
Parts of this script require administrator priviliges (sudo access).
--------------------------------------------------------------------------------
```

NOTE - The Setup Script will first check to see if the user is running on the Ubuntu 10.04 distribution, if not it will exit. If the user is running on a different distribution they are encouraged to modify the environment setup script to match their distribution.

```
--------------------------------------------------------------------------------
Verifying Linux host distribution
Ubuntu 10.04 LTS found successfully, continuing..
--------------------------------------------------------------------------------
```

The Setup Script is located in the top level of the ti-sdk install directory. To run the script:

```
[prompt]:~$ sudo ./setup.sh
```

Alternatively, you can double click *setup.sh* from ti-sdk install directory and click **Run In Terminal**.

After validating Ubuntu 10.04 LTS, the Setup Script will proceed to guide the user through configuration of the following:

- Installation of required host packages
- Target FileSystem installation
- NFS setup
- TFTP setup
- Minicom setup
- uboot setup
- Load uboot script

Installation of Required Host Packages

```
--------------------------------------------------------------------------------
This step will make sure you have the proper host support packages
installed
using the following command: sudo apt-get install xinetd tftpd
nfs-kernel-server minicom build-essential libncurses5-dev uboot-mkimage
 autoconf automake
```

```
Note! This command requires you to have administrator priviliges (sudo
access)
on your host.
Press return to continue


Reading package lists... Done
Building dependency tree
Reading state information... Done
xinetd is already the newest version.
tftpd is already the newest version.
nfs-kernel-server is already the newest version.
minicom is already the newest version.
build-essential is already the newest version.
libncurses5-dev is already the newest version.
uboot-mkimage is already the newest version.
autoconf is already the newest version.
automake is already the newest version.
The following packages were automatically installed and are no longer
required:
  linux-headers-2.6.32-21 linux-headers-2.6.32-21-generic
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.

Package verification and installation successfully completed
-------------------------------------------------------------------------------
```

Target FileSystem Installation

```
-------------------------------------------------------------------------------
In which directory do you want to install the target filesystem?(if
this directory does not exist it will be created)
[ /home/user/ti-sdk-amxx-evm-x.x.x.x/targetNFS ]

-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
This step will extract the target filesystem to
/home/user/ti-sdk-amxx-evm-x.x.x.x/targetNFS

Note! This command requires you to have administrator priviliges (sudo
access)
on your host.
Press return to continue

/home/user/ti-sdk-amxx-evm-x.x.x.x/targetNFS already exists
(r) rename existing filesystem (o) overwrite existing filesystem (s)
```

```
skip filesystem extraction
[r]

Successfully extracted tisdk-rootfs-amxx-evm.tar.gz to
/home/user/ti-sdk-amxx-evm-x.x.x.x/targetNFS
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
This step will set up the SDK to install binaries in to:
    /home/user/ti-sdk-amxx-evm-x.x.x.x/targetNFS/home/root/am181x

The files will be available from /home/root/amxx on the target.

This setting can be changed later by editing Rules.make and changing
the
EXEC_DIR variable.

Press return to continue

Rules.make edited successfully..
-------------------------------------------------------------------------------
```

NFS Setup

```
-------------------------------------------------------------------------------
This step will export your target filesystem for NFS access.

Note! This command requires you to have administrator priviliges (sudo
access)
on your host.
Press return to continue

/home/user/ti-sdk-amxx-evm-x.x.x.x/targetNFS already NFS exported,
skipping..

 * Stopping NFS kernel daemon
                                                [ OK ]
 * Unexporting directories for NFS kernel daemon...
                       [ OK ]
 * Exporting directories for NFS kernel daemon...
                         [ OK ]
 * Starting NFS kernel daemon
                                                [ OK ]
-------------------------------------------------------------------------------
```

TFTP Setup

```
-------------------------------------------------------------------------------
Which directory do you want to be your tftp root directory?(if this
directory does not exist it will be created for you)
[ /tftpboot ]

-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
This step will set up the tftp server in the /tftpboot directory.

Note! This command requires you to have administrator priviliges (sudo
access)
on your host.
Press return to continue



/tftpboot already exists, not creating..

/tftpboot/uImage already exists. The new installed file can be renamed
and saved under the new name.
(r) rename (o) overwrite (s) skip copy
[r]

Successfully overwritten uImage in tftp root directory /tftpboot

/etc/xinetd.d/tftp already exists..
/tftpboot already exported for TFTP, skipping..

Restarting tftp server
 * Stopping internet superserver xinetd
                                  [ OK ]
 * Starting internet superserver xinetd
                                  [ OK ]
-------------------------------------------------------------------------------
```

Minicom Setup

```
-------------------------------------------------------------------------------
This step will set up minicom (serial communication application) for
SDK development

Which serial port do you want to use with minicom?
[ /dev/ttyS0 ]

Copied existing /home/user/.minirc.dfl to /home/user/.minirc.dfl.old
```

```
Configuration saved to /home/user/.minirc.dfl. You can change it
further from inside
minicom, see the Software Development Guide for more information.
--------------------------------------------------------------------------------
```

uboot Setup

```
--------------------------------------------------------------------------------
This step will set up the u-boot variables for booting the EVM.
Autodetected the following ip address of your host, correct it if
necessary
[ xx.xx.xx.xx ]

Select board memory:
 1: 128MB
 2:  64MB

[ 1 ]

Select Linux kernel location:
 1: TFTP
 2: SD card
 3: flash

[ 1 ]

Select root file system location:
 1: NFS
 2: SD card

[ 1 ]


Available kernel images in /tftpboot:
    uImage
    uImage-2.6.33.7-rt29

Which kernel image do you want to boot from TFTP?
[ uImage ]

Resulting u-boot variable settings:

setenv bootdelay 3
setenv baudrate 115200
setenv bootargs console=ttyS2,115200n8 noinitrd rw mem=64M@0xc4000000
ip=dhcp
setenv bootcmd 'dhcp;setenv serverip xx.xx.xx.xx;tftpboot;bootm'
setenv autoload no
```

```
setenv serverip xx.xx.xx.xx
setenv bootfile uImage
--------------------------------------------------------------------------------
```

Load uboot Script

```
--------------------------------------------------------------------------------
Would you like to create a minicom script with the above parameters
(y/n)?
[ y ]

Successfully wrote
/home/user/ti-sdk-amxx-evm-x.x.x.x/setup_uimage-tftp_fs-nfs.minicom

Would you like to run the setup script now (y/n)? This requires you to
connect
the RS-232 cable between your host and EVM as well as your ethernet
cable as
described in the Quick Start Guide. Once answering 'y' on the prompt
below
you will have 300 seconds to connect the board and power cycle it
before the setup times out.

After successfully executing this script, your EVM will be set up. You
will be
able to connect to it by executing 'minicom -w' or if you prefer a
windows host
you can set up Tera Term as explained in the Software Developer's
Guide.
If you connect minicom or Tera Term and power cycle the board Linux
will boot.

[ y ]
```

Code Composer Studio v5 Preview Edition Users Guide

Overview

Code Composer Studio v5 Limited Edition is currently provided with the Sitara Software Development Kit. It uses the Eclipse backend and includes a number of plug-ins that provides support for Linux, the Qt application framework, html editing through a web page editor and tools to provide access to the remote target board.

- ARM GNU Toolchain plug-in
- Qt Integration plug-in
- Web Page Editor plug-in
- Remote System Explorer plug-in

Running the CCSv5 installer

NOTE: The 30-day trial version language in the CCS installer license agreement applies only for the case of using high-speed JTAG

emulators (does not apply to use of the XDS100v2 JTAG emulator). If a debug configuration is used that requires a high-speed

JTAG emulator, you will be prompted to register your software for a fee. All use of CCSv5 (excluding use of high-speed JTAG

emulators) if free and has no 30-day time limit.

1. When the CCSv5 installer runs:

a) On Installation Location dialog:
Do NOT check "Add TI plug-ins into an existing Eclipse install"
Browse to installation folder. Recommended folder is /user/home/tools/CCSv5. That way you can have multiple SDKs
installed that use the same CCSv5 installation. When CCSv5 starts, you can create a unique workspace folder for each SDK.

b) On Product Configuration dialog, select "Platinum Edition"

c) For Sitara ARM-only devices, run-mode Linux application debugging using GDB is usually all that is needed. The Code Sourcery ARM Cross Compiler tool chain is used

for the ARM processor by CCSv5. You can exclude installation of the TI compilers for all DSP platforms, JTAG debugging and JTAG emulator components. This reduces

the size of the installed files from about 4GB to about 600MB and save close to 30 min of install time.

- On Choose ISA dialog:

Leave only ARM selected (uncheck all other ISAs: C28x, C54x DSPs, C55x DSPs, C6000 DSPs, etc.)

- On Select Components dialog:

Do NOT check Code Generation Tools, Target Content or Emulators

d) Otherwise, if you DO require a full CCS installation, (including TI compilers for all DSP platforms, JTAG debugging, and JTAG emulation components)

- On Choose ISA dialog:

Leave all items selected.

- On Select Components dialog:

Leave all items selected.

2. To start CCSv5, click the Code Composer Studio v5 icon on the desktop. You will be prompted to specify a workspace folder.

If you have more than one Sitara SDK installed, we recommend using a unique name for the workspace folder for each SDK.

How to Start Code Composer Studio v5

Refer to How to Start CCSv5 [1] to start Code Composer Studio v5.

How to Import the Matrix GUI Qt Project into CCSv5

Refer to How to Import the Matrix GUI Qt project into CCSv5 [2]

How to Build and Install the Matrix GUI Qt Project

At this point you are done importing the project. See this page for building and installing the Matrix GUI Qt project: → How to Build the Matrix GUI

How to Import C/C++ Demo Projects for Matrix GUI into CCSv5

The following article will guide you through the steps to import all C/C++ projects (other than Qt applications) under the Matrix GUI into CCSv5: → How to Import Matrix Projects into CCSv5.

How to Build C/C++ Demo Projects for Matrix GUI

Click on the project name under the Projects tab to select the project.

Select the build configuration by clicking either of these two menu items:

Project -> Build Configurations -> Set Active -> Debug

or

Project -> Build Configurations -> Set Active -> Release

Click the Project -> Build Program menu item to build the program.

How to Install C/C++ Demo Projects for Matrix GUI into the Target File System

The table below lists the location on the Ubuntu Linux host where the executable files reside for the debug and release builds and also the location in the target file system where either executable file must be copied to. (X.Y refers to the current version of the example application, which is subject to change.)

Location of Debug Executable	Location of Release Executable	**Location in Target File System**
./matrix-gui-e-X.Y/matrix_gui	./matrix-gui-e-X.Y/matrix_gui	/usr/bin/matrix_guiE
./matrix-tui-X.Y/Debug/matrix_tui.elf	./matrix-tui-X.Y/Release/matrix_tui.elf	/usr/bin/matrix_tui
./am-benchmarks-X.Y/dhrystone/Debug/dhrystone	./am-benchmarks-X.Y/dhrystone/Release/dhrystone	/usr/bin/dhrystone
./am-benchmarks-X.Y/whetstone/Debug/whetstone	./am-benchmarks-X.Y/whetstone/Release/whetstone	/usr/bin/whetstone
./am-benchmarks-X.Y/linpack/Debug/linpack	./am-benchmarks-.Y/linpack/Release/linpack	/usr/bin/linpack
./am-sysinfo-X.Y/mem_util/Debug/mem_util.elf	./am-sysinfo-X.Y/mem_util/Release/mem_util.elf	/usr/bin/mem_util

There are several methods for copying the executable files to the target file system:

1) Use the top-level Makefile in the SDK install directory. The advantage of this is that all associated data files also get copied to the target file system.

The following make install commands are supported. First change directory to your SDK install directory, for example:

cd /home/user/ti-sdk-am181x-evm-4.0.1.0

Enter the make commands shown below at the Ubuntu Linux host command line.

```
+----------------------------------+-----------------------------------------------------+
| make matrix_gui_install          | Builds matrix_gui release version,
 copies          |
|                                  | executable and associated data
files to target file|
|                                  | system. (Assumes release version
was built last.)  |
+----------------------------------+-----------------------------------------------------+
| make matrix_gui_install_debug    | Builds matrix_gui debug version,
copies executable |
|                                  | and associated data files to
target file system.   |
|                                  | (Assumes debug version was built
last.)            |
+----------------------------------+-----------------------------------------------------+
| make matrix_tui_install          | Copies matrix_tui release version
and associated   |
|                                  | data files to target file system
                  |
+----------------------------------+-----------------------------------------------------+
| make matrix_tui_install_debug    | Copies matrix_tui debug version
and associated     |
|                                  | data files to target file system
```

```
                    |
+---------------------------------+-------------------------------------------------+
| make am_benchmarks_install       | Copies am_benchmarks release
version and associated|
|                                  | data files to target file system
                    |
+---------------------------------+-------------------------------------------------+
| make am_benchmarks_install_debug | Copies am_benchmarks debug version
 and associated  |
|                                  | data files to target file system
                    |
+---------------------------------+-------------------------------------------------+
| make am_sysinfo_install          | Copies am_sysinfo release version
and associated   |
|                                  | data files to target file system
                    |
+---------------------------------+-------------------------------------------------+
| make am_sysinfo_install_debug    | Copies am_sysinfo debug version
and associated     |
|                                  | data files to target file system
                    |
+---------------------------------+-------------------------------------------------+
```

2) The Remote System Explorer can be used to copy the required executable file from the Ubuntu Linux host to the target file system. Once copied to the target file system, you may also need to rename the file and change the file permissions so that the file is executable.

How to Setup and Use the Remote System Explorer Eclipse Plug-in

CCSv5 as installed with this SDK includes the Remote System Explorer (RSE) plug-in. RSE provides drag-and-drop access to the target file system as well as remote shell and remote terminal views within CCS. Refer to How to Setup and Use Remote System Explorer [3] to establish a connection to your target EVM and start using RSE.

Using GDB Server in CCSv5 for Linux Debugging

In order to debug Linux code using Code Composer Studio v5, you first need to configure the gdbserver on both the host and target (EVM) side.

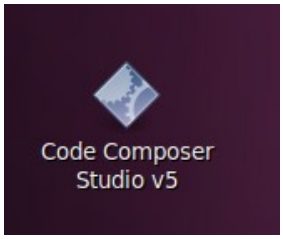
Please refer to Running GDB Server on CCSv5 [4] for more information.

References

[1] http://processors.wiki.ti.com/index.php/How_to_start_Code_Composer_Studio_v5
[2] http://processors.wiki.ti.com/index.php/How_to_Import_Matrix_GUI_into_CCSv5
[3] http://processors.wiki.ti.com/index.php/How_to_setup_Remote_System_Explorer_plug-in
[4] http://processors.wiki.ti.com/index.php/How_to_Run_GDB_on_CCSv5

How to start Code Composer Studio v5

Double-click the Code Composer Studio v5 icon on the Ubuntu desktop.



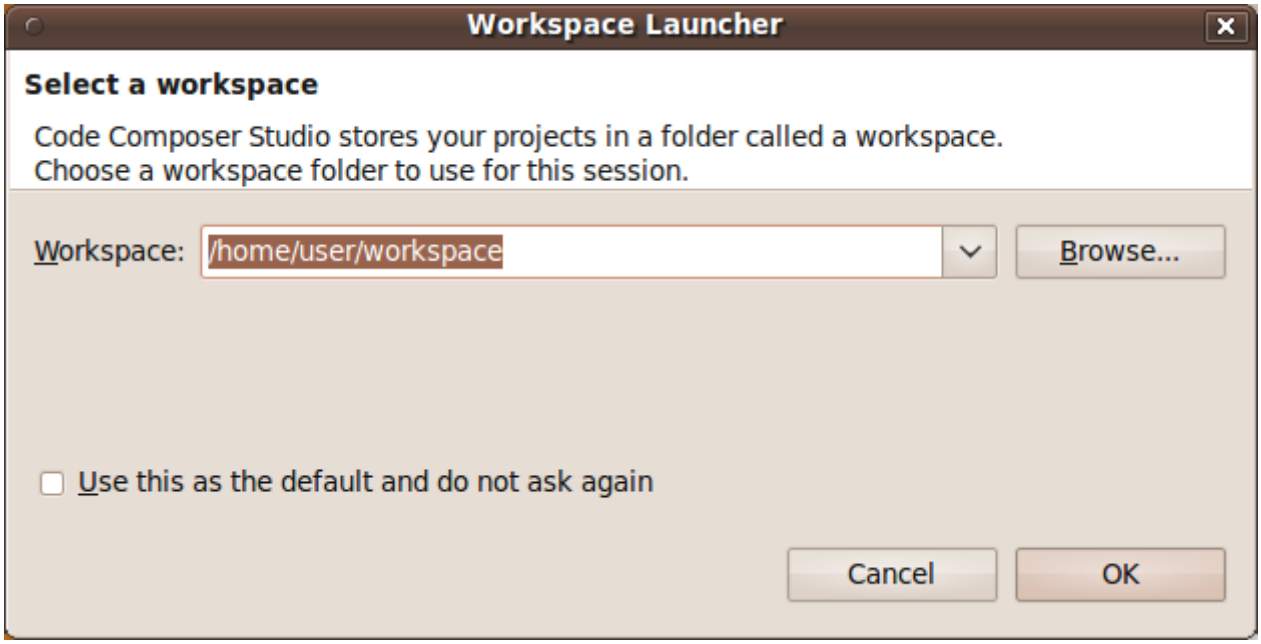
(Earlier Sitara SDKs used CCSv5 Preview Edition which did not provide an icon on the desktop to start CCS.

In that case, CCS can be started by running /home/user/<SDK_Install_Folder>/CCSv5_Preview_Edition/eclipse).

Once you launch it, you should see the CCSv5 splash screen.

Soon after the Splash screen, you will be asked to select a workspace. You should simply replace your linux user name with the 'user' shown

in the image below: So if your username is johnQ, then you would select /home/johnQ/workspace.



Once you selected the location of your workspace, and hit OK, the main CCSv5 window should be displayed.

The first time CCSv5 runs it displays the Welcome screen. Click the X on the Welcome tab to get to the CCS workbench.

How to Import Matrix GUI into CCSv5

Getting Started Importing the Matrix Gui

The following instructions will help you to import the Matrix GUI project into CCSv5.

- This article assumes you have already installed the AM SDK on your Linux Ubuntu Host.

If not see this link to install the AMxx SDK: TBD

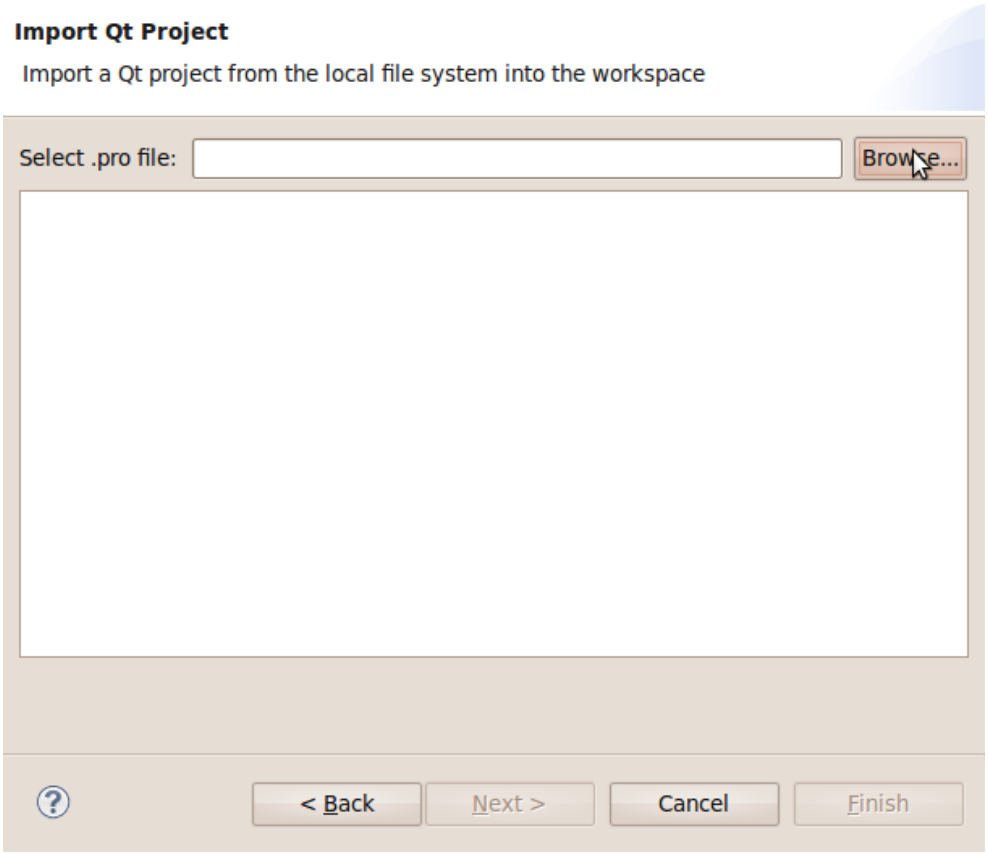
Starting CCSv5

Refer to How to Start CCSv5 [1] to start Code Composer Studio v5.

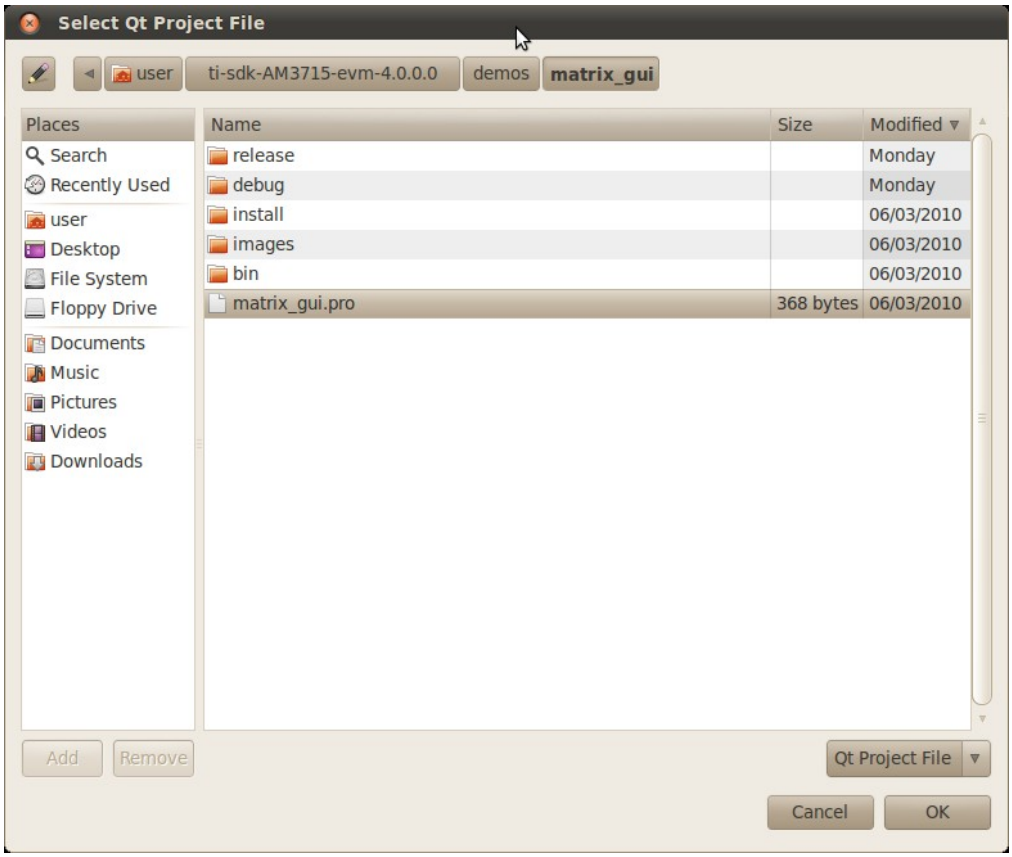
Importing the Project into CCSv5

From the main CCSv5 window, click on the File -> Import... menu item to import a project.
Select a Qt Project as the project type to import.

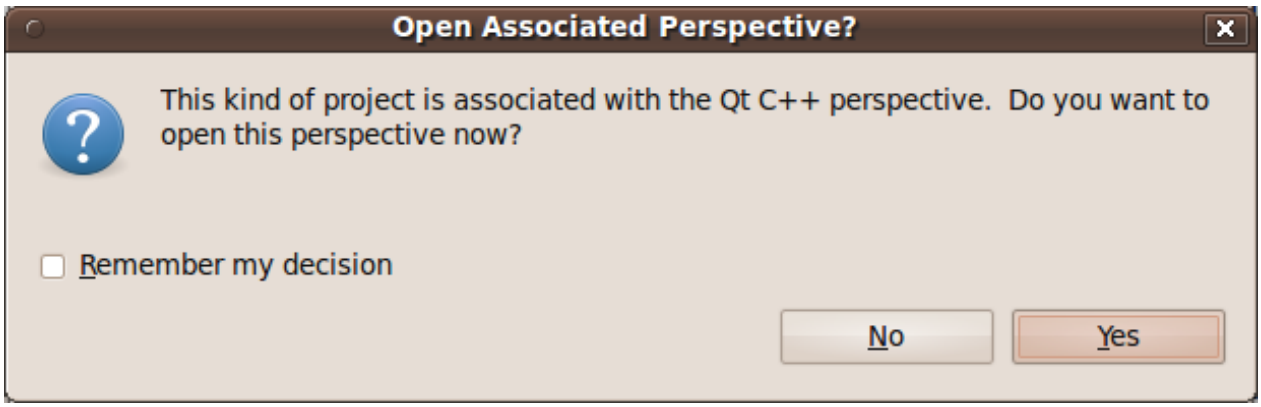
Once you select a QT project and click on next, you will be asked to select a .pro file which is your QT project. You can browse to find the matrix_gui.pro file.



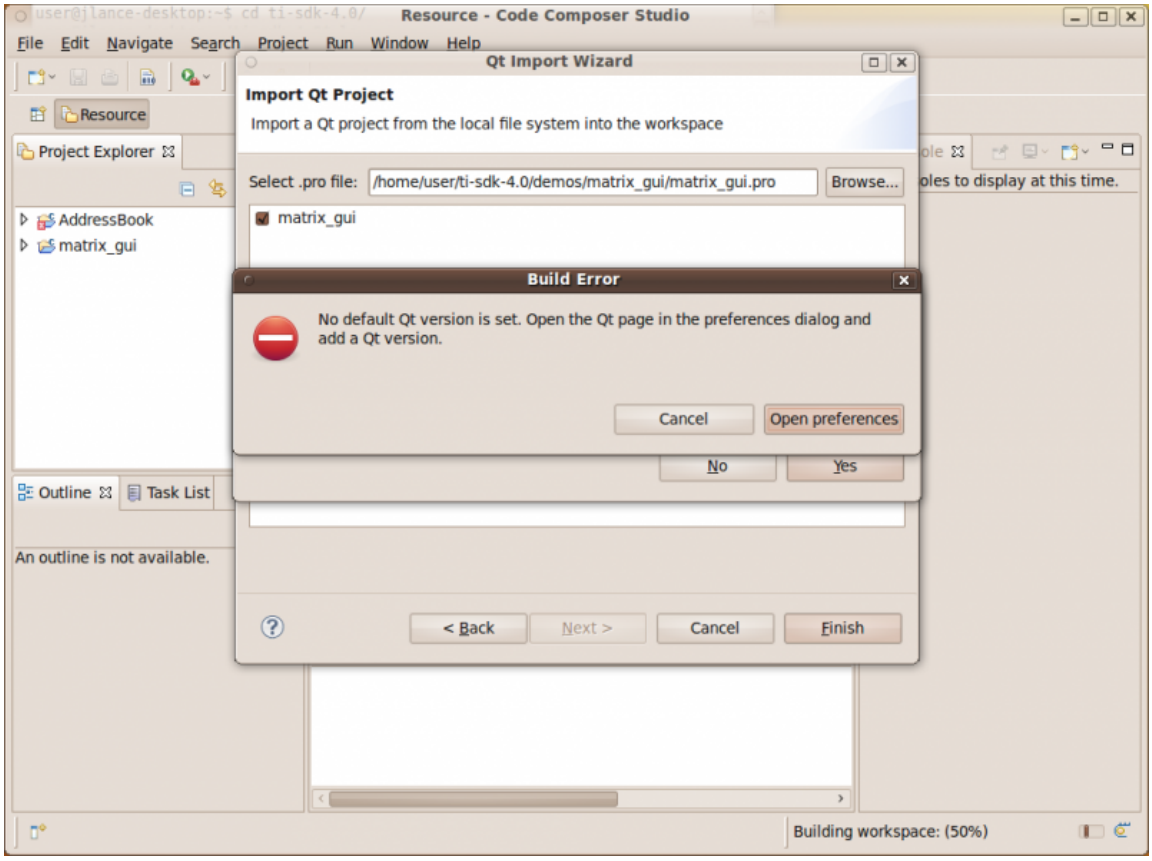
Now you can select the matrix_gui.pro which you can find in your SDK at /home/user/ti-sdk-AM3715-evm-4.0.0.0/demos/matrix_gui



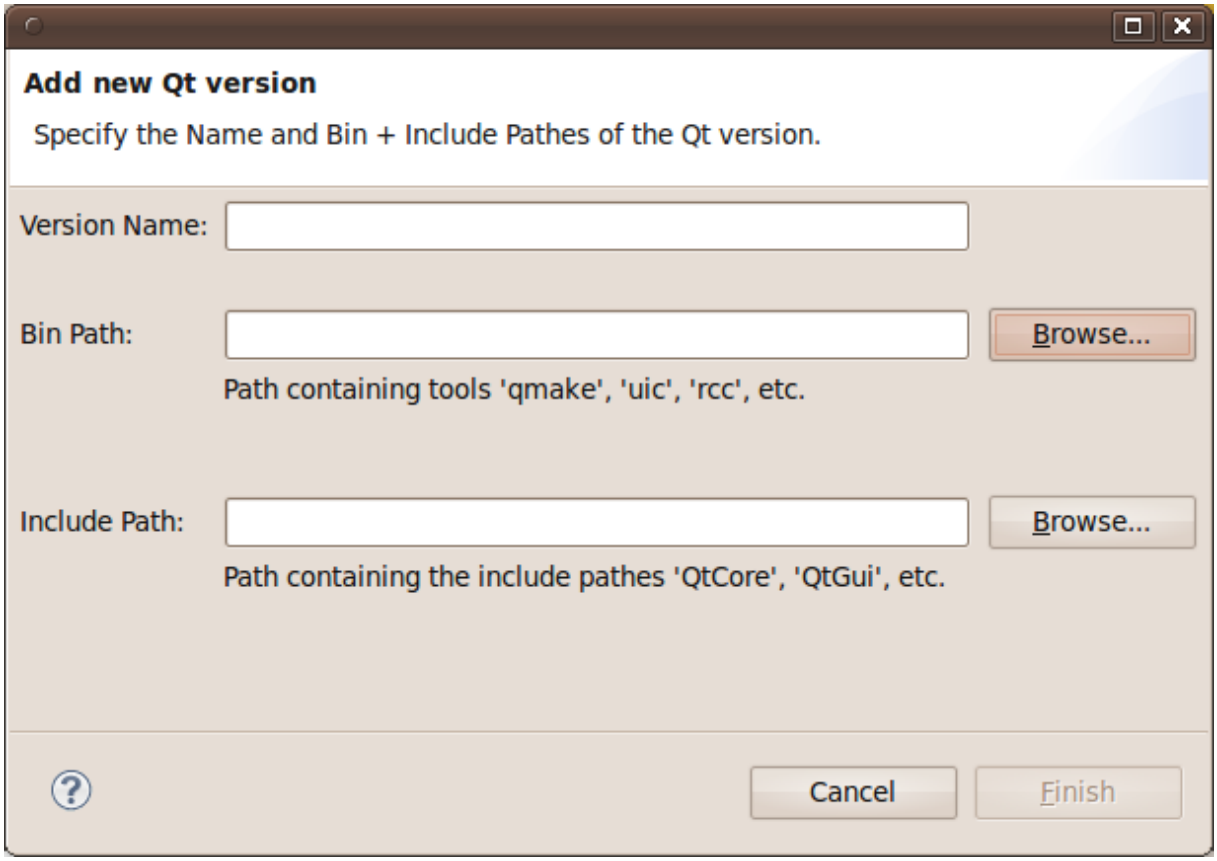
Once you hit OK, the matrix_gui has been imported. When the next prompt asks if you would like to switch to the QT C/C++ perspective, click on yes.



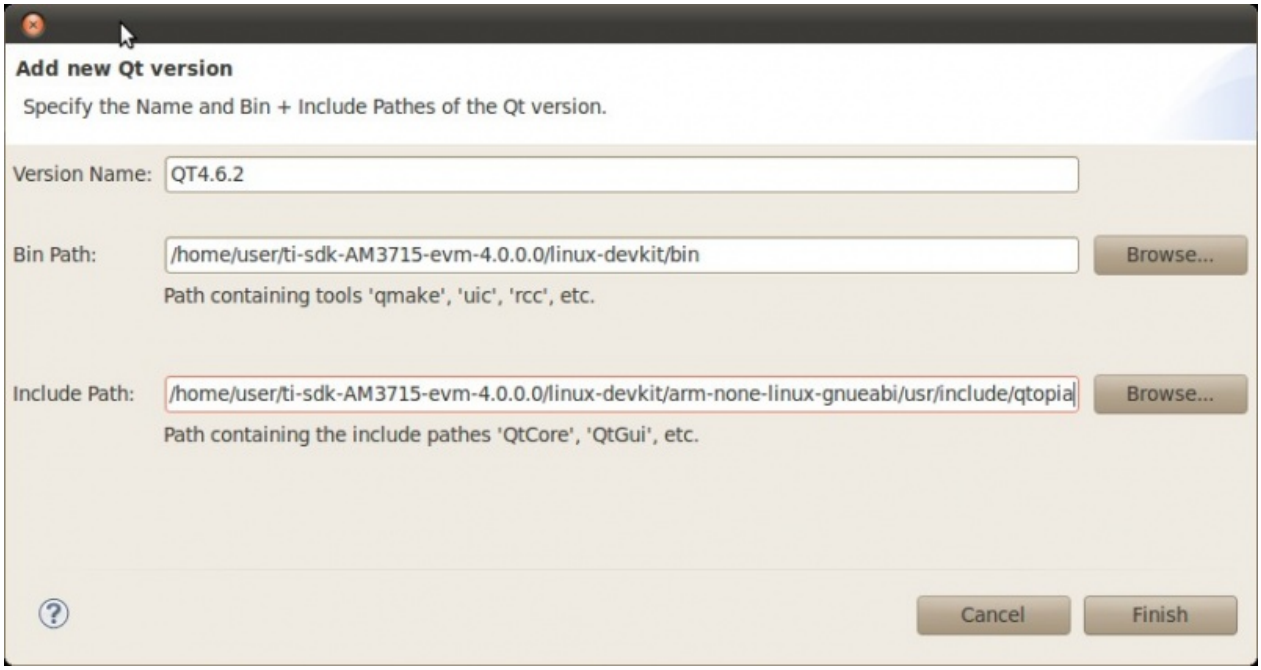
Next the project will try to automatically build. If you have not set up a default QT version to work with, the dialog box will ask you to do this.



All the necesssary files come along with your SDK, so you can fill in the paths. If you do not select a valid path, you will get an error message and you will not be able to continue.



Below, you can see the necessary paths. Substitute your actual SDK install folder name in place of "ti-sdk-AM3715-evm-4.0.0.0".



At this point you are done importing the project and it should automatically build. See this page for Building the Matrix GUI: → How to Build the Matrix GUI

How to Build the Matrix GUI

Getting Started Building the Matrix Gui

The following instructions will help you to Build the Matrix GUI project using CCSv5.

- This article assumes you have already installed the AM SDK on your Linux Ubuntu Host.

If not see this link to install the AMxx SDK: TBD

Starting CCSv5

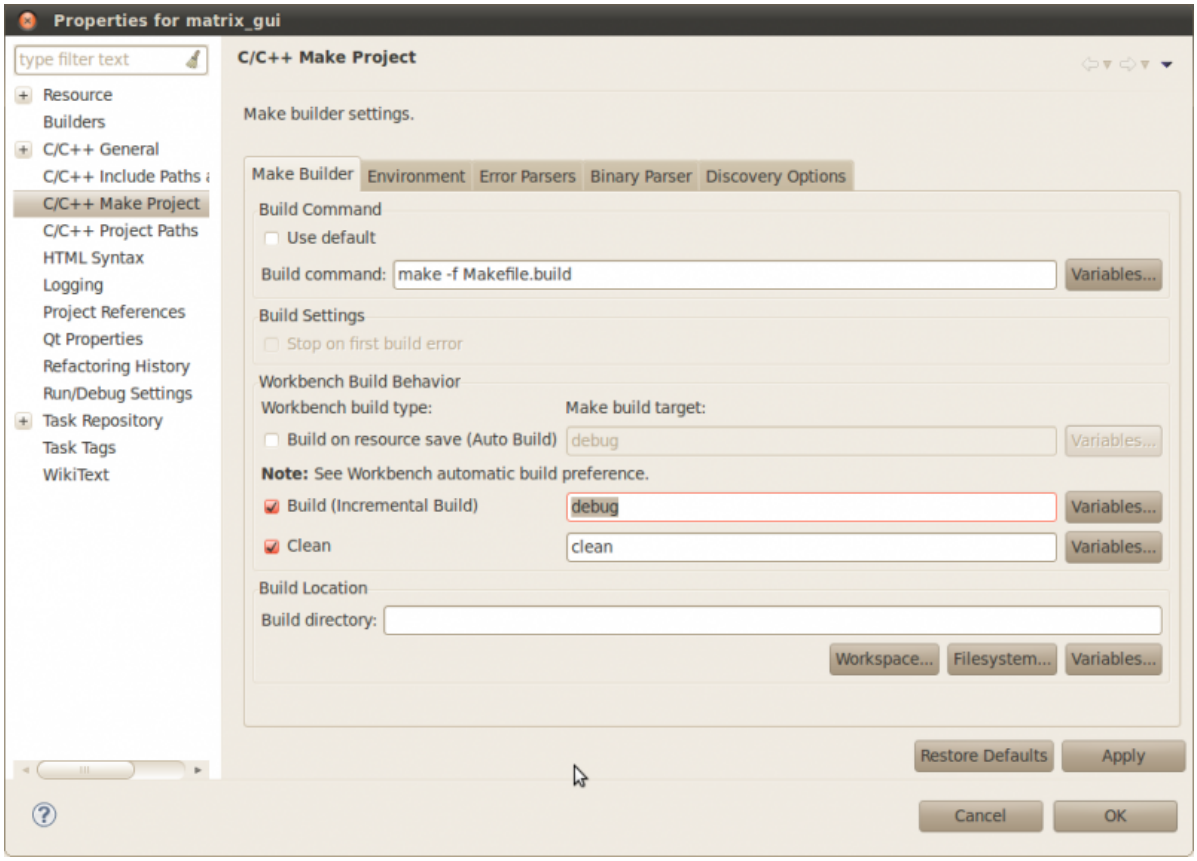
Refer to How to Start CCSv5 [1] to start Code Composer Studio v5.

Select the Build Configuration

Click on the matrix_gui project under the Project tab to select the project. R-Click on the matrix_gui project and select Properties from the context menu.

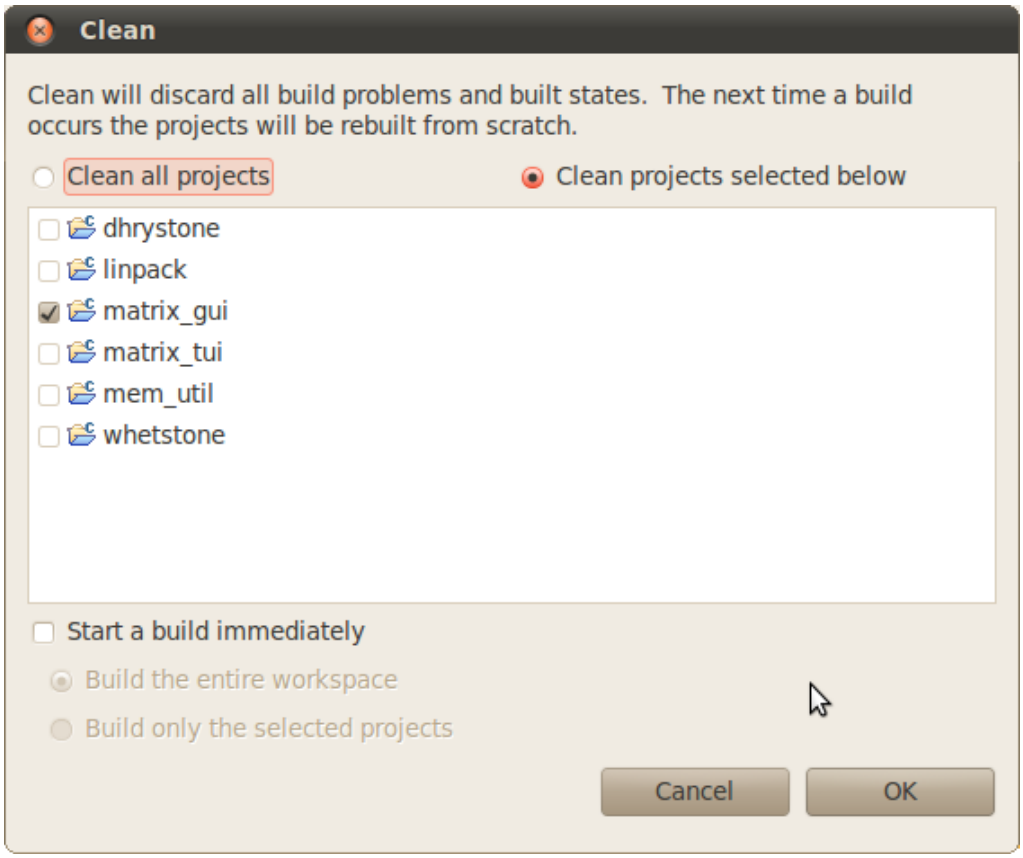
Select C/C++ Make Project. Initially the make Builder tab will be active but the displayed page is empty. Click on another tab then click on the Make Builder

tab to see the dialog box as shown below. You can set the default build configuration by typing "debug" or "release". Click OK to close thedialog box.



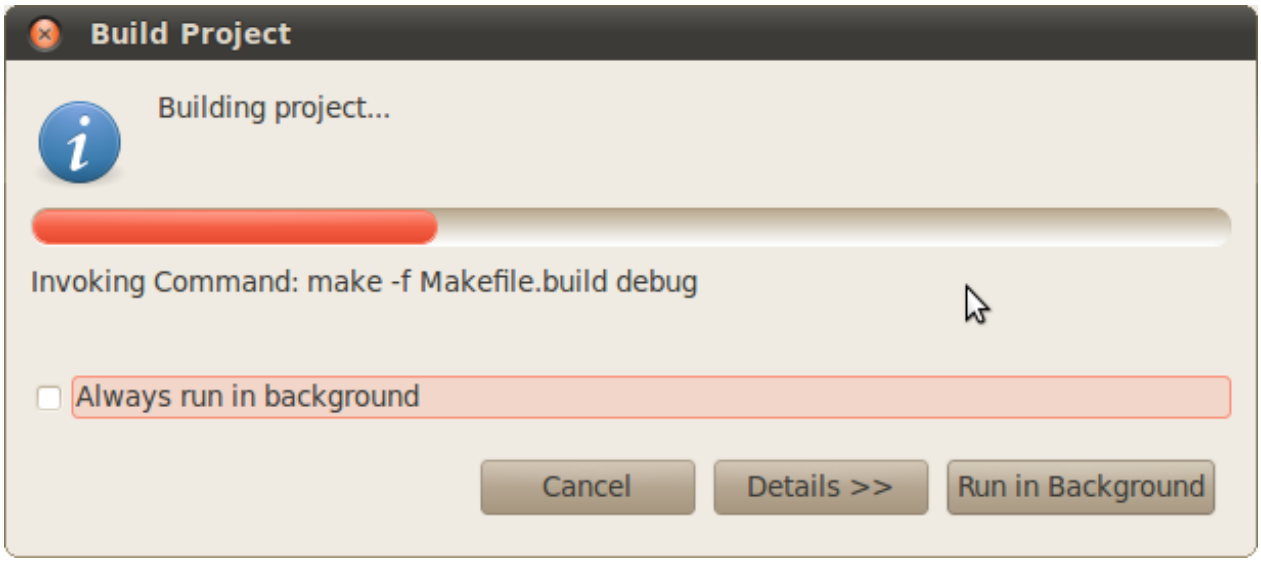
Clean the Project

Click the matrix_gui project if not already selected. Click the Project -> Clean menu item. Select "Clean projects selected below". Leave matrix_gui selected. Uncheck "Start a build immediately". Click OK. This will clean both the debug and release builds.

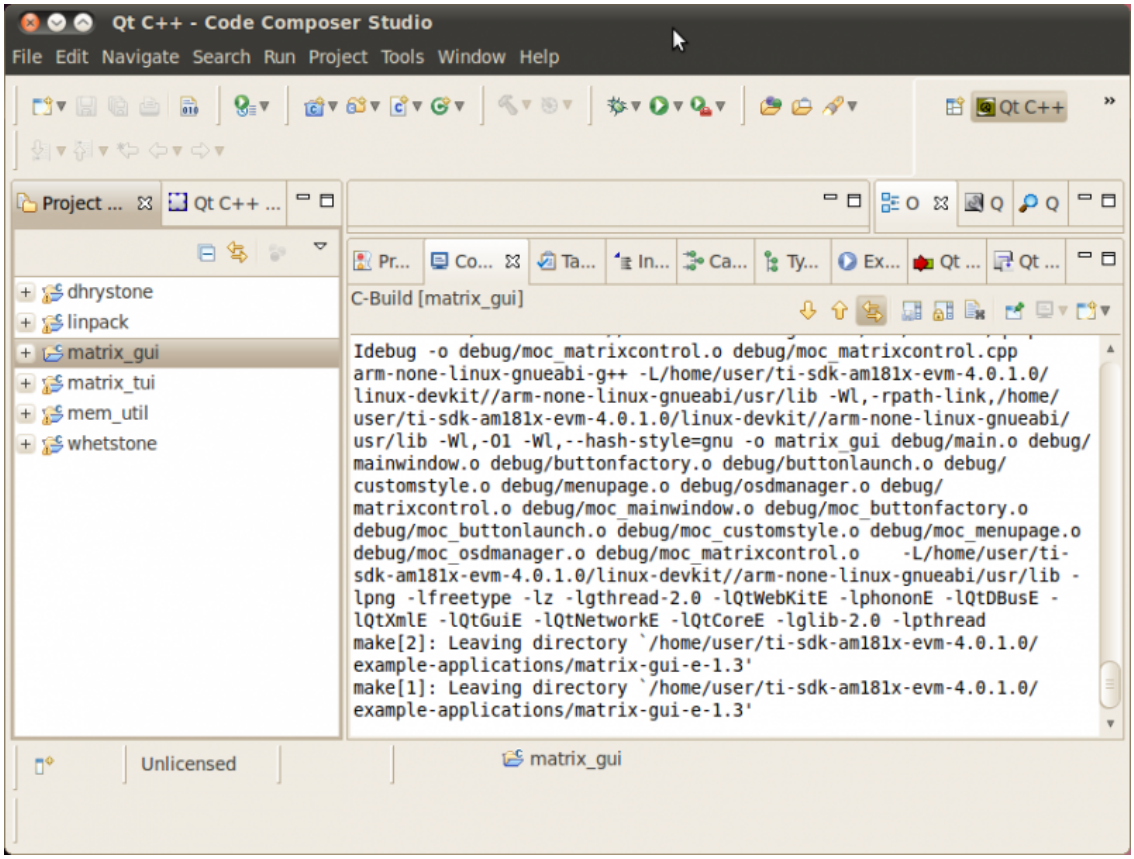


Build the Project

Click on the matrix_gui project if not already selected. Click the Project -> Build Project menu item. A progress indicator dialog will be displayed during the build.



Below is example build console output for the matrix_gui debug build configuration.



Install the Project

The next step is to copy the built executable to the target file system so that it can be run on the EVM. Note that the matrix_gui Qt application

uses the same executable file for the debug and release build configurations. The commands below to install the debug version or install the

release version assume that the last build which updated the executable file was performed using the required build configuration.

Using the Ubuntu Linux Host Command Line

The SDK top-level Makefile provides separate install targets for the debug and release build configurations.

Both of these copy the executable and all associated data files to the target file system. First change

directory to the SDK install directory. (Substitute your SDK install directory name.)

```
cd ~/ti-sdk-am181x-evm-4.0.1.0
```

To install the matrix_gui release version:

```
make matrix_gui_install
```

To install the matrix_gui debug version:

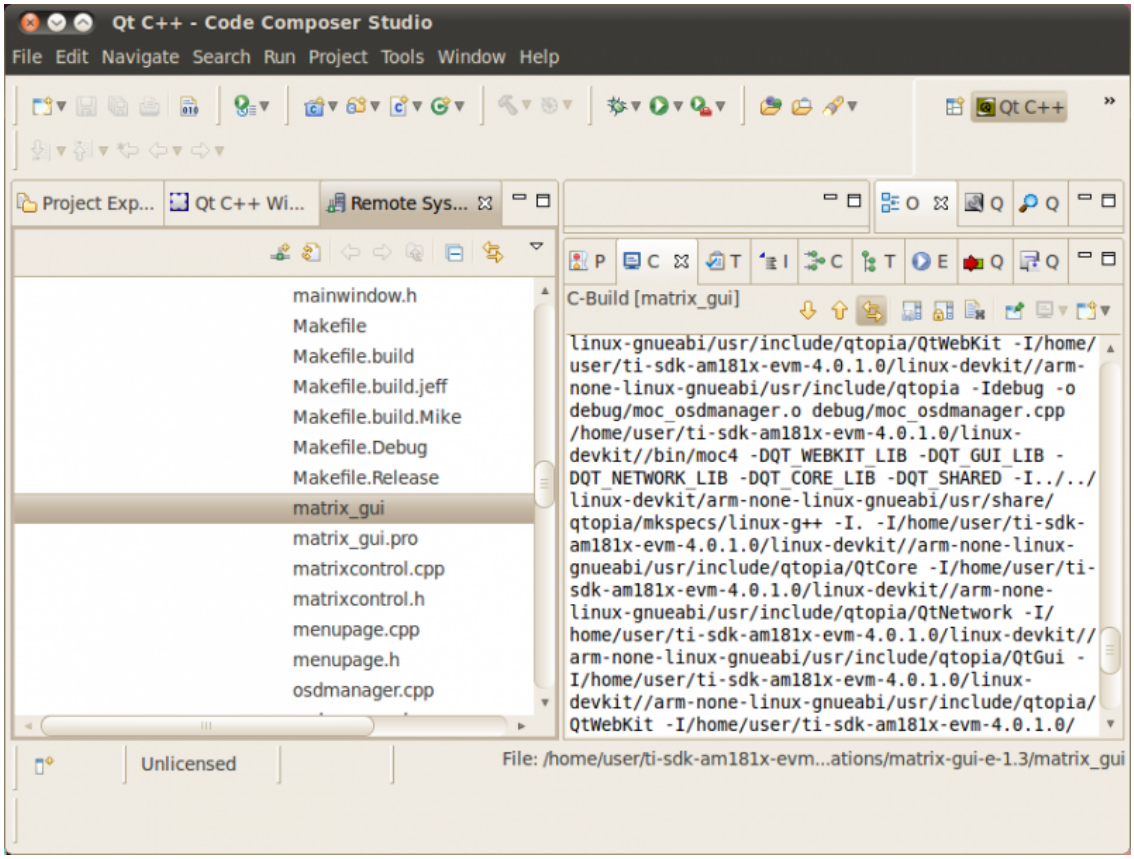
```
make matrix_gui_install_debug
```

Using Remote System Explorer

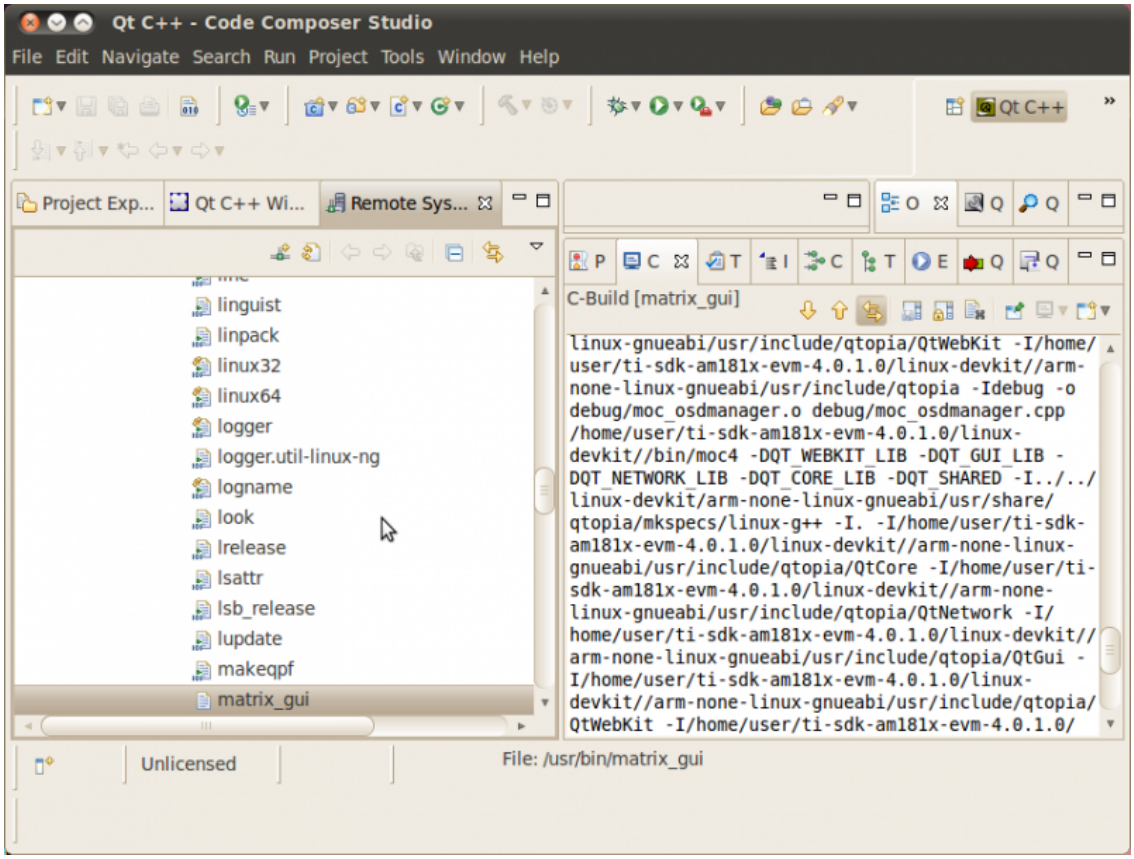
The Remote System Explorer plug-in can be used to transfers files to the target file system.

(See How_to_Setup_and_Use_the_Remote_System_Explorer_Eclipse_Plug-in [1] to setup the RSE plug-in.)

Under the Remote Systems tab, expand the Local Files node and browse to the location of the matrix_gui executable. R-Click on the matrix_gui file and select Copy from the context menu.

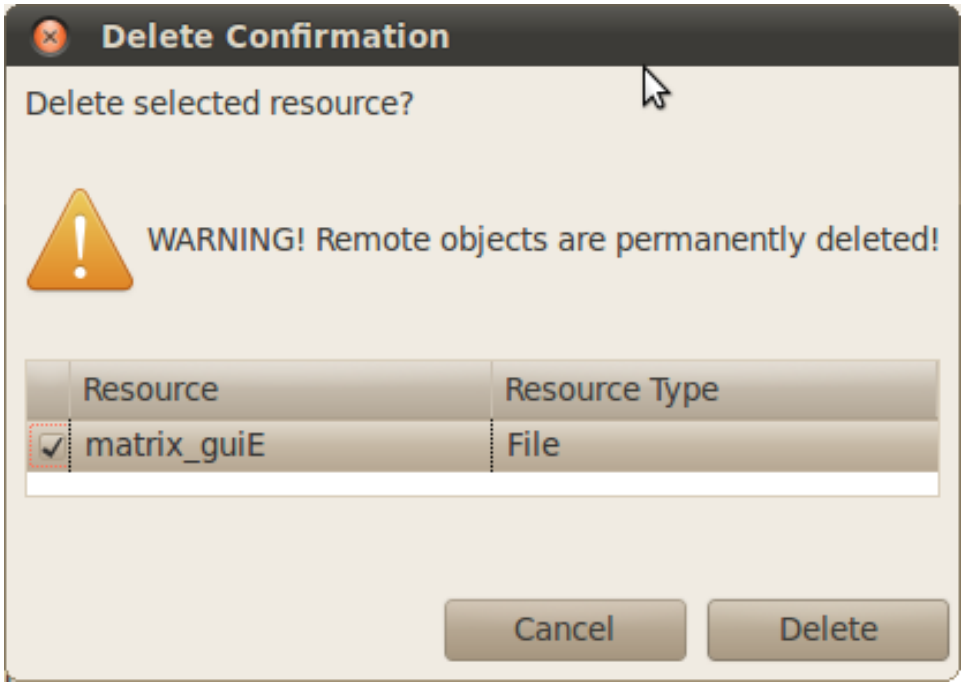


Expand the node with your target board connection. Browse to the /usr/bin folder. R-Click on the /usr/bin folder and select Paste from the context
menu.



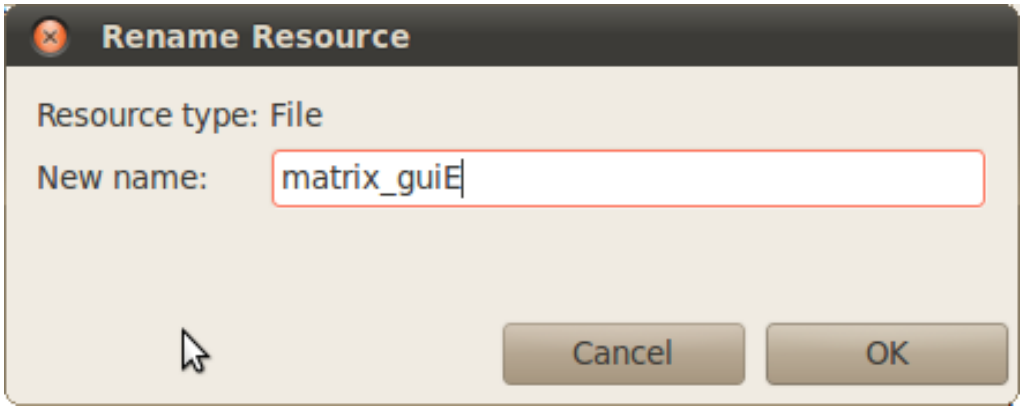
Since the executable on the target board is named matrix_guiE, the new executable needs to be renamed. Scroll down to the existing file

named matrix_guiE. R-Click on matrix_guiE and select Delete from the context menu. Click OK on the Delete Confirmation dialog box.

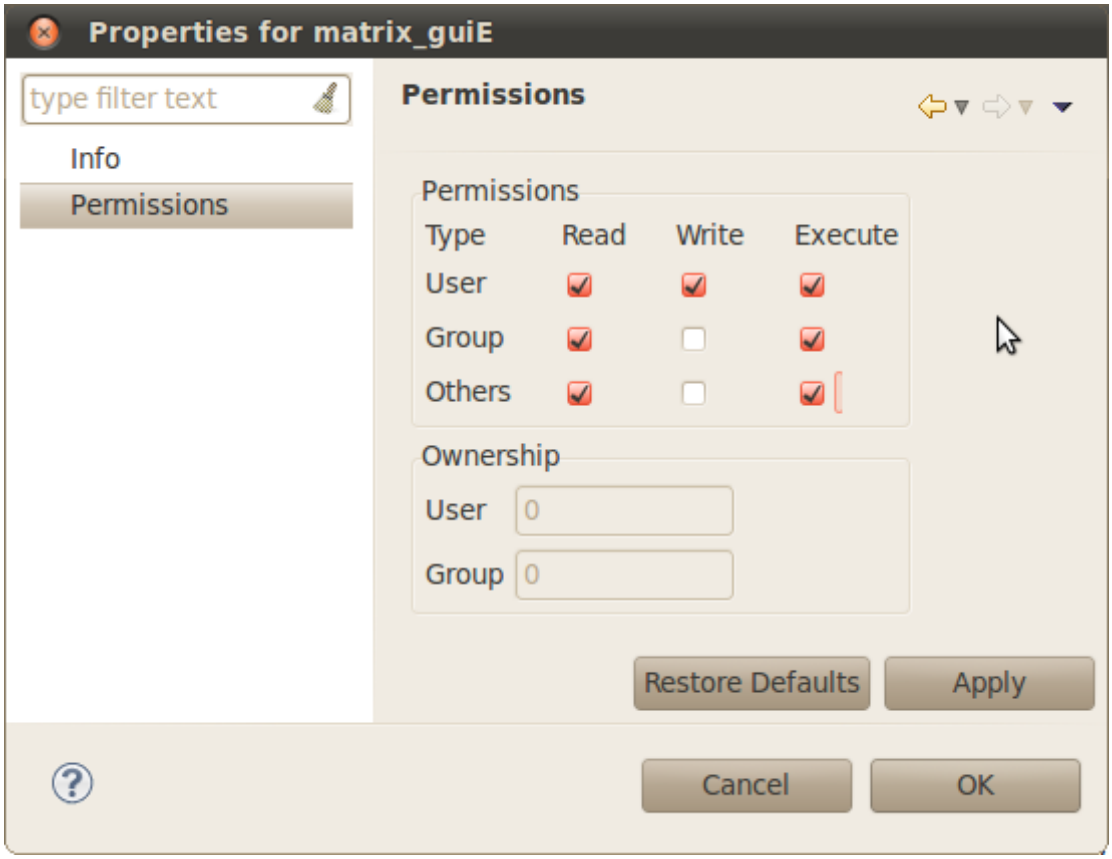


R-Click on the matrix_gui file and select Rename from the context menu. In the Rename Resource

dialog box, change the name to matrix_guiE and click OK.



R-Click on the matrix_guiE file and select Properties from the context menu. Click on Permissions. Check Execute for User, Group and Others and click OK.



Code_Composer_Studio_v5

References

[1] http://processors.wiki.ti.com/index.php/
Code_Composer_Studio_v5_Preview_Edition_Users_Guide#How_to_Setup_and_Use_the_Remote_System_Explorer_Eclipse_Plug-in

How to Import Matrix Projects into CCSv5

Getting Started Importing Projects into CCSv5

The following instructions will help you to import the Matrix GUI example application C/C++ projects into CCSv5.

- This article assumes you have already installed the Sitara SDK for your hardware platform on your Ubuntu Linux Host.

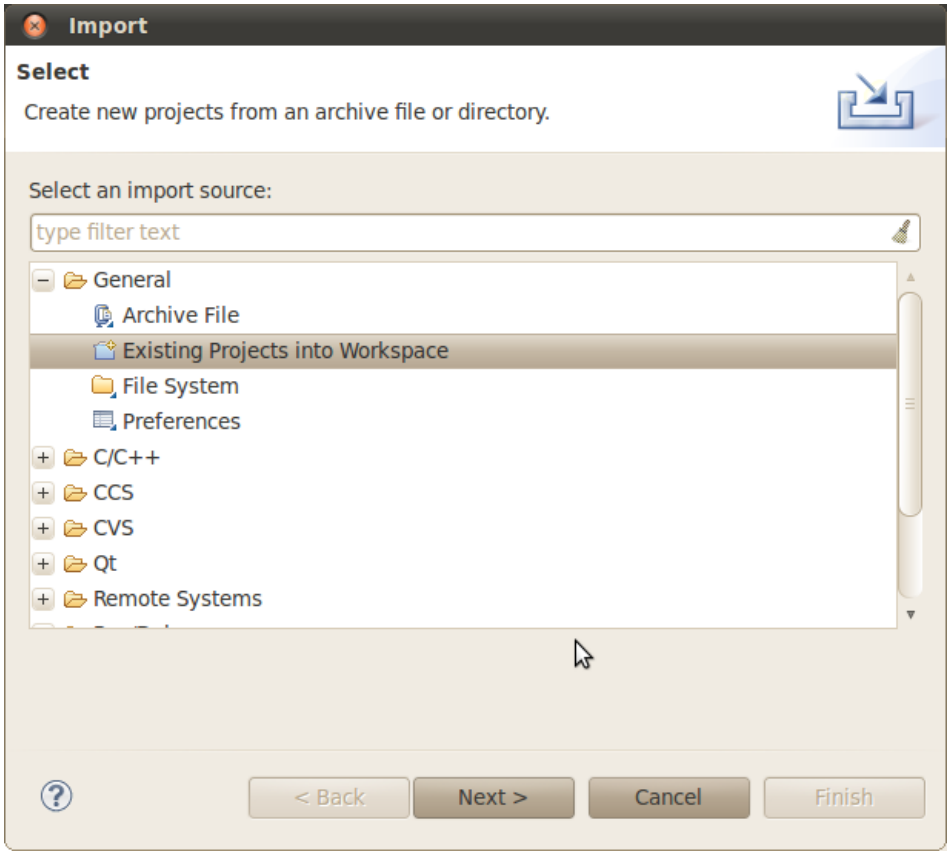
If not, see this link to install the Sitara SDK: Installing_the_Sitara_SDK [1]

Starting CCSv5

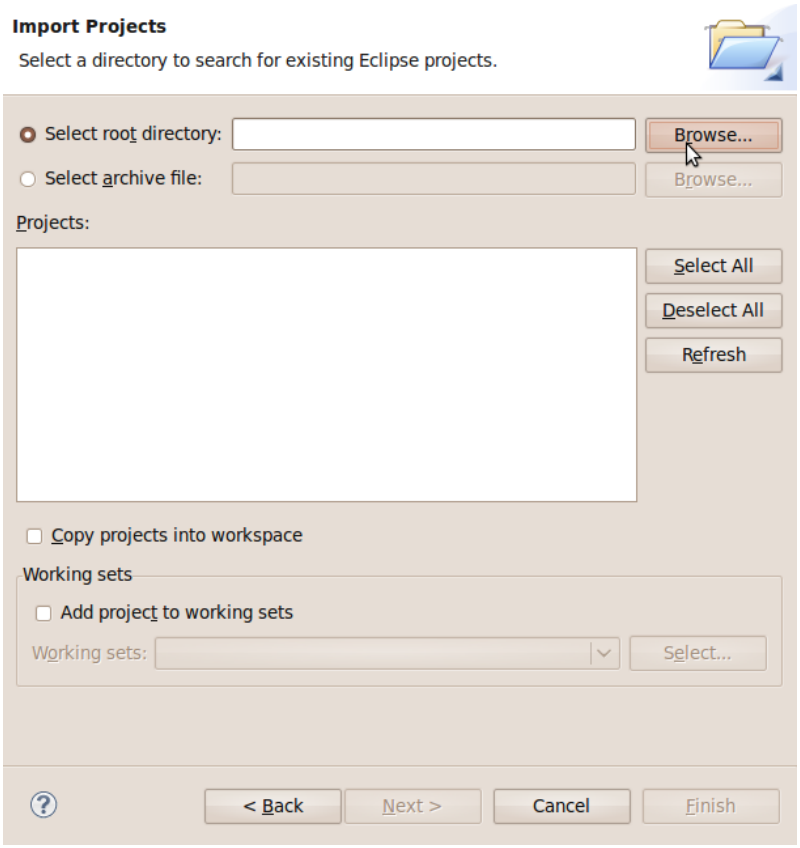
Refer to How to Start CCSv5 [1] to start Code Composer Studio v5.

Importing the C/C++ Projects into CCSv5

From the main CCSv5 window, click on the File -> Import... menu item to import projects. Now you should see the window below. Expand the General node, select "Existing Projects into Workspace" and hit Next.



Now select "Browse" to find the location of the Matrix Projects.

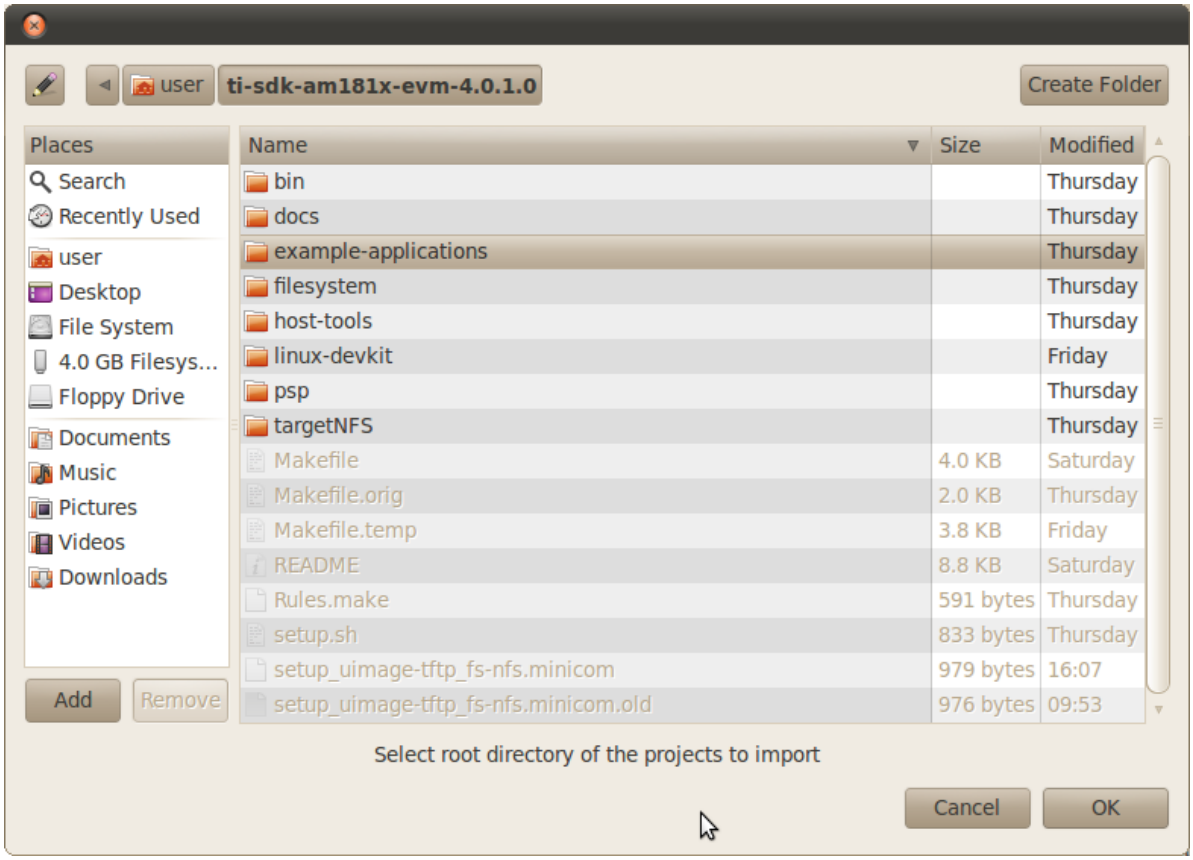


Browse to where the Matrix example applications are located in the SDK.

```
/home/user/ti-sdk-am181x-evm-4.0.1.0/example-applications
```

You may need to replace "user" with your Unix User ID and substitute your actual SDK install folder name for "ti-sdk-am181x-evm-4.0.1.0".

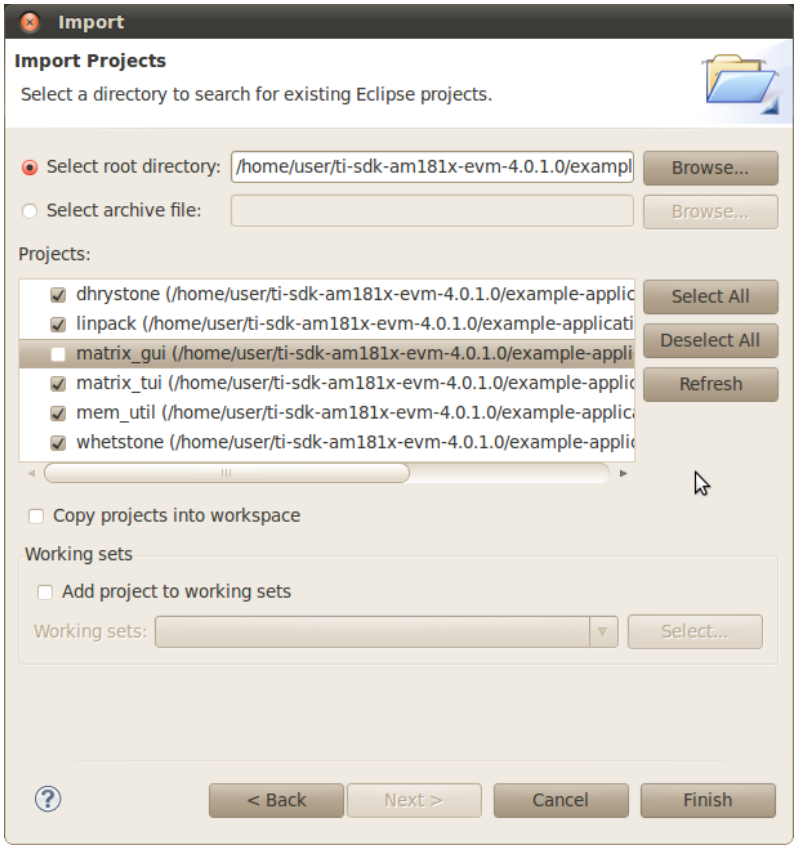
Then hit "OK" to continue.



The Import dialog box will open, displaying all CCS projects that were found. Uncheck the matrix_gui project, since it is a QT-based project.

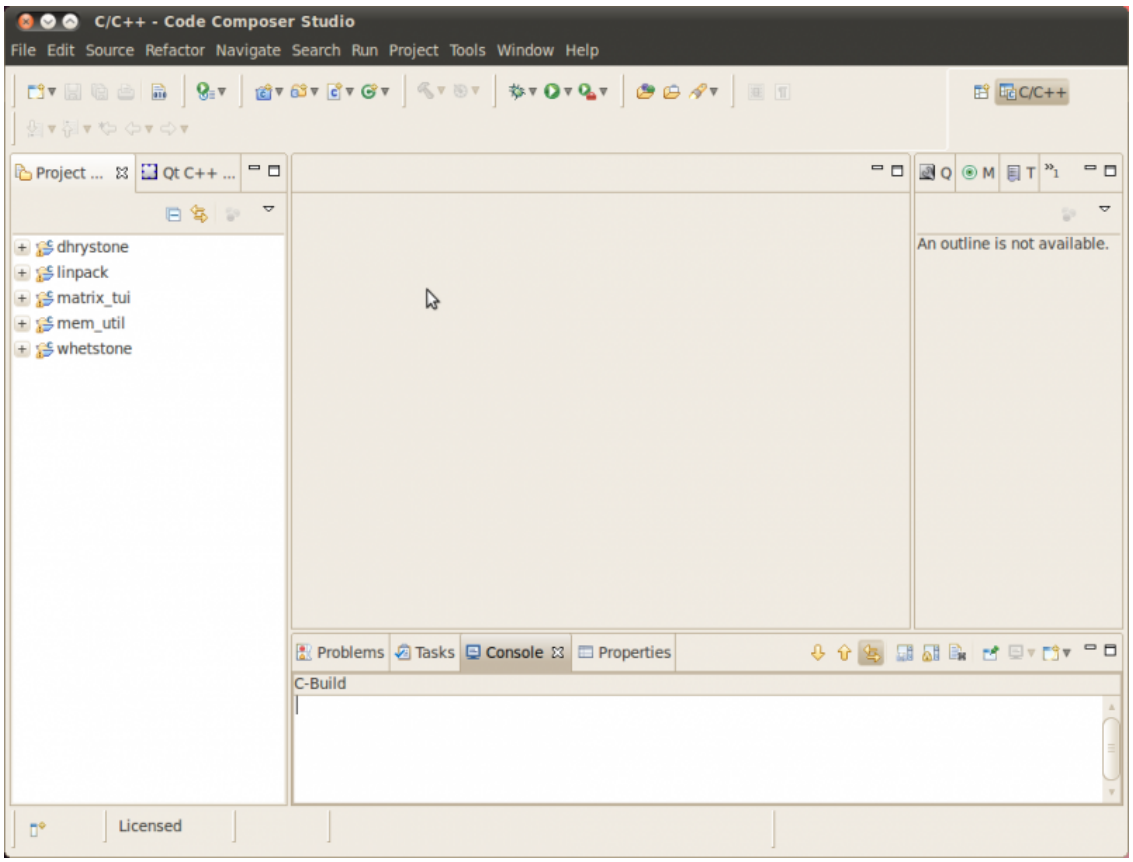
QT-based projects have a different import procedure. To import the matrix_gui project, see the following wiki page: How_to_Import_Matrix_GUI_into_CCSv5 [2].

After de-selecting matrix_gui, hit "Finish" to import the existing projects.



At this point you are done importing the C/C++ projects. See screen shot below. Notice that all the imported projects are now listed at the the top left

under the Project tab. Later, you can make the Project tab visible by activating the C/C++ Eclipse perspective (in upper right area of the IDE).



References

[1] http://processors.wiki.ti.com/index.php/Installing_the_AM37x_SDK

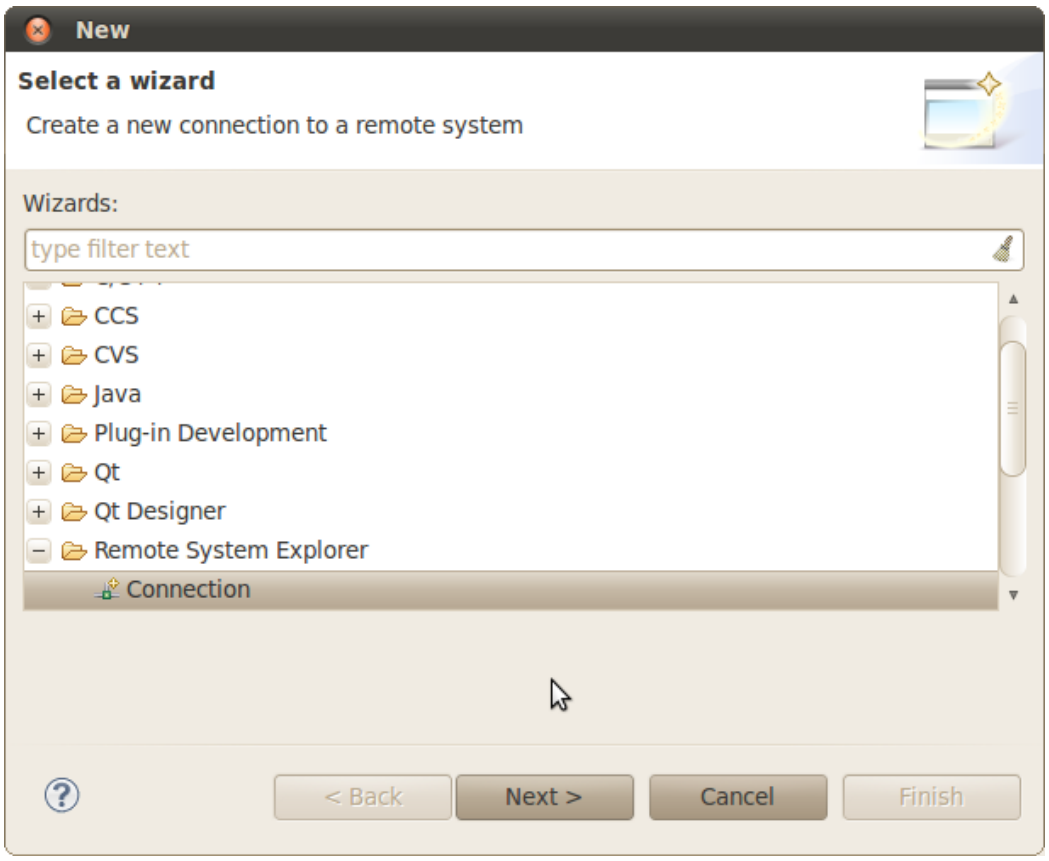
How to setup Remote System Explorer plug-in

How to Setup Remote System Explorer Eclipse Plug-in

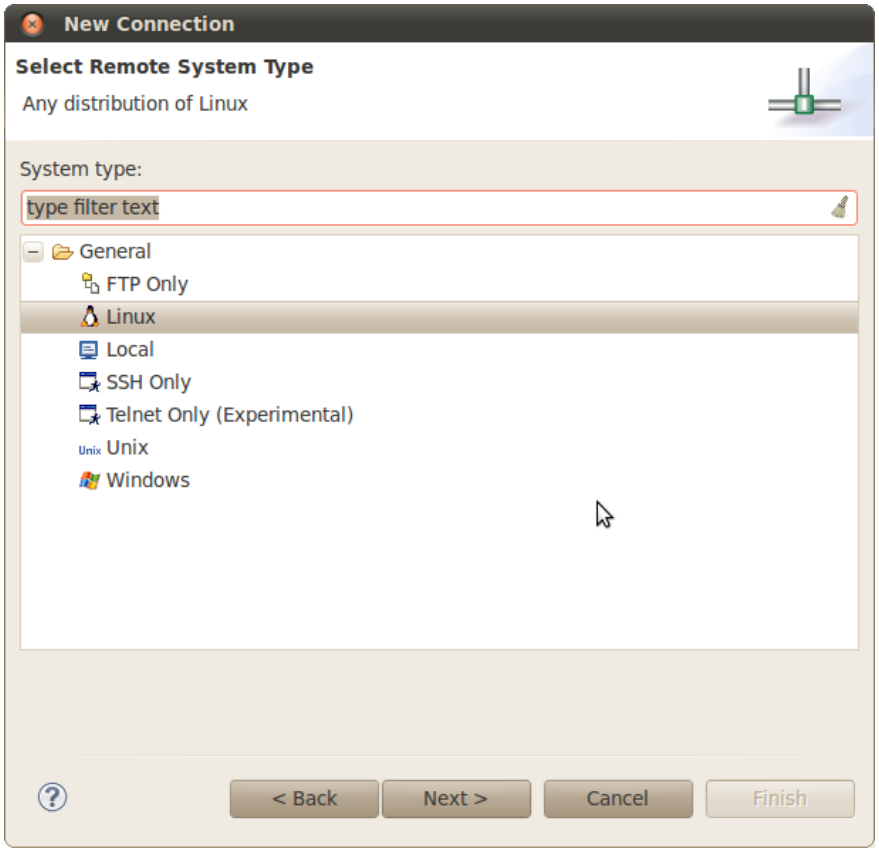
Remote System Explorer (RSE) is an Eclipse plug-in that provides drag-and-drop access to the remote file system, and also provides views for a remote shell, remote terminal and a remote process monitor which displays information about all the processes running on the target. The first time this plug-in is used, a connection to the target EVM needs to be established and configured. Follow the procedures "Running the New Connection Wizard", "Opening the Remote System Explorer View" and "Configuring the Target EVM Connection".

Running the New Connection Wizard

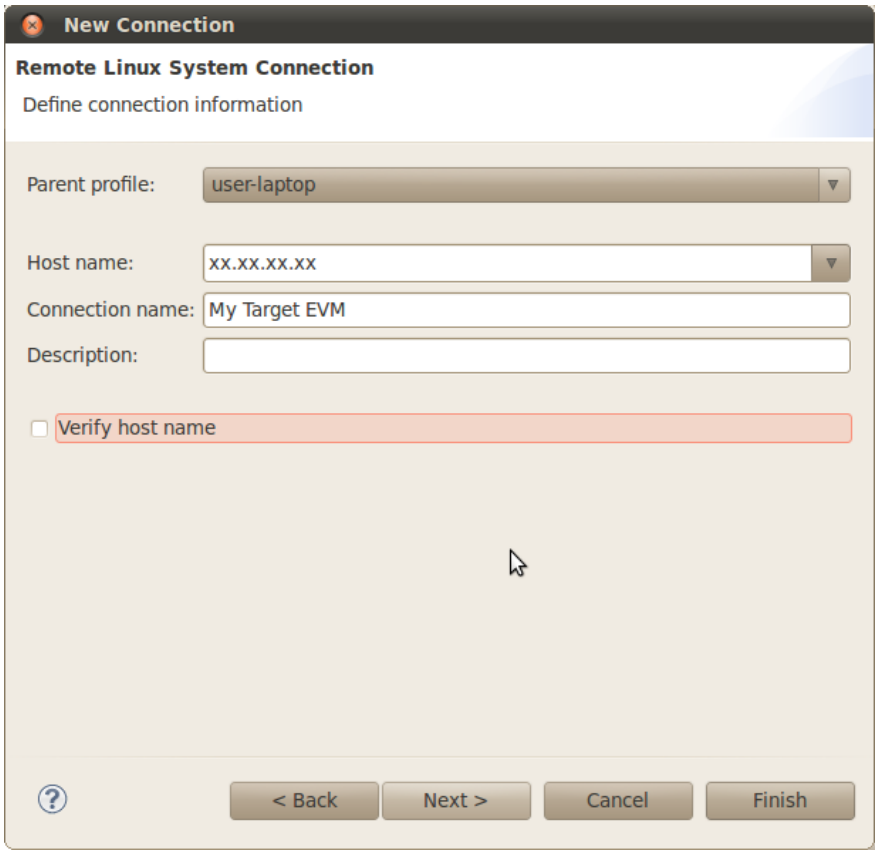
To establish a new connection with the target EVM you must run the New Connection Wizard. To open this click on the File -> New -> Other menu item. In the dialog below, expand the Remote System Explorer folder, click on Connection and Next.



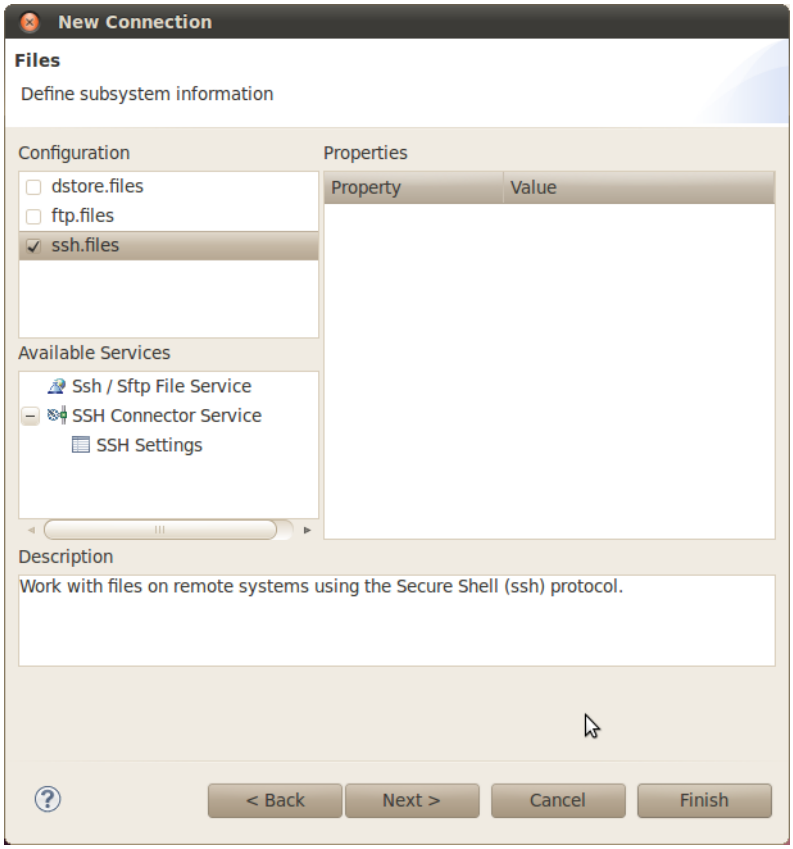
Select the Linux system type and click Next.



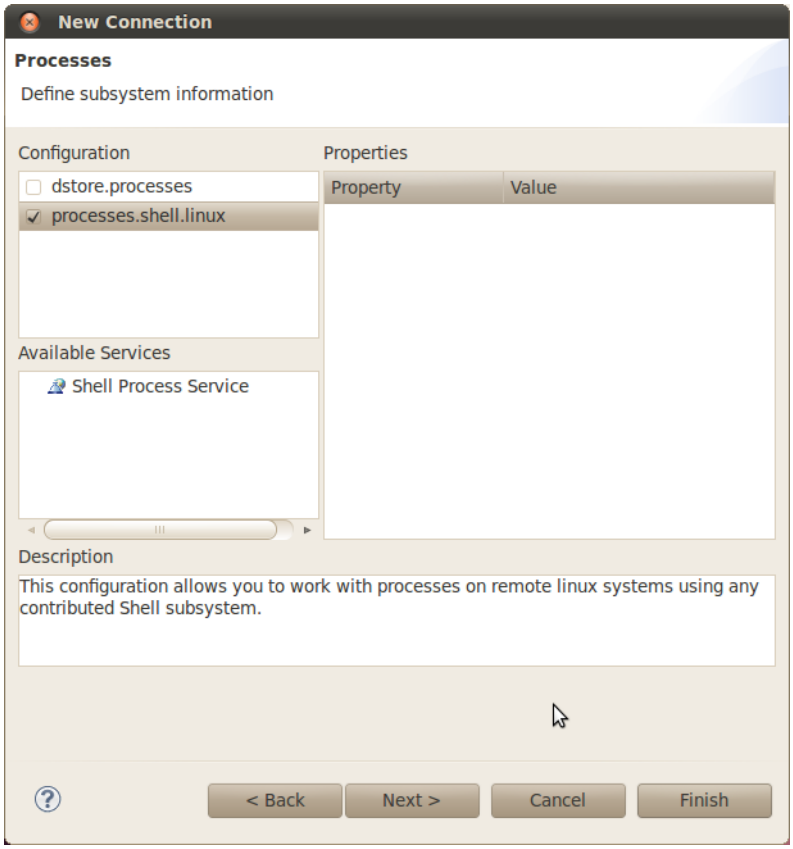
Next to "Host name" enter the IP address of your target EVM. This can be determined using a serial terminal program. Halt the boot process by typing a key immediately after power-on. At the U-BOOT prompt, type dhcp. In the response, look for "Our IP address is xx.xx.xx.xx". Substitute that IP address for the xx.xx.xx.xx shown below (and anywhere else in this Wiki article that xx.xx.xx.xx appears.) Next to "Host name" enter a connection name for your target EVM.



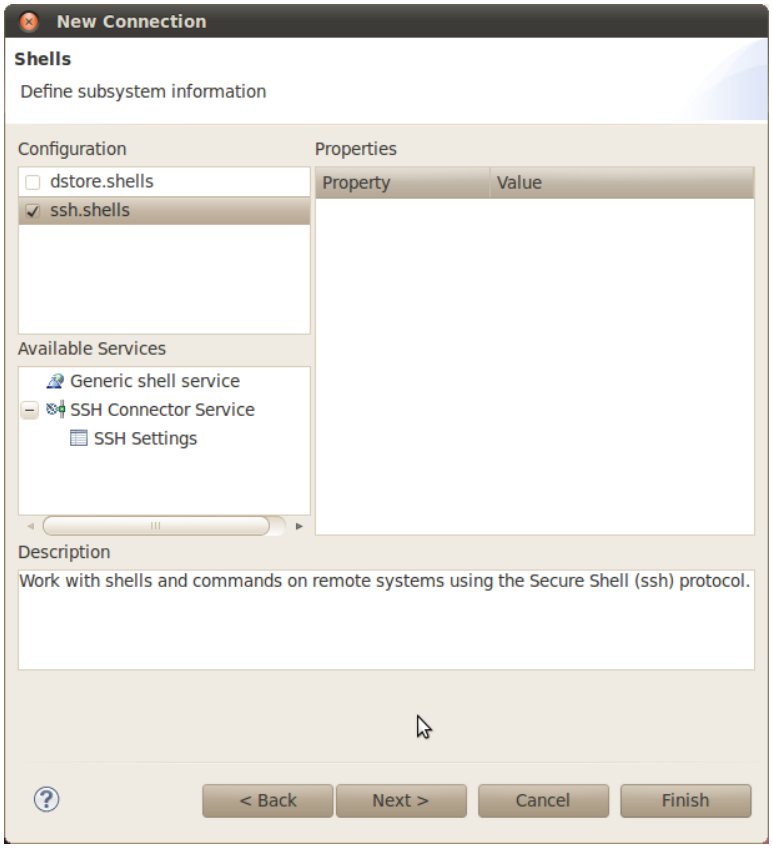
Check ssh.files and click Next.



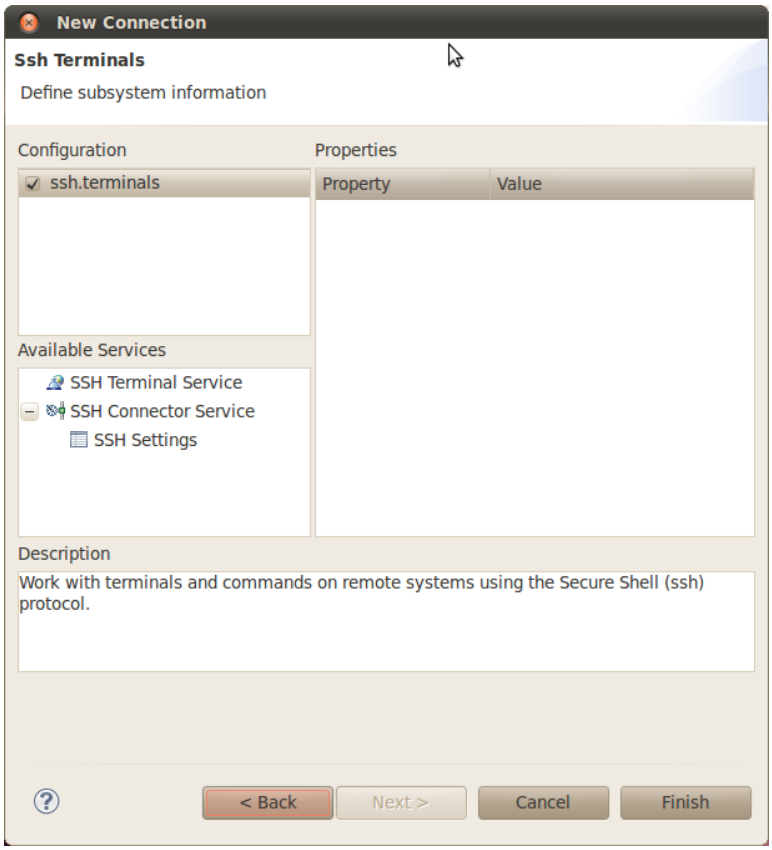
Check processes.shell.linux and click Next.



Check ssh.shells and click Next.

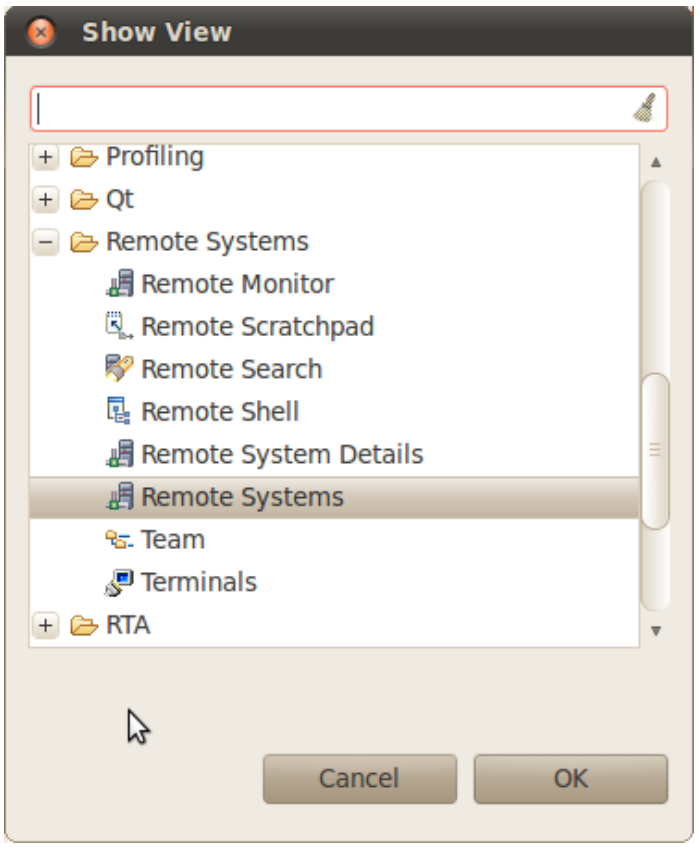


Check ssh.terminals and click Finish to complete the wizard.



Opening the Remote System Explorer View

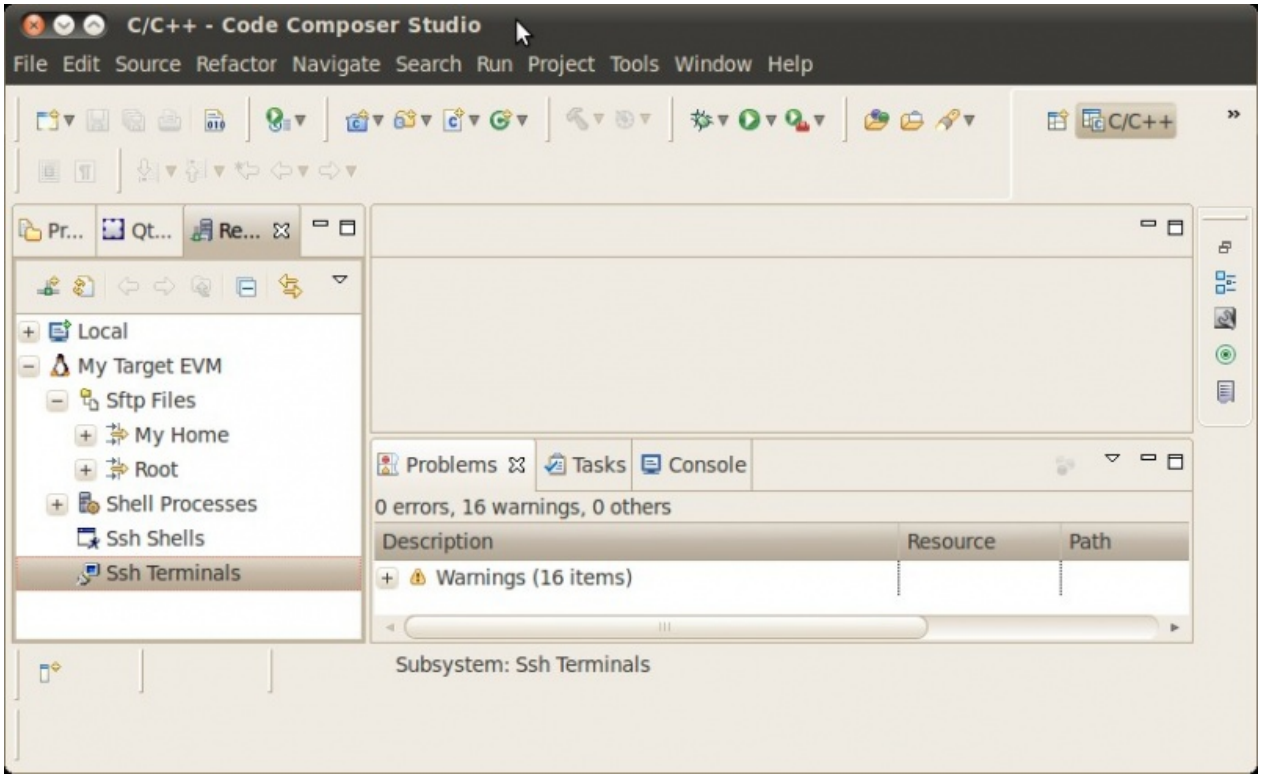
In CCSv5 click the Window -> Show View -> Other menu item. In the Show View dialog select Remote Systems -> Remote Systems. This adds the Remote Systems view to the current perspective.



A Remote Systems tab appears in the CCS perspective. After doing a R-Click Detach on the Remote Systems tab and moving it to the left side of the screen the CCS window will look like the screen capture below. The target connection named My Target EVM is shown in a tree structure with branches for the various Remote System functions. All communicate with the target EVM using a secure SSH connection.

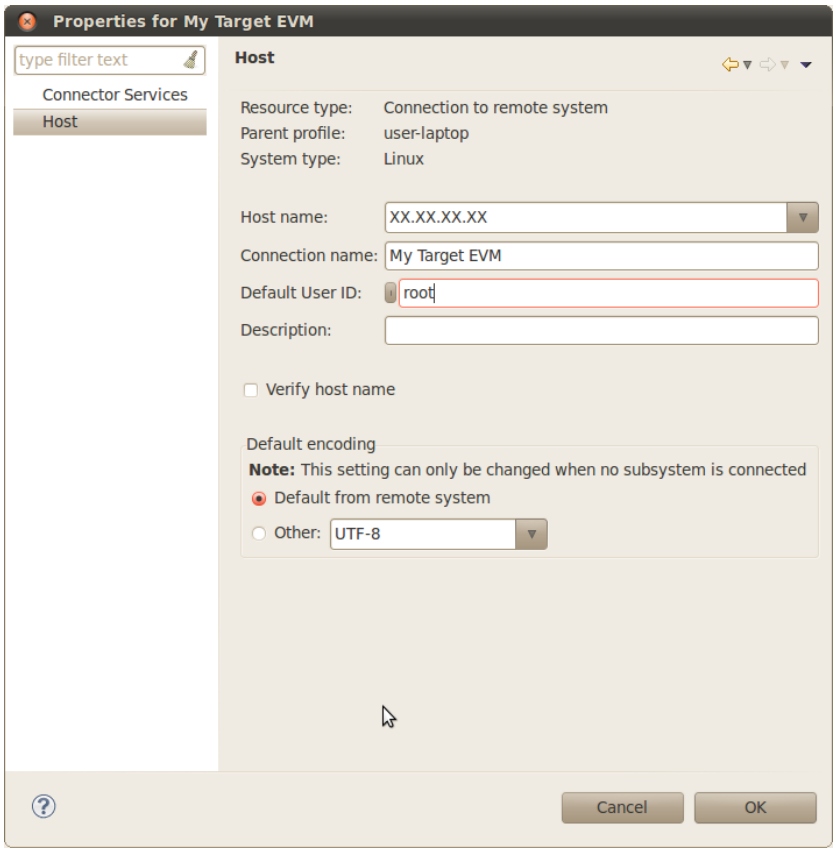
1. Sftp Files - Provides a drag and drop GUI interface to the target file system.
2. Shell Processes - Provides a listing of processes running on the remote system and allows processes to be remotely killed.
3. Ssh Shells - Provides a Linux shell window for the remote system within CCS.
4. Ssh Terminals - Provides a terminal window for the remote system within CCS.

If you have not yet configured the target EVM connection (see next section) the Remote Systems functions will not be functional yet. The timeout period before an error message appears can be as long as 60 seconds, so don't try them just yet.



Configuring the Target EVM Connection

After the New Connection Wizard has been completed and the Remote System Explorer view has been opened, the new connection must be configured to communicate with the target EVM. R-Click on the My Target EVM node and select Properties from the context menu. After the Properties window opens, click on Host. Change the Default User ID to root and click OK.

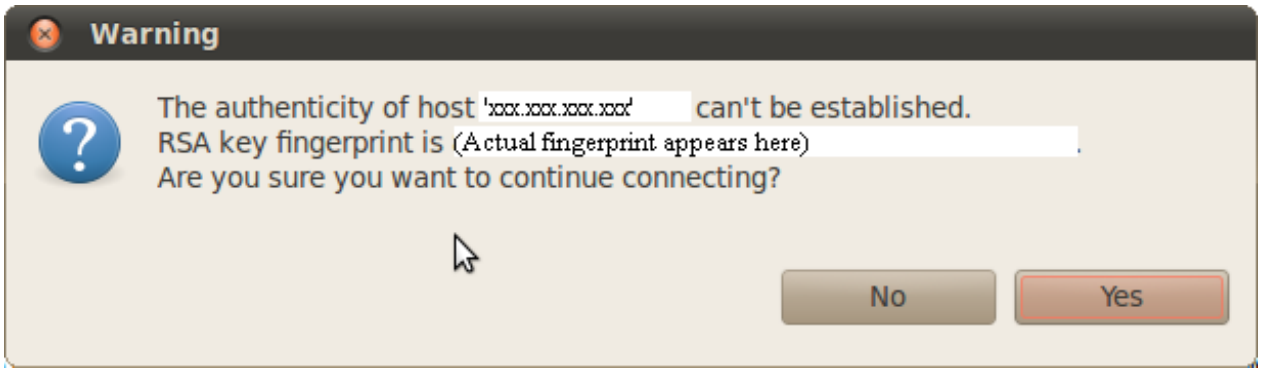


Click the Window -> Preferences menu item. Go to General -> Network Connections. In the top part of the dialog box, change the active provider from Native to Manual. Highlight the HTTP item and click the Edit button to enter your company's host proxy URL and port number. Do the same for the HTTPS item. Both items should be checked as shown below.

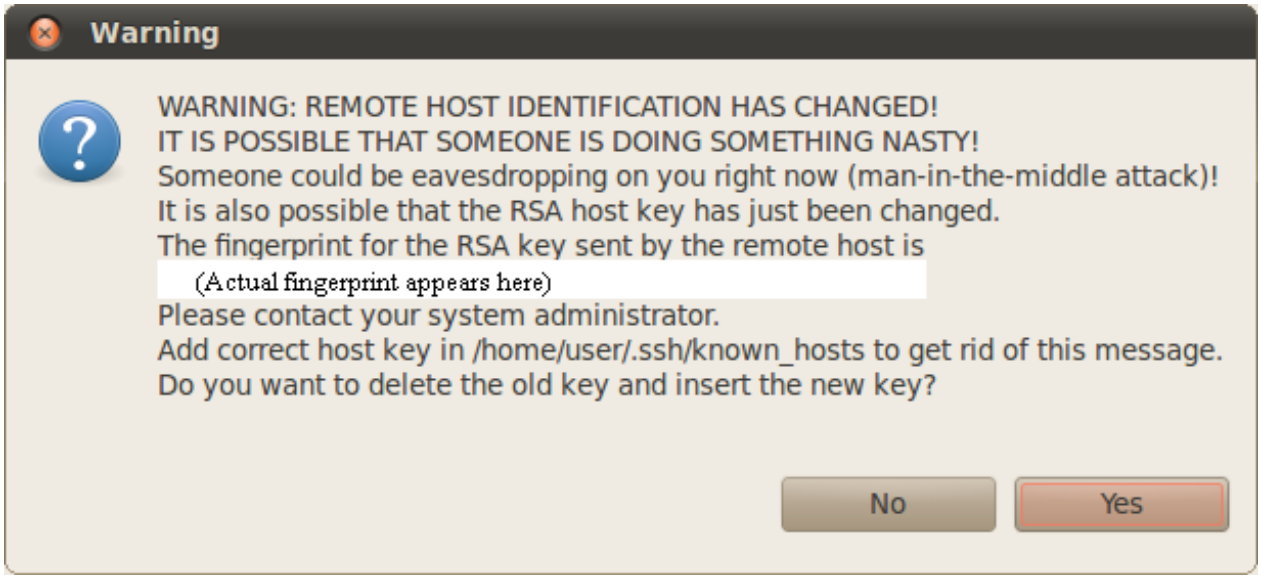
In the bottom part of the dialog box, in the Proxy Bypass section, click Add Host..., and add the IP address of target board (in place of xx.xx.xx.xx)

and click OK.

The Remote System Explorer is now ready for use. The first time the target EVM file system is booted a private key and a public key is created in the target file system. Before connecting to the target EVM the first time, the public key must be exported from the target EVM to the Linux host system. To open the SSH connection, expand the ROOT node under the Sftp node. A warning dialog box will appear as shown below. Click Yes, and the public key will be exported to the Ubuntu host.



Under certain circumstances a warning message can appear when the initial SSH connection is made as shown below. This could happen if the user deletes the target file system and replaces it with another target file system that has a different private RSA SSH key established (and the target board IP address remains the same). This is normal. In this case, click Yes and the public key from the target board will be exported to the Ubuntu host overwriting the existing public key.



At this point, all Remote System Explorer functions will be functional. After this, each time CCSv5 is started, the first time a Remote System Explorer function is accessed, a login prompt will appear. Just click OK and leave the password blank.

Using Remote System Explorer

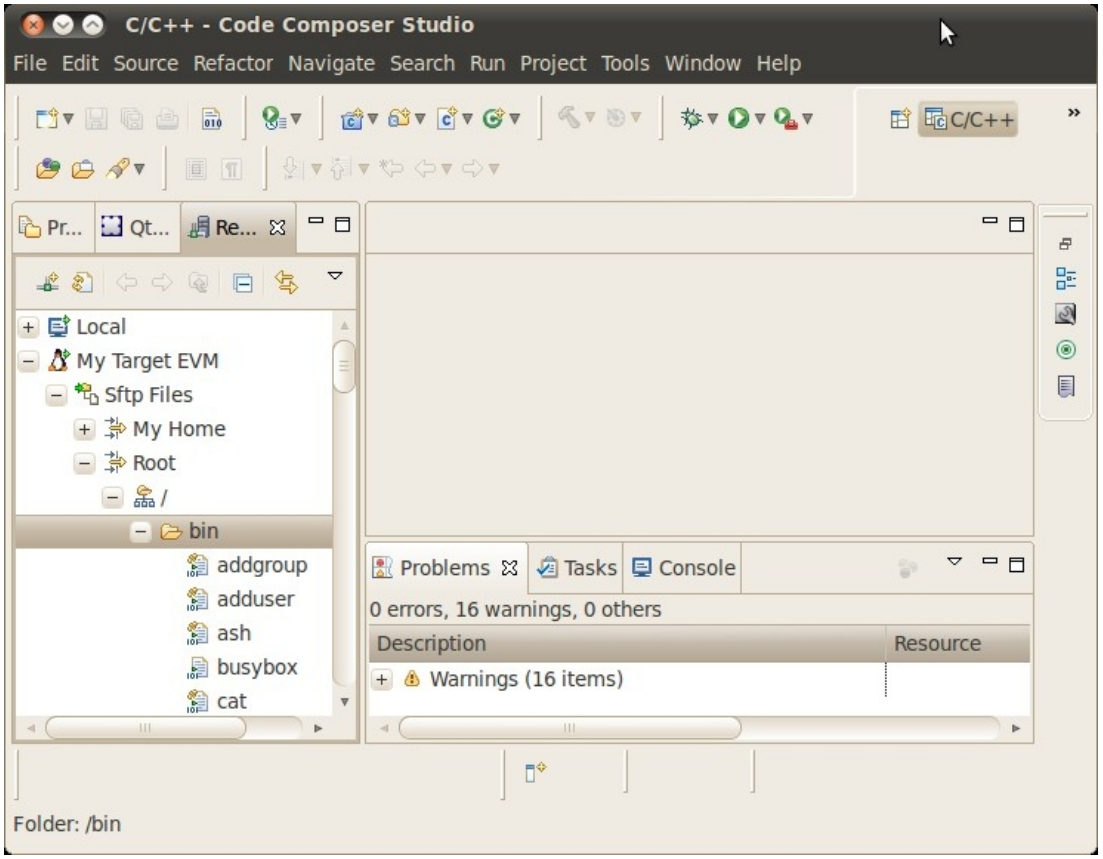
If Remote System Explorer is not included in the current CCS perspective, open the RSE view via the Window -> Show View -> Other dialog. Select Remote Systems -> Remote Systems.

The first time you access any of the Remote System functions under the Target EVM node a login prompt will appear. Type root for the User ID, leave the password blank and click OK.

This login prompt will also appear whenever CCSv5 starts up if the Remote Systems view is included in the current perspective.

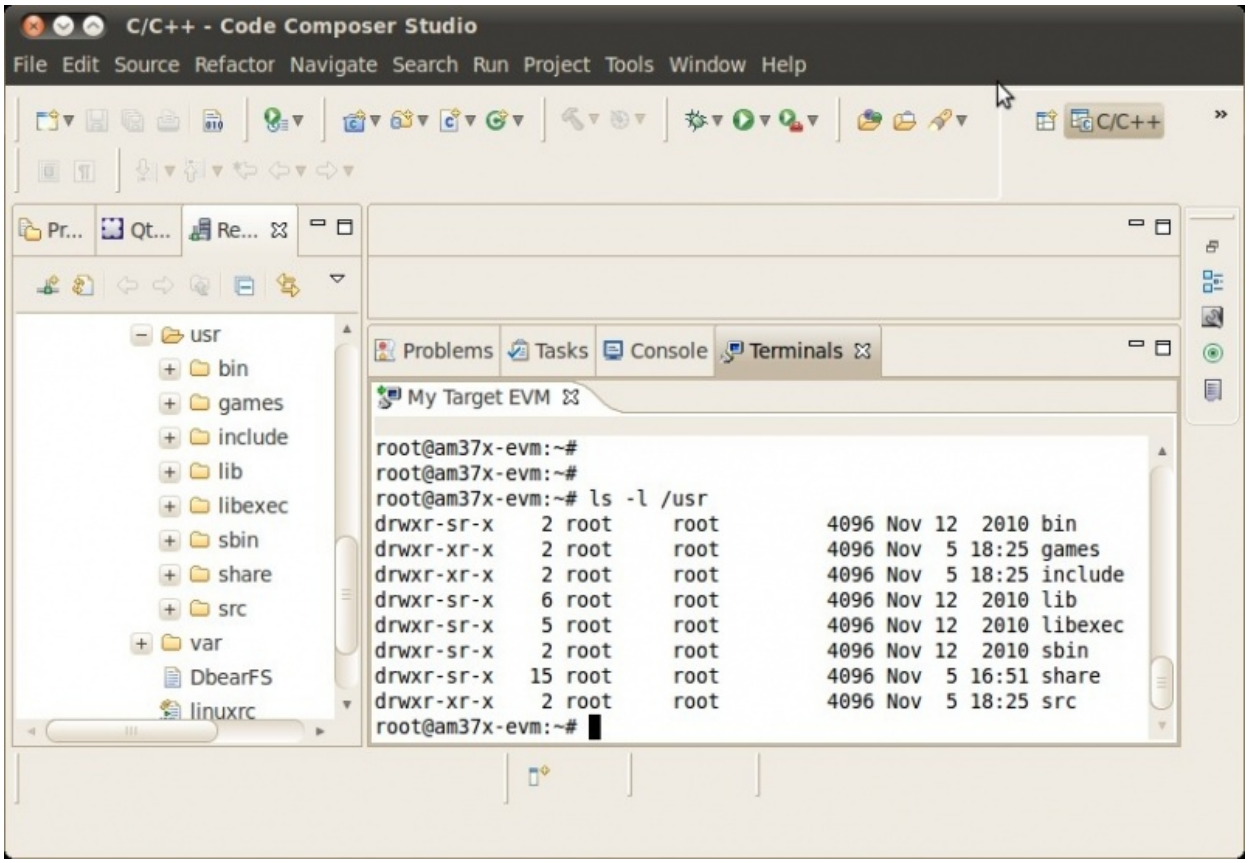
Target File System Access

Expand the ROOT node under the Sftp Files node. The remote system file tree should now show the root directory. You can navigate anywhere in the remote file system down to the file level. Files can be dragged and dropped into the remote file tree. A context menu allows you to create, rename or delete files and folders. The local file system on the Linux host can also be accessed by expanding the Local node.



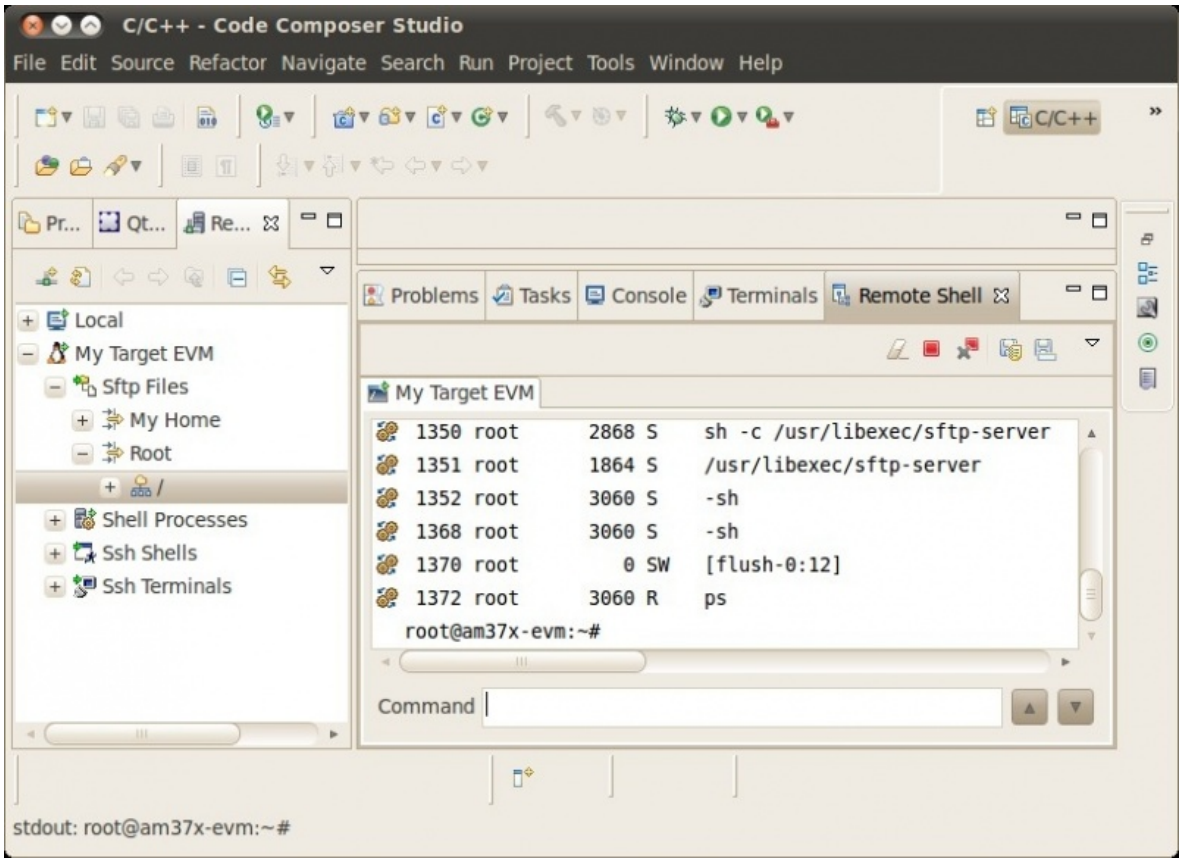
SSH Terminals

To open an SSH Terminal view, R-Click the Ssh Terminals node under the target EVM connection and select Launch Terminal from the context menu. Type shell commands at the prompt in the terminal window. Below is a sample command to list the contents of the remote /usr folder.



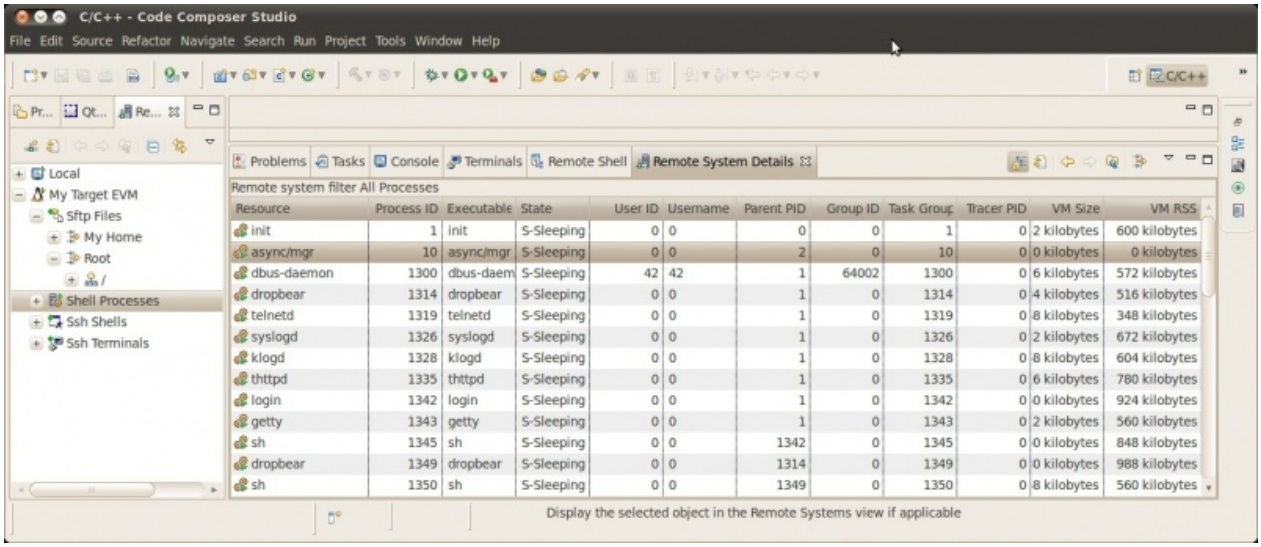
SSH Shells

To open an SSH Shell view, R-Click the Ssh Shells node under the target EVM connection and select Launch Shell from the context menu. Type a shell command into the Command edit box and press the Enter key. Below is the output from the ps command which displays the processes running on the remote system.



Shell Process Monitor

To open the Shell Processes view, R-Click the Shell Processes node under the target EVM connection and select Show in Table from the context menu. Double-click All Processes to see the list of processes running on the target. By clicking any column header, the list can be sorted by the parameter in the selected column, such as executable name, memory size, or PID. You can R-Click on a particular process and click 'Kill' in the context menu to kill a process.



How to Run GDB on CCSv5

This article can help you to start a gdb session from CCSv5

Preliminary Requirements

1. The Sitara SDK for your hardware platform has been installed on your Ubuntu host machine.
2. A serial connection between your host machine and the target board with a serial terminal emulator running on the host. Minicom is recommended. You could also use a telnet window to the target system.
3. An Ethernet connection between your host machine and the target board.
4. Boot up the target EVM. If there are any power-on applications running, exit from them.
5. Start CCSv5 and open the project you wish to debug.
6. The executable to be debugged must reside in the target file system. It must have been built from the debug build configuration so that it contains the symbol information.

NOTE: For this example, we are going to debug the matrix_tui project

Determining the Target Board Ethernet IP Address

1. Ensure that the Ethernet port on the target board is enabled:

```
root@dm3730-am3715-evm:~# ifup eth0
```

You should see a response that is similar to this:

```
net eth0: SMSC911x/921x identified at 0xc8860000, IRQ: 336

udhcpc (v1.13.2) started

Sending discover...

Sending discover...

Sending select for 128.247.107.145...

Lease of 128.247.107.145 obtained, lease time 28800

adding dns 128.247.5.10

adding dns 157.170.147.7
```

2. Get the target Ethernet IP address:

```
root@dm3730-am3715-evm:~# ifconfig
```

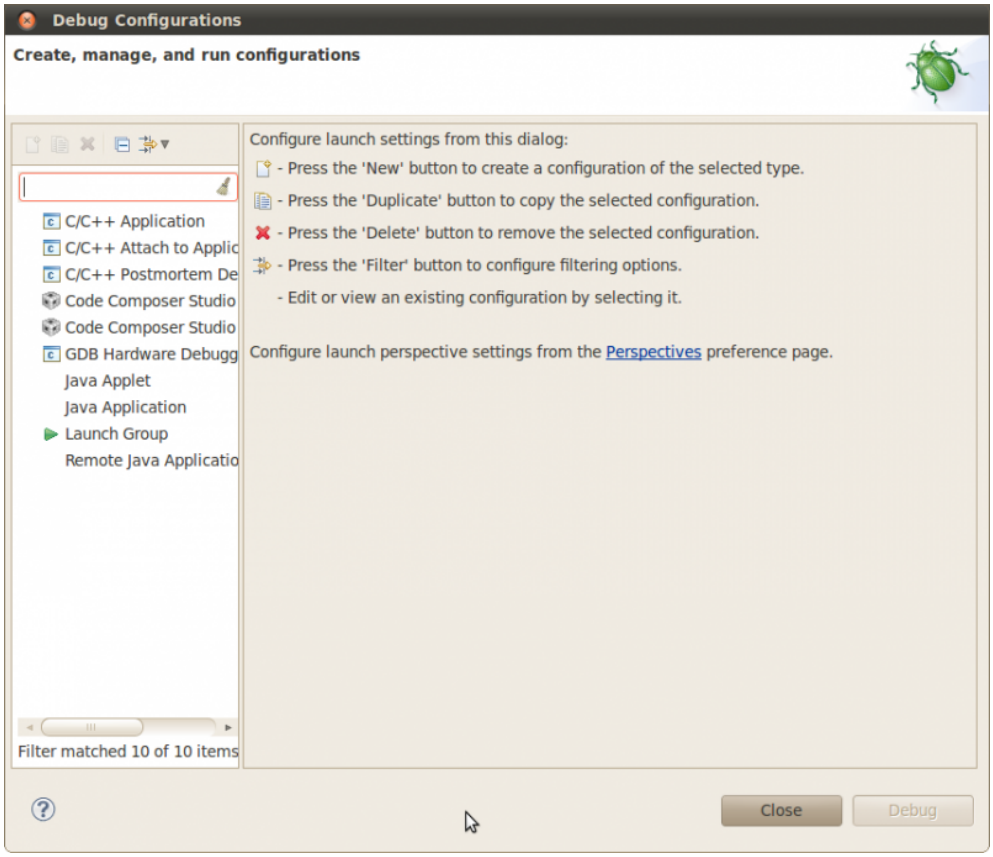
You should see a response that is similar to this. The actual target Ethernet IP address will appear where the xxx.xxx.xxx.xxx is shown below. (This IP address will be used in a later step):

```
eth0      Link encap:Ethernet  HWaddr 00:50:C2:7E:8F:D4

          inet addr:xxx.xxx.xxx.xxx  Bcast:0.0.0.0  Mask:255.255.254.0
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:24 errors:0 dropped:0 overruns:0 frame:0

          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000

          RX bytes:3174 (3.0 KiB)  TX bytes:1770 (1.7 KiB)

          Interrupt:80


lo        Link encap:Local Loopback

          inet addr:127.0.0.1  Mask:255.0.0.0

          UP LOOPBACK RUNNING  MTU:16436  Metric:1

          RX packets:0 errors:0 dropped:0 overruns:0 frame:0

          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:0

          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

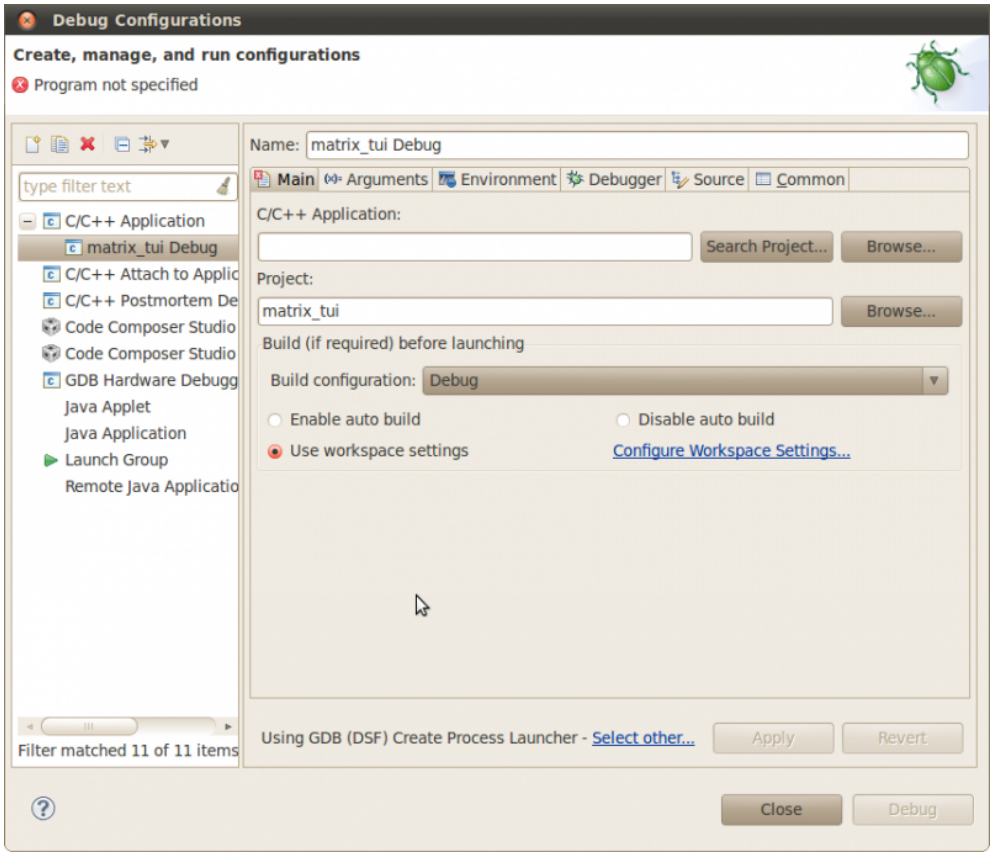
Creating the Debug Configuration for the Project

1. In CCS, select the project you wish to work with by clicking on it and highlighting it.
2. Select the Run -> Debug Configurations menu item. This opens a dialog box as shown below.

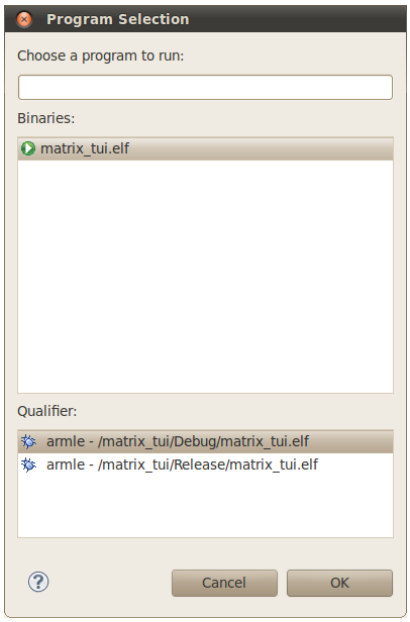


3. Double click C/C++ Application. You should then see a new debug configuration named "matrix_tui Debug" as shown below. Be sure that the

Build Configuration is set to Debug.

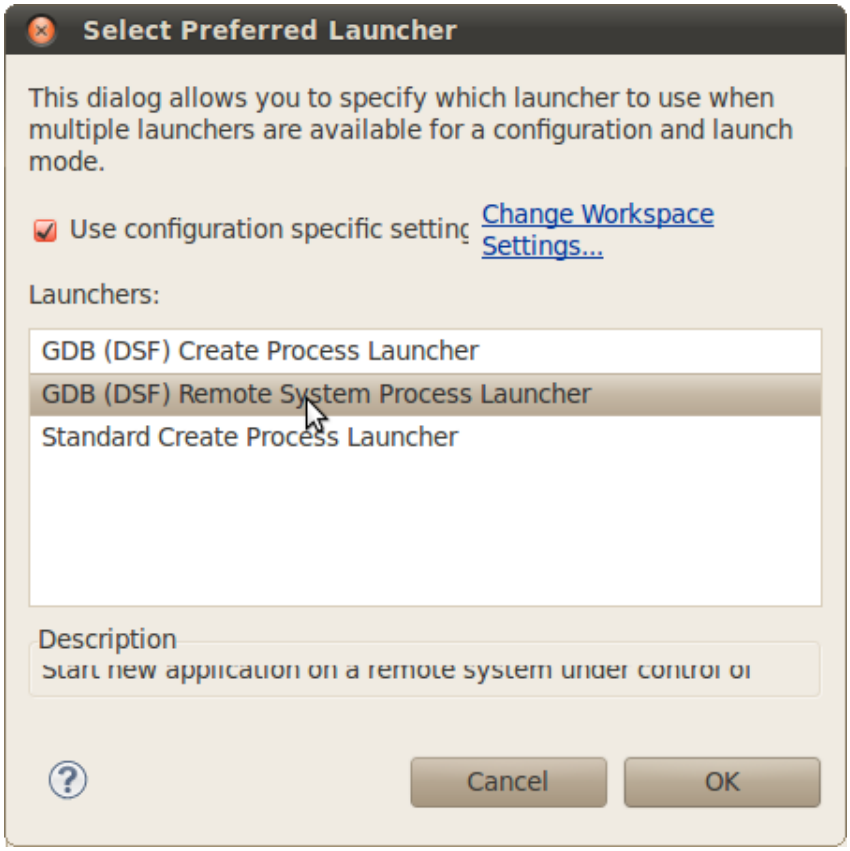


4. Click the Search Project button to open the Program Selection dialog box below. Click on the "armle - /matrix_tui/Debug/matrix_tui.elf" item and click OK.



5. Back to the Debug Configurations dialog box (still on the Main tab). Click Select Other to open the Select Preferred Launcher dialog box shown below.

Check the Use Configuration Specific Settings box. Select "GDB (DSF) Remote System Process Launcher" and click OK.



6. Back to the Debug Configurations dialog box. Click the Debugger tab. On the Debugger page, the Main tab should be selected.

Click Browse next to "GDB debugger" and browse to the Code Sourcery GDB executable. Click browse next to "GDB command file" and browse

to the .gdbinit file in the SDK install directory. (Actual SDK install directory name will depend on your hardware platform). When you try to browse

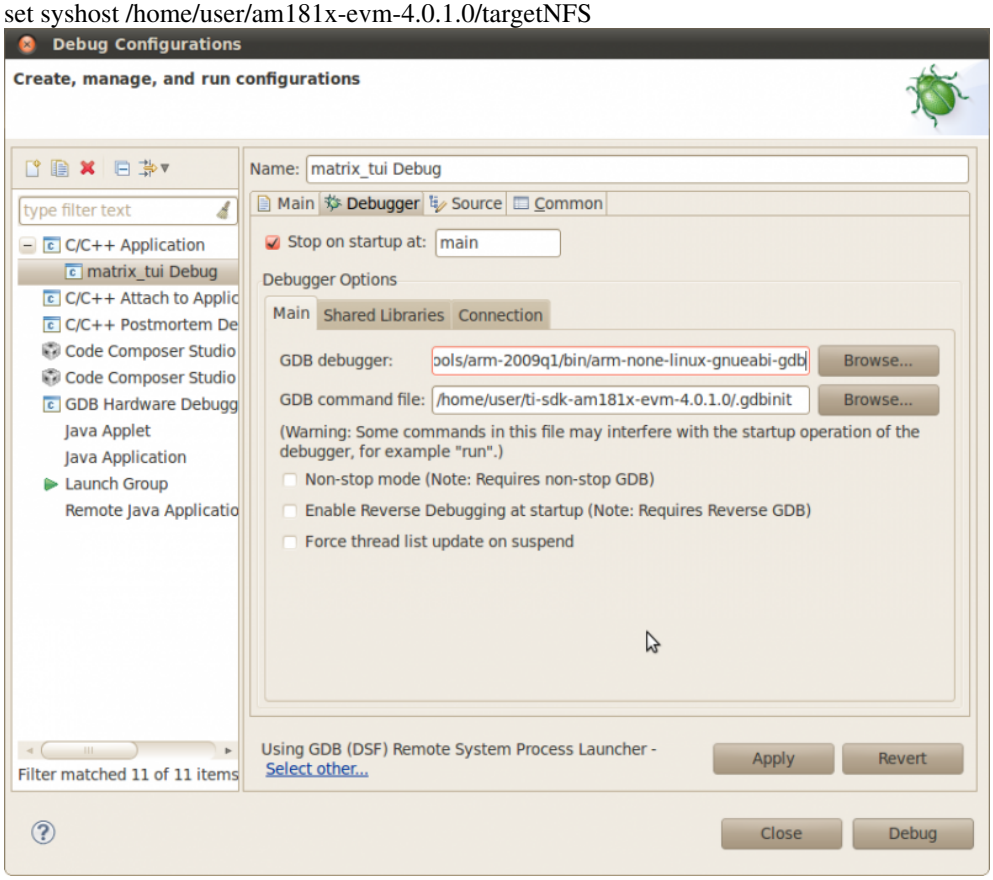
to the .gdbinit file, it will not show up because the name starts with a dot (it is a hidden file). You can select a different file in the same directory,

and then edit the resulting file name to .gdbinit as shown below.

The .gdbinit file is used by GDB to locate source files and library files on the target. The .gdbinit file is created when the SDK environment script runs.

An example.gdbinit file (for am181x platform) would contain this:

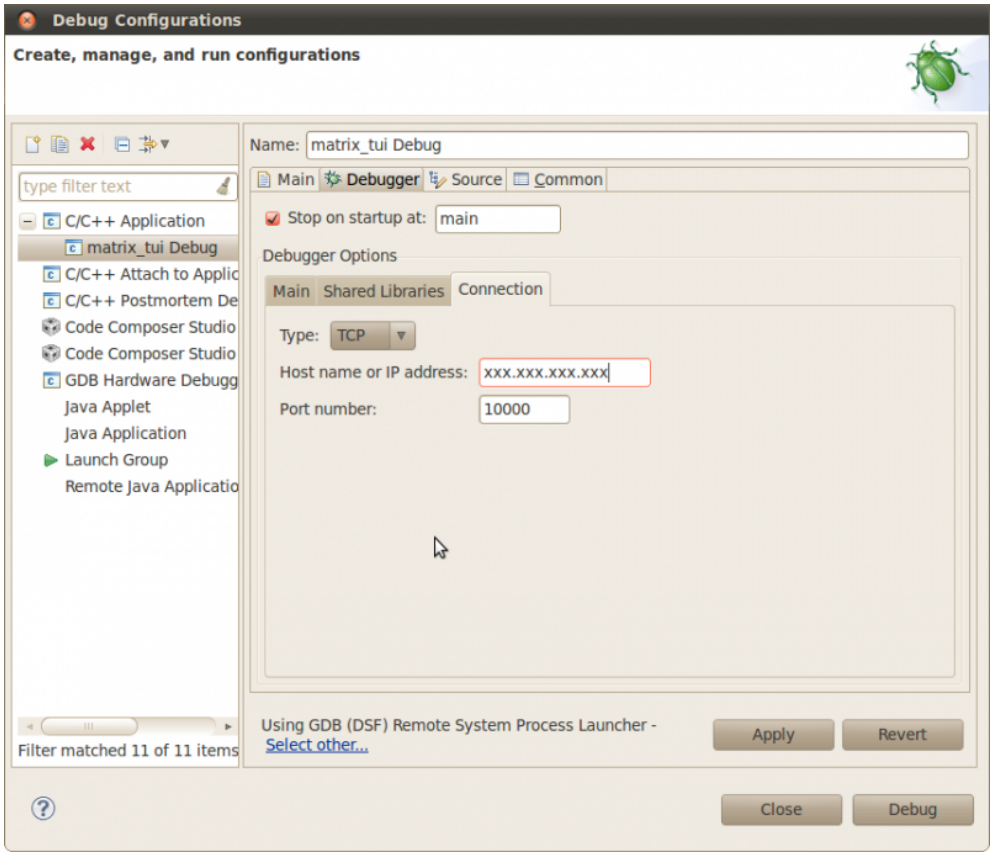
set syshost /home/user/am181x-evm-4.0.1.0/targetNFS



7. Still on the Debugger page, click the Connection tab. Change the Type to TCP. Enter the IP address of the target board (in place of

xxx.xxx.xxx.xxx) and enter port number 10000. (The required IP address is obtained by typing ifconfig at the target console, as discussed earlier.)

Click Close to close the Debug Configurations dialog box.



Running the Debug Session

1. Each time you start the debugger, you must first start the gdbserver program on the target. Start gdbserver for the matrix_tui project with a port number of 10000.

(This port number must match the number that was entered in the Debug Configuration earlier.) The path must be the path to the executable on the target board.

The executable on the target must have been built from the debug build configuration (which means it contains the symbol information.)

At the target console command line, type:

```
gdbserver :10000 /usr/bin/matrix_tui
```

Once started, you should see a response similar to the following 2 lines:

```
Process matrix_tui created; pid = 1506
Listening on port 10000
```

2. In CCSv5, click the "bug" icon to start the debugger. (The "bug" icon is directly below the word "Navigate" in the main menu.)

CCS will change to the CCS Debug perspective. The debug tab will show the running threads and their status. The source code window will show

the program halted at the first executable source code line in the main() function. The Variables window will show the local variables and their current

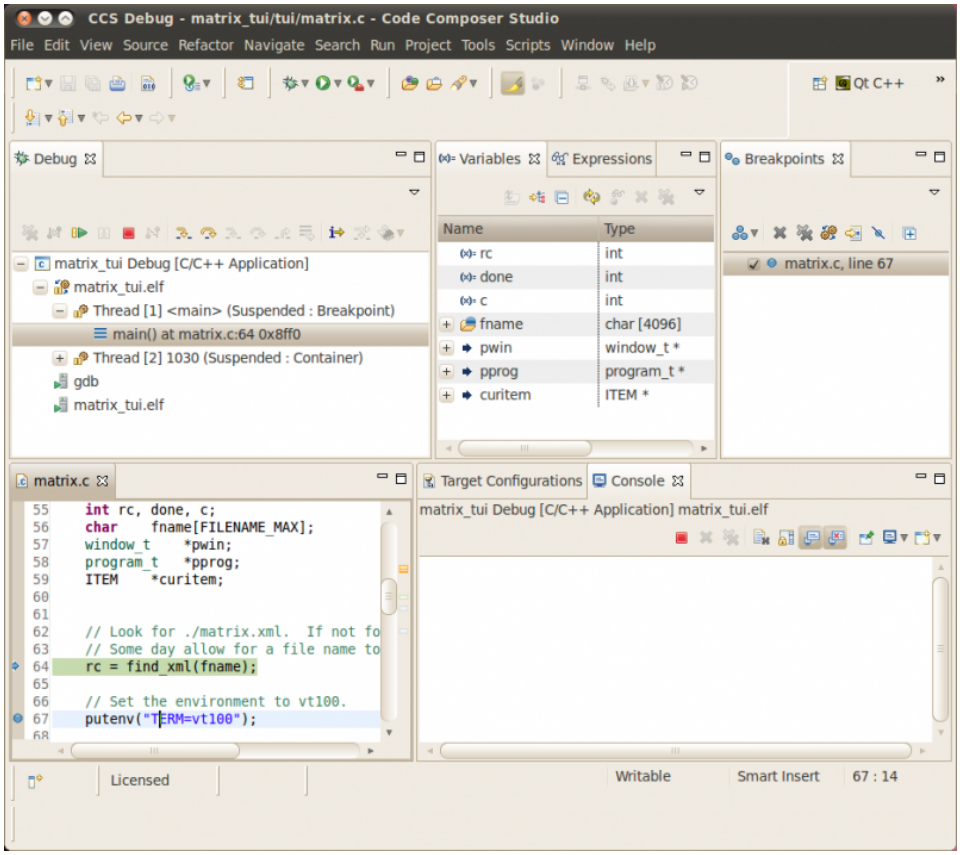
values.

3. To toggle a breakpoint, highlight the line of code in the source code window. Then click the Run -> Toggle Breakpoint menu item.

4. Use the debugger "Step Over" and "Step Into" icons to step through the source code.

5. To resume program execution, click the Run -> Resume menu item.

NOTE: Do not click the Run -> Debug menu item, as that will attempt to start a new debug session.



Stopping the Debug Session

When finished debugging the matrix_tui application, click the Run -> Resume menu item. At the target console, hit the escape key until the

matrix_tui program exits normally. If the program exits normally, the target console will return to the command prompt. Another way to

stop the debug session is to click the Terminate icon in CCS (this icon is a red square).

NOTE: There is currently a problem with gdbserver. If the program being debugged ends abnormally or crashes, the target console running

gdbserver will no longer respond. In that case, the target board must be rebooted to start another debug session.

To avoid having to reboot the target system, the Remote System Explorer Shell Processes feature can be used to terminate the

gdbserver process.

See processors.wiki.ti.com/index.php/How_to_setup_Remote_System_Explorer_plug-in [3] to setup the Remote System Explorer plug-in.

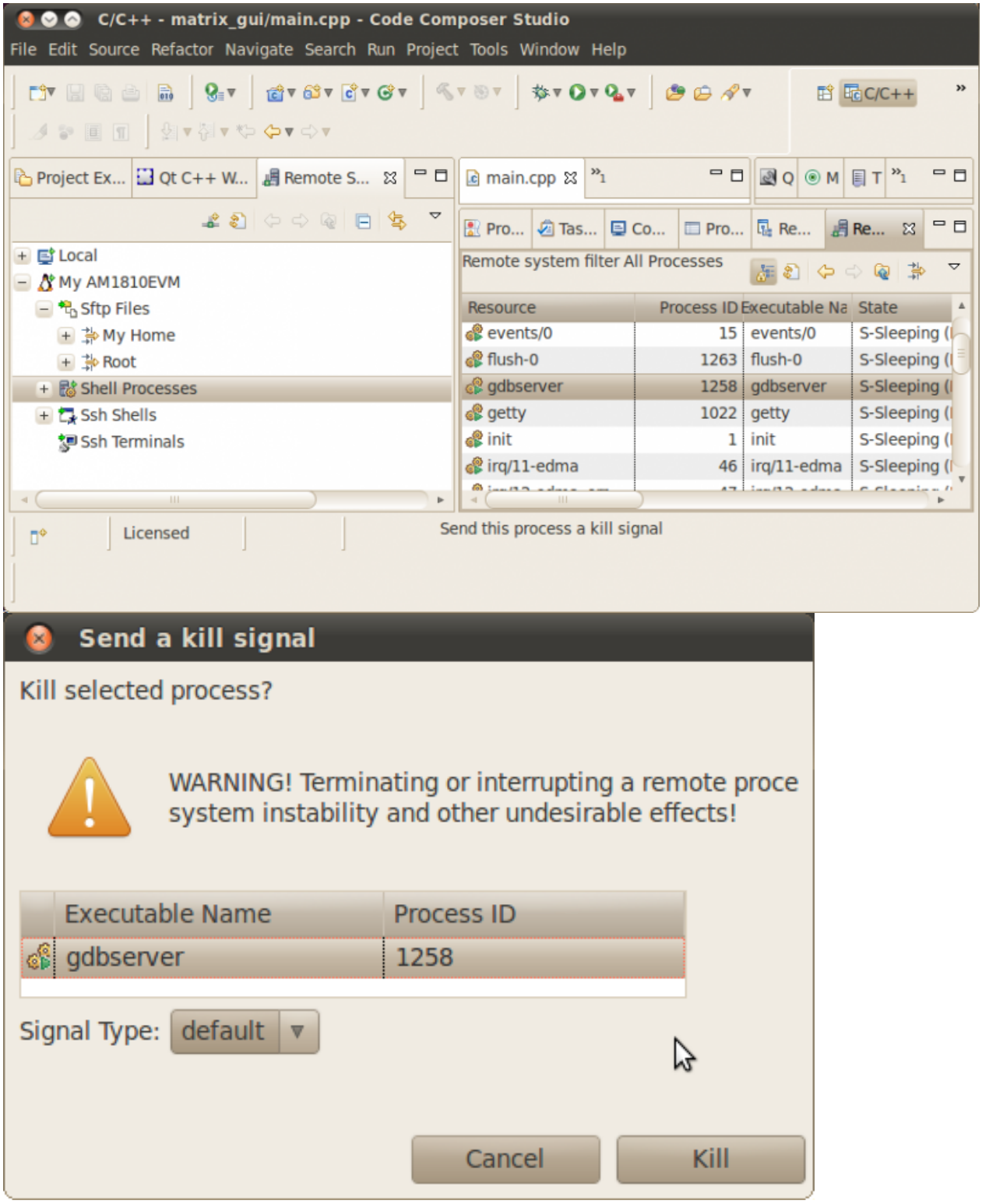
Once setup you can follow these steps to terminate gdbserver:

1) R-Click on Shell Processes and select Show in Table to open a Remote System Details window.

2) Double-click on "All Processes" in the table to display the list of processes runnning on the target system.

3) Click on "Executable Name" in the table headers to sort the list by executable name.

4) Find the gdbserver process. R-Click on it and select Kill. This will open a "Send a Kill Signal" dialog box. Click the Kill button.



Matrix Users Guide

Initial Boot Up

When you first boot up an EVM containing a Sitara Software Development Kit (SDK), Matrix GUI should be automatically started. Matrix GUI can be either operated by touchscreen or mouse. Default startup for most SDK platforms is touchscreen. Should you encounter any problems below are some tips to get everything running smoothly. See #Matrix GUI Startup Debug

Overview

Matrix is an application launcher created to highlight available applications and demos provided in new Software Development Kits. There are two forms of Matrix, a Graphical User Interface (GUI) and Text User Interface (TUI). All of the example applications and demos are available using either the graphical or text version. The graphical version launches by default when the EVM is booted and uses the EVM's touchscreen interface for user input. Optionally, the user can exit the graphical interface and run the same applications using the text version. The mechanical push buttons on the EVM (located beside the LCD) are not used by either version. Matrix GUI comes as a 4x3 matrix of icons or as a 4x2 matrix depending on the display resolution.

Matrix Graphical User Interface

How to Use the Matrix GUI

The Matrix GUI is based on HTML and Cascading Style Sheets (CSS) and designed to be easily cusomizable. You can find a set of HTML files in your SDK at /usr/share/matrix/html. These files contain all the information to generate all the menus and submenus you see displayed by the Matrix GUI. They also contain all the information to launch the applications. It is easy to customize Matrix GUI and add any number of new applications.

Matrix GUI Components

Below is a summary of all the Matrix GUI components:

Menu Pages

- Menu pages are at level 0 and can contain icons which link to either submenus or applications.
- Menu pages contain an exit icon which causes the Matrix GUI to terminate.
- You can add any number of additional Menu pages, so you can have any number of submenus and applications.
- The initial menu page 1 is defined by the file menu_main_1.html.

Submenu Pages

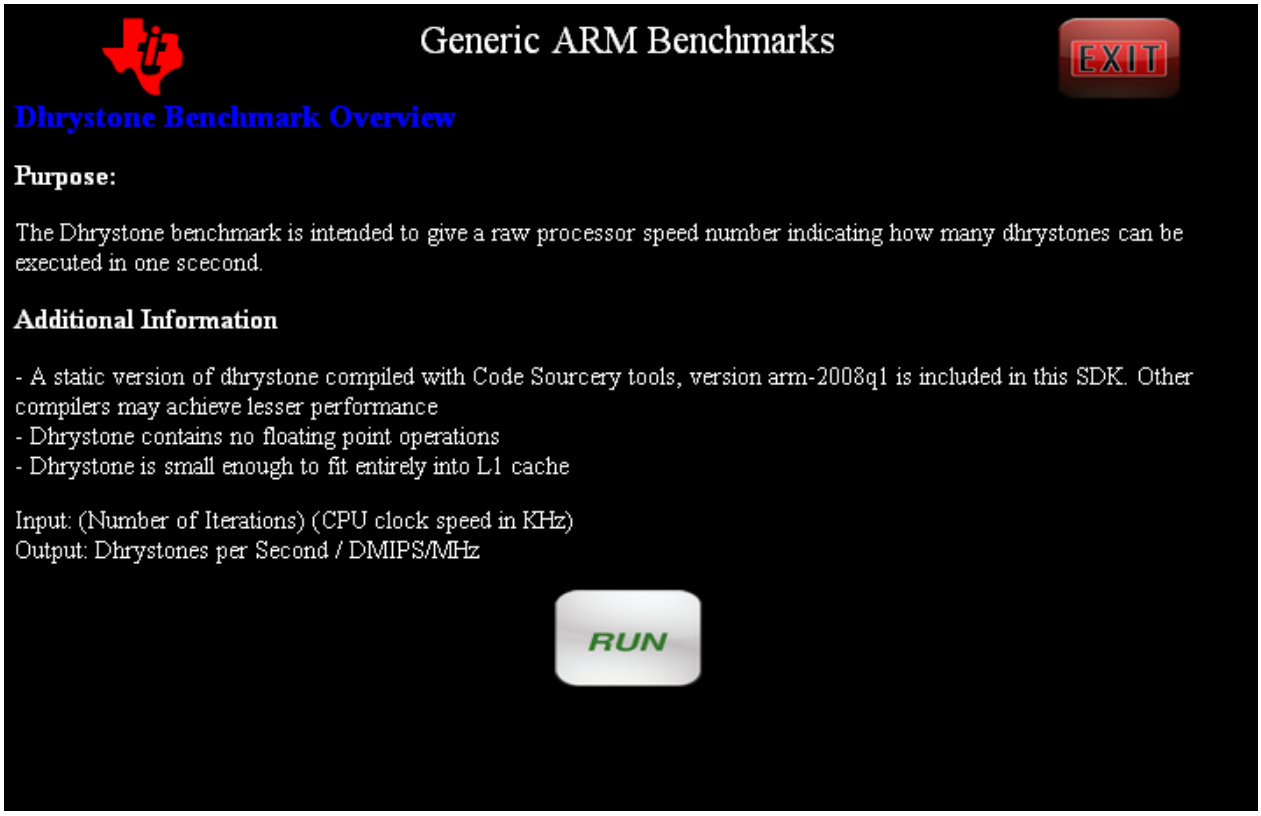
- located at level 1 can can only contain icons which link to applications.
- contain close icons which close the submenu and return to the main menu.
- can add any number of additional Submenus, so you can have any number off applications associated with a particular submenu.

Application Description Pages

- optional and associated with a particular application.
- provide additional information which can be useful for various reasons
- displayed when the associated application icon is pressed.
- can be disabled by turning off description display mode.

Example Application Description Page

Below is an example application description page. They can be used to add addition information that may not be obvious.



Icons

- 96x96 pixel image files which are associated to a submenu or an application in an HTML file.
- can be re-used by many applications
- blank icons in different colors available for customization
- found in your SDK target filesystem under /usr/share/matrix/images
- found in your SDK host development area at example-applications/matrix-gui

HTML Files with Cascading Style Sheets

- one HTML file generates 1 menu, submenu or description page.
- easy to edit and add new applications, additional main menus, additional submenus and application descriptions.
- pulls in the Cascading Style Sheet, matrix.css, to ensure a common look and feel.
- found in your SDK target filesystem under /usr/share/matrix/html
- found in your SDK host development area at example-applications/matrix-gui

Applications

- Any application can be launched by Matrix GUI
- Matrix GUI uses the graphics display layer. If a launched application also uses the graphics display layer there will be a conflict.

Matrix GUI Launcher

- Designed using QT/WebKit/C++
- no need to edit the Matrix GUI project to add applications, or new menu pages.
- Matrix GUI project available in the SDK host development area under example-applictions/matrix-gui.

Launching Matrix

Use the following shell script in the SDK to run Matrix as a background task:

```
/etc/init.d/matrix-gui-e  start
```

This script ensures that the touchscreen has been calibrated and that the Qt Window server is running and configured with a 90 degree rotation so that the menus will be displayed in a landscape orientation. This is required if the EVM has a portrait LCD and VRFB rotation has not been enabled.

Alternatively, the Matrix GUI can be launched manually with this full syntax:

```
matrix_gui  -qws  -display transformed:Rot90  menu_main_1.html
```

The “-qws” parameter is required to start the Qt window server if this is the only/first Qt application running on the system.

The “-display transformed:Rot90” parameter causes the Qt windowing system to rotate the display 90 degrees.

The third parameter is the name of the file that contains the top menu in the menu hierarchy. Since there is only room on the display for 12 buttons, the top menu (menu_main.html) consists mainly of buttons to launch submenus. Each submenu has a “Close” button which will close that menu and return to the parent menu.

Matrix GUI Startup Debug

The following topics cover debugging Matrix GUI issue at startup or disabling Matrix GUI at start up.

Touchscreen not working

Type the following commands at the console window:

```
/etc/init.d/matrix-gui-e stop
```

This will stop the matrix_gui from running

```
ts_calibrate
```

This will run the touch screen calibration routine. Follow the prompts on where to touch the screen. If you should accidentally double touch, just repeat this step.

```
/etc/init.d/matrix-gui-e start
```

Start the matrix_gui running again.

Matrix is running but I don't want it running

1. Just exit matrix normally, by pushing the exit button on the main menu of the touchscreen LCD
2. Or if the touch screen is not working, from the console, type:

```
/etc/init.d/matrix-gui-e stop
```

I don't want matrix_gui to run on boot up

From the console type the following commands:

```
 cd /etc/rc5.d
 mv S99matrix-gui-e K99matrix-gui-e
```

This will cause matrix_gui to not automatically start on boot up.

Reference Documentation

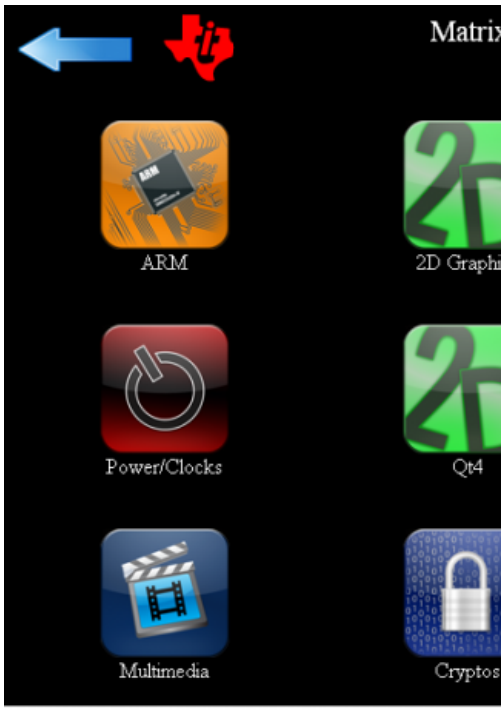
- → How to Make a Hello World Matrix Application
- More coming soon...

Matrix Enhancements

Since the intial matrix release in the AM37x SDK in March 2010, there have been enhancements added. The most significant ones are summarized below:

1. Auto resize of matrix-gui inteface using Cascading Style Sheets (CSS) with HTML. This allows matrix to be ported easily to different platforms with any display resolution.
2. Allow umlimited addition of main menu pages and sub-menu pages.
3. Add an optional description to applications. This can give valuable information about important feature of the application.
4. Add a control menu bar at the top of the matrix interface to allow navigation between the new addtional pages and also move exit to the control bar.

Below are the differences between the orignal and enhance Matrix GUI main menu:



How to Import the Matrix GUI into CCSv5

You can find how to import the Matrix GUI here: → How to Import Matrix GUI into CCSv5

How to Build the Matrix GUI

You can find how to build the Matrix GUI here: → How to Build the Matrix GUI

How to Enable Mouse Instead of Touchscreen for the Matrix GUI

You can enable mouse by referring to the following: How to Enable Mouse for the Matrix GUI [1]

How to Switch Display from LCD to DVIout for the Matrix GUI

You can switch the display output by referring to the following: How to Switch Display Output for the Matrix GUI [2]

How to Customize the Matrix GUI

This documentation is based on the second version of Matrix GUI. If you have the original Matrix Gui reference this article: Matrix Gui Version 1

The C++ source code for the Matrix GUI application is provided with the SDK package but it is not necessary to edit or recompile this to customize the application menus. All that is usually required is to edit the HTML files that define the menus.

It is important to understand that Matrix is basically a specialized web browser built upon the Qt SDK and WebKit. Each menu is an HTML page that is displayed by this browser. The HTML pages define the buttons that are displayed to compose each menu. Each button can either launch an example application, open a submenu, close a submenu or display a general webpage from the internet.

Each menu file for Matrix contains a 4 x 3 table and each cell in these tables contain a special object tag named "application/x-matrix". This object tag will not be recognized by general purpose web browsers because it requires a custom plug-in which is built into the Matrix browser. Each occurance of this object tag causes Matrix to generate an active button complete with a displayed icon, text label and a path to a corresponding executable application that will be used if/when that button is pressed.

To create a new menu page, simply copy one of the existing files with a new name and edit the HTML table cells. To launch your new menu from an existing menu, edit a table cell in the existing menu file so that its “appName” parameter is “Submenu” and its “appParameter” has the name of your new menu page.

This is the top part of the HTML. The next section controls the menu bar at the top of the display.

Cascading Style Sheet

There is one Cascading Style Sheet, matrix.css, that controls all the Matrix GUI menu displays. It is read in to each HTML file - see HTML Header.

- Controls look and feel
- Automatically spaces the icon and centers the labels with the icons.
- Controls font size

In order to give an idea what is needed to it is to move the Matrix GUI to a different resolution display while maintaining the same look and feel, an example is shown below. The example matrix.css file below is for a VGA 640x480 resolution. The comments in red are the modifications needed to move the Matrix GUI with a 4x3 matrix of icons on a VGA resolution to a 4x2 matrix of icons on a wQVGA 480x272 resolution display.

```
*{color: #ffffff;} /* Default all text to white */

/* Set the background color to black and do not allow highlighting */
body {background-color: #000000; -khtml-user-select:none;}

/* This section controls the text title of a matrix page */
#header {
  font-size: 22px;       /* font-size: 16px; */
  width: 60%;
  height: 10%;
  float: left;
}

/* This section controls the 4 object in the top menu bar, the Left
right arrows, the TI logo and the Close/Exit button */
div.topBar object {
  height: 10%;
  width: 10%;
  float: left;
}

/* This secton controls both the icon image and the text label together
 */
div.object
{
  text-align: center;
  float: left;
  background-color:#000000;
  width: 25%;
  height: 30%;           /* height: 45%; */
}

/* This section control only the icon */
div.object object
{
  width: 96px;          /* width: 64px; */
  height: 96px;         /* height: 64px; */
  padding-top: 8%;      /* padding-top: 15%; */

}

/* This section controls only the label that goes with the icon */
div.desc
{
  font-size: 16px;    /* font-size: 12px; */
  width: auto;
  height: auto;
```

```
  background-color:#000000;
}

/* Everything below this comment is for application description pages
*/
/* The following control the body of the application description */
h1 { font-size: 18px; color: blue; }   /* font-size: 12px; color: blue;
 */
h2 { font-size: 16px; }               /* h2 { font-size: 10px; */
p { font-size: 14px; }                /* font-size: 10px; */

/* The following sections control the run icon of the application
description */
div.run object
{
  text-align: center;
  width: 12%;
  height: 12%;
}
div.run
{
  text-align: center;
  background-color:#000000;
}
```

HTML Header

This part of the HTML file pulls in the matrix.css, the one Cascading Style Sheet that controls all the Matrix GUI displays. This section should not need to be edited. `<html> <head> <title>Matrix Application Launcher</title> <link rel="stylesheet" href="/usr/share/matrix/html/matrix.css" /> </head>`

HTML Menu Bar

This part of the HTML controls the menu bar located at the top of the display. It is made up of 5 items from left to right:

- Left arrow used to navigate to the previous page (In the example below there is no previous page so no arrow)
- A TI logo icon with no functionality
- A menu page description
- A close or exit icon - exit-icon.png for menus and multi-icon.png for submenus
- Right arrow used to navigate to the next page. (In the example below there is a page 2 to the right)

```
<object >
</object>
```

```
<object type="image/svg+xml" data="/usr/share/matrix/images/tex.svg"
 >
<img src="/usr/share/matrix/images/tex.svg" />
```

Matrix Application Launcher p1

```
<object type="application/x-matrix" >
<param name="iconName"
value="/usr/share/matrix/images/exit-icon.png" />
<param name="appName" value="Close" />
</object>
```

```
<object type="application/x-matrix" >
<param name="iconName"
value="/usr/share/matrix/images/right-arrow-icon.png" />
<param name="appName" value="AddPage" />
<param name="appParameters"
value="/usr/share/matrix/html/menu_main_2.html" />
</object>
```

Object Tag Parameters

Required Tags:

- iconName - Path and name of the image file for the icon for this button. This can be a PNG or BMP format file.
- appName - This parameter can be one of the following:
  - The path and name of an executable application to launch.
  - The keyword "Submenu" which indicates to launch the menu named by appParameters.
  - The keyword "Close" which indicates this button will exit Matrix GUI for a main menu and close the submenu and return to the main menu.
  - The keyword "AddPage" which is tied to a blue left or right arrow that when pressed navigates to the next/previous main menu.
  - The keyword "AddSubPage" which is tied to a blue left or right arrow that when pressed navigates to the next/prevous submenu page.

Optional Tags:

- appParameters - This can define optional parameters for the executable application or menu file name or URL of webpage, depending on appName. This entire line can be deleted if no parameters are needed.
- appText - This flag parameter controls whether a text window will be displayed when the application is launched to display the stdout text generated by the application. The value for this flag can be set to either “Enable” or “Disable”. If this parameter is not defined, the default configuration is to enable the text window, but only if the application actually generates stdout text during its execution. This window will have a close button in the bottom right corner which must be pressed to close it. Setting this parameter to “Disable” prevents the text window from being created and it causes the parent menu to be disabled before the application is launched. This is important for non-Qt applications because they may otherwise assume control of the display without knowledge of the Matrix menu that launched it. This could allow the buttons on the obscured menu to still be active even though they are not visible. Setting this parameter to “Disable” prevents this problem.
- appDesc - This defines an optional HTML description page which is associated with a particular application. Description are not associated with submenus. See example description page below:

Object Tag Examples

There are 2 types of object tag entries that can be selected by the “appName” parameter; “Submenu”, “Close” or a named external application executable. The “appParameters” parameter should define any parameter(s) that are required for the application.

Submenu Button

Here is an example which generates a Submenu with a button labeled “Cryptos”. This button will launch a submenu for crypto related applications defined by the file named “menu_cryptos_1.html” when it is pressed.

```
<object type="application/x-matrix" >
<param name="iconName"
value="/usr/share/matrix/images/encrypt-icon.png" />
<param name="appName" value="Submenu" />
<param name="appParameters"
value="/usr/share/matrix/html/menu_cryptos_1.html" />
</object>
```

Cryptos

Application Launch Button

Here is an example application labeled "Browser" which will launch an QT Browser application named "browser" with the input parameters "http://ti.com".

```
<object type="application/x-matrix" >
<param name="iconName" value="/usr/share/matrix/images/web-icon.png"
 />
<param name="appName" value="/usr/bin/qtopia/demos/browser/browser"
/>
<param name="appParameters" value="http://ti.com" />
<param name="appText" value="Disable" />
</object>
```

Browser

Note that the "appText" parameter is used to disable any text window that might otherwise be displayed. This also ensures that the parent menu is disabled before the external application takes control of the display.

How to Use the Matrix TUI

The Matrix TUI is a text-based user interface program launcher. It is based on the ncurses library. The ncurses library is a freeware version of the curses library developed in 1980 to provide screen oriented user interfaces. All keyboard input is done on your serial console using the arrow keys and the enter key.

To launch the Matrix TUI, go to your serial console and simply type in
matrix_tui

You will be presented with the main window that lets you select sub windows or programs.

```
Main Window

 * ARM Benchmarks A selection of ARM benchmarks to run
   2D Demo        2D Graphics Demo
```

```
   3D Demos       OpenGL ES Graphics Demos
   Power/Clocks   Set various voltage/frequency
   MM Demos       Multimedia Demos
   USB            USB Testing
   Ethernet       Ethernet Testing
   Settings       System Settings and Information
   Crypto         Crypto Testing
   Shutdown       Shut down the EVM

Press ESC to exit
```

Use the down arrow to scroll down to the test you want to run and press the Enter key. You will either run a demo or be presented with a second window with programs to run. So, for example, selecting the 3D Demos, the following screen will be displayed

```
3D Demos

 * Vase      OpenGL ES 1.1 Translucency and reflections
   Chameleon OpenGL ES 1.1 VGP Skinning with DOT3 Per Pixel Lighting
   Flow      OpenGL ES 2.0 Cover Flow Demo
   Shaders   OpenGL ES 2.0 Mesh Torus

Press ESC to exit
```

How to Import the Matrix TUI into CCSv5

The following article will guide you through the steps to import all projects under the Matrix GUI into CCSv5: → How to Import Matrix Projects into CCSv5. (The Matrix GUI Qt application itself has a different import procedure: → How to Import Matrix GUI into CCSv5).

How to Build the Matrix TUI

Dependencies

The Matrix TUI uses two well-known libraries: ncurses and libxml2. The ncurses library is used for console display while the libxml2 library is used to parse the matrix.xml file that contains configuration information. The include files and shared libraries for both packages are included in the installed linux-devkit.

Using CLI

To build using the command line, simply change directory to the matrix_tui directory and type in make. To install into your target root file system, type in make install.

How to Customize the Matrix TUI

The configuration file for the Matrix TUI is the matrix.xml file in your target root file system /usr/share/matrix/xml directory. A sample few lines are shown below `<window> <name>ARM Benchmarks</name> <program> <choice>Dhrystone</choice> <desc>Dhrystone benchmark</desc> <path>runDhrystone</path> </program> <program> <choice>Whetstone</choice> <desc>Whetstone benchmark</desc> <note>This can take approx 10 seconds</note> <path>whetstone 10000</path> </program> <program> <choice>Linpack</choice>`

```
<desc>Linpack benchmark</desc> <note>This can take approx 10 seconds</note>
<path>linpack</path> </program> </window>
```

The <window> tag defines the start of a new window and the </window> tag defines the end of the window. To name a window, use the <name> tag. The name of the window will be displayed at the top of the display. Windows are made up of programs, or links to other windows. So a <window> can contain a <program> and </program> to define a program or a <link> and </link> to define a link to another window.

For each program, there are four tags to describe it. The <choice> tag is the left-most text displayed in for the program in the window. The <desc> tag that is a description of the program. The <note> tag that lets you display a note before running a program. And finally <path> is the path to the executable. A <link> tag is used to select another window. The name of the window must match the name in the <link> tag. An example of a <link> tag is shown below

```
<program> <choice>ARM Benchmarks</choice> <desc>A selection of ARM benchmarks
to run</desc> <link>ARM Benchmarks</link> </program>
```

And the corresponding window was shown above. Notice the name in the <link> must match the <name> in the window to be able to select sub-windows.

How to Debug the Matrix TUI

You can find out how to debug the Matrix TUI using CCSv5 in the following wiki article: → How to Run GDB on CCSv5

References

[1] http://processors.wiki.ti.com/index.php/How_to_use_a_Mouse_instead_of_the_Touchscreen_with_Matrix
[2] http://processors.wiki.ti.com/index.php/How_to_Switch_Display_From_Touchscreen_to_DVIout_with_Matrix

How to use a Mouse instead of the Touchscreen with Matrix

Introduction

A mouse can be used instead of touchscreen input with the Matrix GUI by following these steps. It is not possible to use touchscreen and mouse input simultaneously. The mouse must be connected through a USB 2.0 hub to the EVM. It must not be connected directly to the EVM.

Kernel

The kernel must be configured to include the emulated IntelliMouse Explorer PS/2 mouse support. To verify this, run the menuconfig utility.

```
cd  ti-sdk-4.0\AM37x-BSP\src\kernel\linux
make  menuconfig
```

Look under:

Device Drivers ---> Input device support ---> <*> Mouse interface

Device Drivers ---> Input device support ---> <*> Mice --->

If mouse support is not already included in the kernel, enable the above two configuration by typing 'y' and rebuild the kernel.

Compiling the Linux Kernel [1]

QWS_MOUSE_PROTO

Edit the Matrix GUI script file in the target file system and change the "export QWS_MOUSE_PROTO" line as shown:

```
cd  /etc/init.d
vi  matrix-gui
```

```
export  QWS_MOUSE_PROTO=Tslib:/dev/input/touchscreen0         --->
    export  QWS_MOUSE_PROTO=Auto:/dev/input/mice
```

Cursor Enable

Matrix normally disables the cursor on the display for all menu pages when using touchscreen input. To use a mouse, this line must be disabled in the matrix_gui source code to make the cursor visible on the display. Edit the file main.cpp in the matrix_gui source code and recompile.

// QApplication::setOverrideCursor(QCursor(Qt::BlankCursor));

References

[1] http://processors.wiki.ti.com/index.php/AM35x-OMAP35x-PSP_03.00.00.05_UserGuide#Linux_Kernel

How to Switch Display From Touchscreen to DVIout with Matrix

Introduction

A DVI display can be used instead of the LCD touchscreen with the Matrix GUI by following these steps.

Stop Matrix

First stop the matrix application by running the following command.

```
/etc/init.d/matrix-gui stop
```

A stale output from matrix will still show on the LCD even though the application is now dead.

Run commands to Switch the Display

The following set of commands will switch the display output from the LCD to the DVI port. The stale output from when matrix was killed will now be sent to the DVI port.

```
echo 0 > /sys/devices/platform/omapdss/display0/enabled
echo "" > /sys/devices/platform/omapdss/manager0/display
fbset -fb /dev/fb0 -xres 720 -yres 480
echo "dvi" > /sys/devices/platform/omapdss/manager0/display
echo 1 > /sys/devices/platform/omapdss/display2/enabled
echo 1 > /sys/class/graphics/fb0/rotate
```

Restart Matrix

```
/etc/init.d/matrix-gui start
```

Matrix will now be displayed to the DVI port. Depending on the type of display/monitor connected to the DVI port, it may necessary to adjust the setting to get the proper display. Refer to the instructions for the particular display or monitor that you are using.

How to Make a Hello World Matrix Application

How to Make a 'Hello World' Matrix Application

This article will discuss making an arbitrary Matrix application in a 'how to' tutorial like format, for details on the inner workings of Matrix, please see the → Matrix Users Guide.

Add a new button to Matrix

First we add a new button to the Matrix GUI somewhere, in this case lets put one under System Settings. Since Matrix uses html files to define the user interface, we need to edit the menu_settings.html file, which is found in /usr/share/matrix/html on the target filesystem, or the base folder of the matrix_gui sources.

Note that here we are assuming you have a NFS exported filesystem in your SDK install (replace the Xs with your version), if you have a NFS exported filesystem elsewhere you can adjust the command accordingly.

Note that depending on your filesystem's permissions you may need to use root to edit this file (as well as others in the NFS filesystem), if so you can start your editor with gksudo (i.e. gksudo gedit ~/...).

```
host $ gedit
~/ti-sdk-XXXX-evm-x.x.x.x/filesystem/usr/share/matrix/html/menu_settings.html
```

This defines what buttons the user sees in the settings menu, where the buttons are, what image they have, what label they have, and what they do, for details on the formatting please see this section of the Matrix Users Guide.

Since the existing buttons already have the basics laid out for us, we can just duplicate one of those, so copy the button contents from an existing button, to a new button lower down so that we add a new button to the second row, the existing button code should look something like the below.

```
          <object type="application/x-matrix" width="96px"
height="96px">
           <param name="iconName"
value="/usr/share/matrix/images/info-icon.png" />
           <param name="iconLabel" value=" Task Information" />
           <param name="appName" value="taskInfo" />
          </object>
```

If we copy this and paste it in a lower <td> segment (these define entries in a table in HTML, so we are making a table of buttons), we can make a duplicate button. At this point your button should have the same behavior of the original button if you were to run Matrix on a target and click into the menu, the html file should have something like the below in it (with the rest of the html above and below).

```
...
          <param name="appName" value="taskInfo" />
         </object>


     <tr>
       <td align="center" width="96px" height="96px">
          <object type="application/x-matrix" width="96px"
height="96px">
           <param name="iconName"
value="/usr/share/matrix/images/info-icon.png" />
```

```
            <param name="iconLabel" value=" Task Information" />
            <param name="appName" value="taskInfo" />
          </object>

        <td align="center" width="96px" height="96px">

...
```

From here you can change the button image, the label, or the app itself.

Modifying the New Button

The HTML used to define the button in the table of buttons is fairly self explanatory (detailed further here), in general you probably want to keep the same sizes so the heights and widths can be left at 96.

To change the icon you simply need to point to an other icon on the target file system, either another from within the provided Matrix icons, or one of your own.

```
<param name="iconName"
value="/usr/share/matrix/images/info-icon.png" />
```

The label underneath each icon is defined by the next value, and is also easily changed.

```
<param name="iconLabel" value=" Hello World!" />
```

The final value points to the particular application you want Matrix to run upon the user clicking on the icon, this is the name of the application's start up script which will be discussed in the next section.

```
<param name="appName" value="helloWorld" />
```

Preparing an App to Run from Matrix

Matrix does not execute applications directly, rather it relies on executing scripts that exist in /usr/bin of the target filesystem (the scripts used by matrix are in the /bin directory of the matrix_gui sources). This being said, the appName value mentioned in the prior section must match the script name in /usr/bin of your target filesystem. For a simple hello world example, make a text file in /usr/bin of the target called helloWorld.

Note that depending on your filesystem's permissions you may need to use root to edit this file (as well as others in the NFS filesystem), if so you can start your editor with gksudo (i.e. gksudo gedit ~/...).

```
host $ gedit ~/ti-sdk-XXXX-evm-x.x.x.x/filesystem/usr/bin/helloWorld
```

Now add the below to the helloWorld file using your favorite editor.

```
echo "Hello World!"
```

Now to ensure we can execute this script we will need to give it execute permissions, which can be done on the target.

```
target $ cd /usr/bin
target $ chmod +x helloWorld
```

After this, if you start up Matrix on the target and navigate through the GUI to the button you created, you should see the Hello World! output when you press the button.

```
target $ /etc/init.d/matrix-gui start
```

Congratulations, you now have a very simple Matrix application, and can run other basic command line commands from the Matrix GUI by modifying the helloWorld script, or creating a new script.

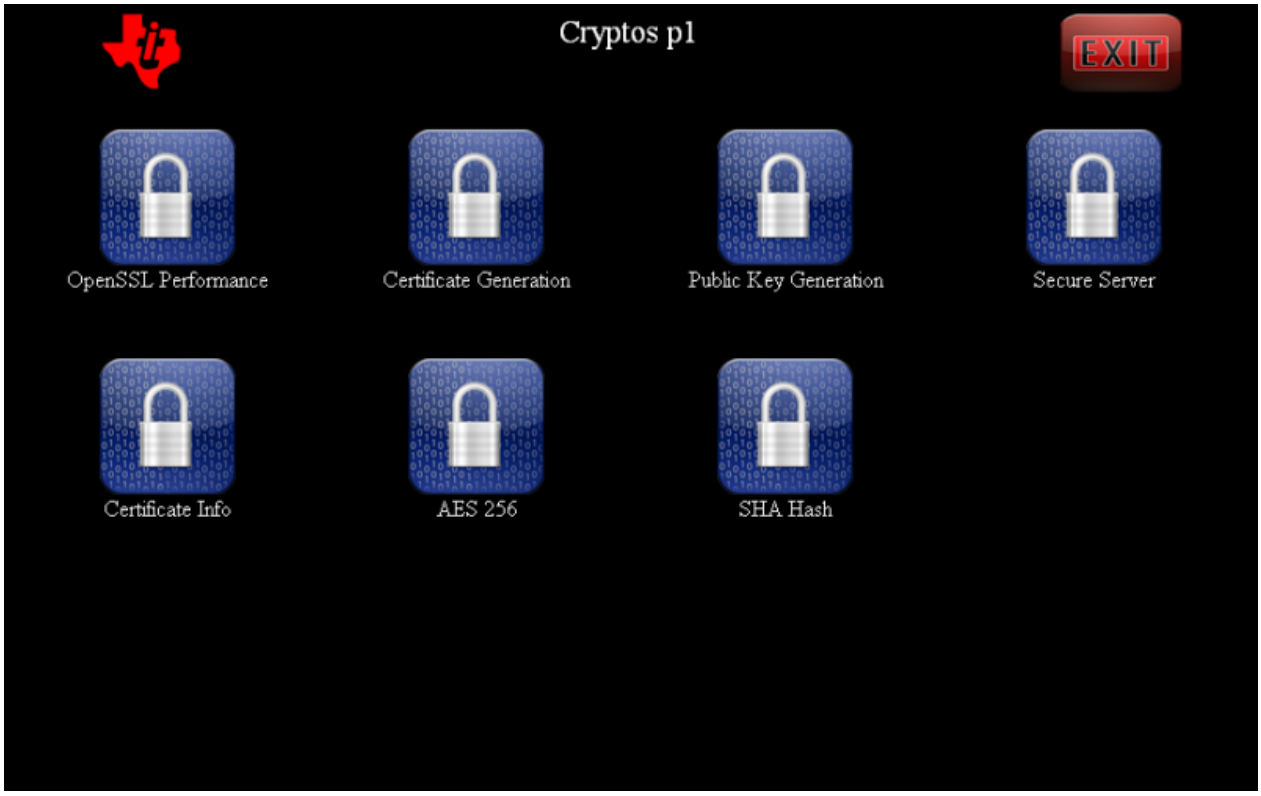
Cryptography Users Guide

Overview

This article provides a description of the example applications under the cryptography page of the Matrix application that comes with the Sitara SDK. This page is labled "Cryptos" in the top-level Matrix GUI.

Cryptography Examples

All of the examples under the Cryptos page use the OpenSSL command line application to perform cryptographic functions. A comprehensive list of cryptographic functionality using OpenSSL is beyond the scope of the out-of-box experience intended with the Matrix GUI. However, the examples present a nice variety of cryptographic functions that are available with OpenSSL on the Sitara platform.



OpenSSL Performance

This example executes the OpenSSL built-in speed test for a variety of cryptographic algorithms. The results of the test are displayed on the screen and also written to the file OpenSSLspeedResults.txt in the top level directory of the target filesystem.

Certificate Generation

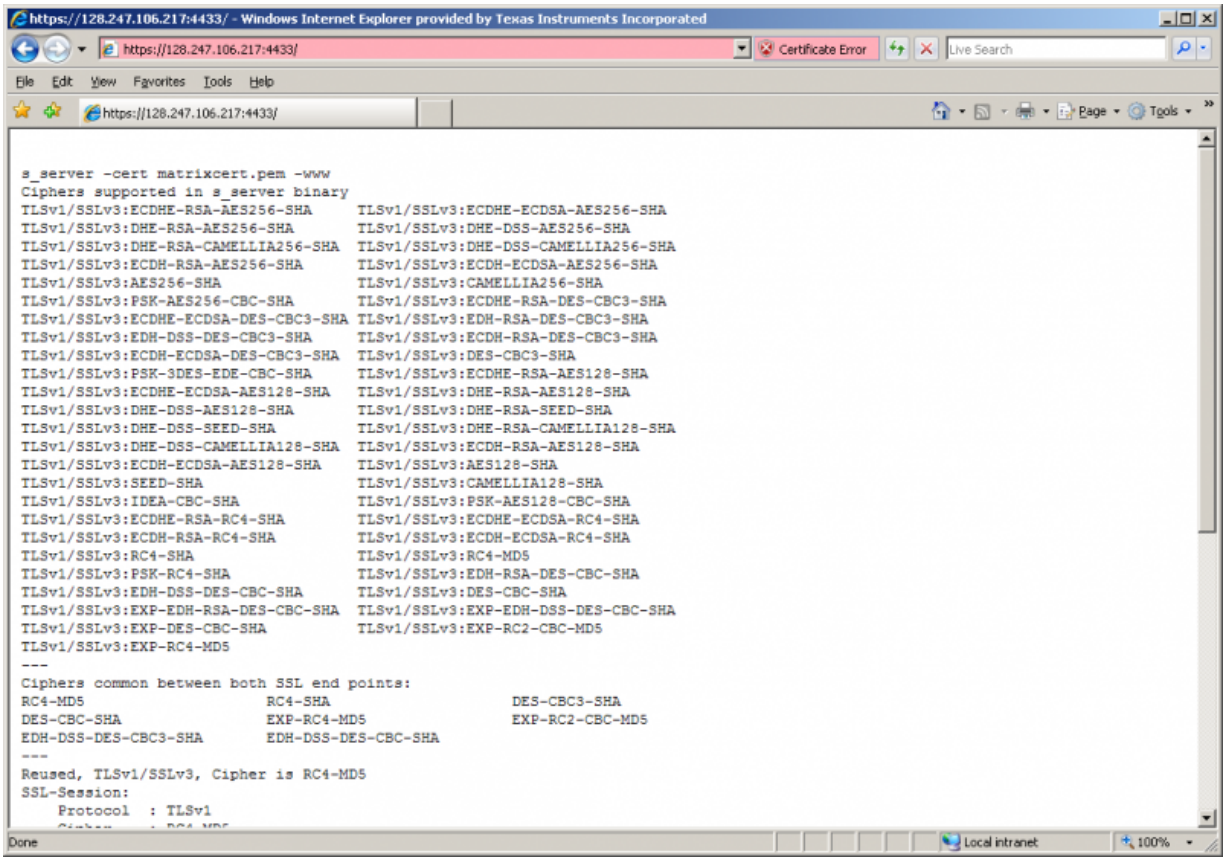
This example generates a web page certificate for use with a secure web server. The certificate is held in the file matrixcert.pem. This file will appear in the top level directory of the target file system. If the certificate already exists, then the example will fail and prompt the user to delete the existing certificate (matrixcert.pem) before generating a new one.

Public Key Generation

This example generates a public key based on the certificate (matrixcert.pem) generated in the previous example. If the certificate does not exist the example will fail and pronpt the user to first generate the certificate before trying to generate the public key. The public key will be saved to a file pubkey.pem in the top level directory of the target file system.

Secure Server

Once the web certificate (matrixcert.pem) has been generated, the secure web server can be started on the target board. Pointing a modern web browser at the target should generate a warning that the certificate is self-signed. This means that the certificate has not been verified by a trusted third party such as Verisign. Depending on the browser, you can view details of the certificate. In the example below the target board has an IP address of 128.247.106.217. When Internet Explorer is pointed to the URL https://128.247.106.247:4433, it first warns the user that there is a problem with the website's security certificate. Make sure that you use https:// and the port number :4433 with the IP address in the URL of the browser. Clicking the link to continue anyway provides the page below. And clicking the "Certificate Error" button at the top of the page will provide details of the certificate.



Certificate Info

This example simply prints out details of the generated certificate (matrixcert.pem).

AES 256

This example will perform an encrypt/decrypt cycle on a 10M file of random data with the AES 256 algorithm. The 10M file called rnddata will be generated if it doesn't already exist. The result of the decryption is compared to the original file and the results are displayed to the screen.

SHA Hash

This example will perform a SHA1 hash function on the 10M file of random data (rnddata). If the file doesn't exist, it is generated. The result of the hash is displayed to the screen.

Using Cryptographic Hardware Accelerators

The AM37x General Purpose (GP) device allows access to built in cryptographic accelerators that are normally only available in High Security (HS) versions of the device. In order to be useful, a special driver is available which abstracts the access to these accelerators through the Open Cryptographic Framework for Linux (OCF-Linux).

OCF-Linux is itself a special device driver which provides a general interface for higher level applications such as OpenSSL to access hardware accelerators.

The Linux kernel used with the AM37x must be built with OCF-Linux support in order to use the OCF-Linux driver.

The pre-built kernel which comes with the AM37x SDK has OCF-Linux support built in. The filesystem which comes with the AM37x SDK includes the OCF-Linux device driver and the TI driver which directly accesses the hardware accelerators.

From the target boards perspective the drivers are located in the following directories:

```
/lib/modules/2.6.32/kernel/crypto/ocf/cryptodev.ko
/lib/modules/2.6.32/crypto/ocf/ocf_omap3_cryptok.ko
```

To use the drivers they must first be installed. Use the insmod command to install the drivers. The TI crypto driver allows a parameter to be passed in to indicate if DMA should be used. The following log shows the commands used to install the modules and query the system for the state of all system modules.

```
root@am37x-evm:~# lsmod
Module Size Used by
bufferclass_ti 4814 0
omaplfb 8210 0
pvrsrvkm 128866 2 bufferclass_ti,omaplfb
root@am37x-evm:~#
root@am37x-evm:~# insmod
/lib/modules/2.6.32/kernel/crypto/ocf/cryptodev.ko
cryptodev: module license 'BSD' taints kernel.
Disabling lock debugging due to kernel taint
root@am37x-evm:~#
root@am37x-evm:~# insmod
/lib/modules/2.6.32/crypto/ocf/ocf_omap3_cryptok.ko
ocf_omap3_crypto_dma=1
root@am37x-evm:~#
root@am37x-evm:~# lsmod
```

```
Module Size Used by
ocf_omap3_cryptok 17723 0
cryptodev 10560 0
bufferclass_ti 4814 0
omaplfb 8210 0
pvrsrvkm 128866 2 bufferclass_ti,omaplfb
root@am37x-evm:~#
```

After the modules are installed, OpenSSL commands may be executed which take advantage of the hardware accelerators through the OCF-Linux driver. The following example demonstrates the OpenSSL built-in speed test to demonstrate performance. The addition of the parameter **-engine cryptodev** tells OpenSSL to use the OCF-Linux driver if it exists.

```
root@am37x-evm:~# openssl speed -evp aes-128-cbc -engine cryptodev
engine "cryptodev" set.
Doing aes-128-cbc for 3s on 16 size blocks: 108107 aes-128-cbc's in
0.16s
Doing aes-128-cbc for 3s on 64 size blocks: 103730 aes-128-cbc's in
0.20s
Doing aes-128-cbc for 3s on 256 size blocks: 15181 aes-128-cbc's in
0.03s
Doing aes-128-cbc for 3s on 1024 size blocks: 15879 aes-128-cbc's in
0.03s
Doing aes-128-cbc for 3s on 8192 size blocks: 4879 aes-128-cbc's in
0.02s
OpenSSL 1.0.0b 16 Nov 2010
built on: Thu Jan 20 10:23:44 CST 2011
options:bn(64,32) rc4(ptr,int) des(idx,risc1,2,long) aes(partial)
idea(int) blowfish(idx)
compiler: arm-none-linux-gnueabi-gcc -march=armv7-a -mtune=cortex-a8
-mfpu=neon -mfloat-abi=softfp -mthumb-interwork -mno-thumb -fPS
The 'numbers' are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes
aes-128-cbc 10810.70k 33193.60k 129544.53k 542003.20k 1998438.40k
root@am37x-evm:~#
root@am37x-evm:~#
root@am37x-evm:~#
```

Using the Linux time -v function gives more information about CPU usage during the test.

```
root@am37x-evm:~# time -v openssl speed -evp aes-128-cbc -engine
cryptodev
engine "cryptodev" set.
Doing aes-128-cbc for 3s on 16 size blocks: 108799 aes-128-cbc's in
0.17s
Doing aes-128-cbc for 3s on 64 size blocks: 102699 aes-128-cbc's in
0.18s
Doing aes-128-cbc for 3s on 256 size blocks: 16166 aes-128-cbc's in
0.03s
```

```
Doing aes-128-cbc for 3s on 1024 size blocks: 15080 aes-128-cbc's in
0.03s
Doing aes-128-cbc for 3s on 8192 size blocks: 4838 aes-128-cbc's in
0.03s
OpenSSL 1.0.0b 16 Nov 2010
built on: Thu Jan 20 10:23:44 CST 2011
options:bn(64,32) rc4(ptr,int) des(idx,risc1,2,long) aes(partial)
idea(int) blowfish(idx)
compiler: arm-none-linux-gnueabi-gcc -march=armv7-a -mtune=cortex-a8
-mfpu=neon -mfloat-abi=softfp -mthumb-interwork -mno-thumb -fPS
The 'numbers' are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes
aes-128-cbc 10239.91k 36515.20k 137949.87k 514730.67k 1321096.53k
Command being timed: "openssl speed -evp aes-128-cbc -engine cryptodev"
User time (seconds): 0.46
System time (seconds): 5.89
Percent of CPU this job got: 42%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0m 15.06s
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 7104
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 479
Voluntary context switches: 36143
Involuntary context switches: 211570
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
```

When the TI driver is removed, OpenSSL reverts to the software implementation of the crypto algorithm. The performance using the software only implementation can be compared to the previous test.

```
root@am37x-evm:~# lsmod
Module Size Used by
ocf_omap3_cryptok 17723 0
cryptodev 10560 0
bufferclass_ti 4814 0
omaplfb 8210 0
pvrsrvkm 128866 2 bufferclass_ti,omaplfb
root@am37x-evm:~# rmmod ocf_omap3_cryptok
```

```
root@am37x-evm:~# time -v openssl speed -evp aes-128-cbc
Doing aes-128-cbc for 3s on 16 size blocks: 697674 aes-128-cbc's in
2.99s
Doing aes-128-cbc for 3s on 64 size blocks: 187556 aes-128-cbc's in
3.00s
Doing aes-128-cbc for 3s on 256 size blocks: 47922 aes-128-cbc's in
3.00s
Doing aes-128-cbc for 3s on 1024 size blocks: 12049 aes-128-cbc's in
3.00s
Doing aes-128-cbc for 3s on 8192 size blocks: 1509 aes-128-cbc's in
3.00s
OpenSSL 1.0.0b 16 Nov 2010
built on: Thu Jan 20 10:23:44 CST 2011
options:bn(64,32) rc4(ptr,int) des(idx,risc1,2,long) aes(partial)
idea(int) blowfish(idx)
compiler: arm-none-linux-gnueabi-gcc -march=armv7-a -mtune=cortex-a8
-mfpu=neon -mfloat-abi=softfp -mthumb-interwork -mno-thumb -fPS
The 'numbers' are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes
aes-128-cbc 3733.37k 4001.19k 4089.34k 4112.73k 4120.58k
Command being timed: "openssl speed -evp aes-128-cbc"
User time (seconds): 15.03
System time (seconds): 0.00
Percent of CPU this job got: 99%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0m 15.07s
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 7216
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 1
Minor (reclaiming a frame) page faults: 484
Voluntary context switches: 13
Involuntary context switches: 35
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
```

Pin Mux Utility for ARM MPU Processors

Introduction

The Pin Mux Utility for Sitara ARM Cortex-A8 based microprocessors is a Windows-based software tool for configuring pin

multiplexing settings and I/O cell characteristics for the AM389x / C6A816x, AM37x / DM37x, AM35x and OMAP35x devices.

Sitara processors provide pad configuration programmability to control the routing of internal signals to the external balls of the device. Pad configuration also allows I/O cell characteristics to be controlled. These include enabling of internal pull-up / pull-down resistors and specifying I/O cell behavior independently in active and standby modes of operation.

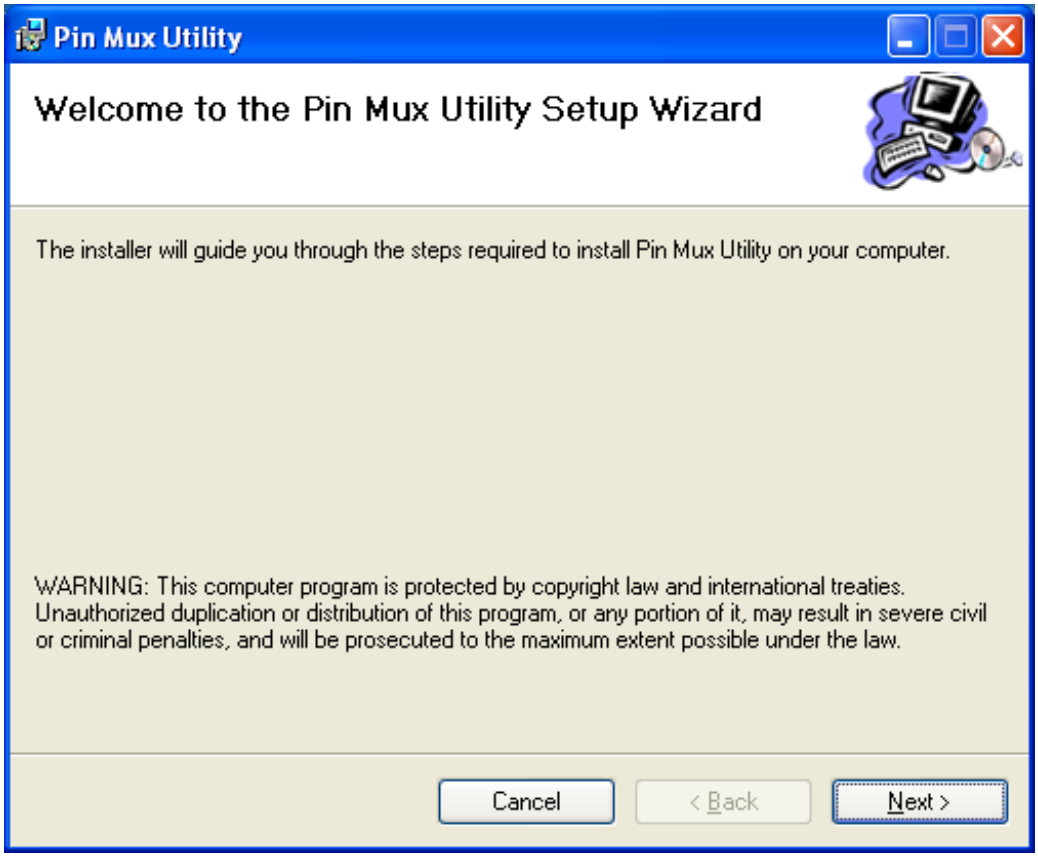
The Pin Mux Utility provides a graphical user interface for selecting the peripheral interfaces that will be used in the system design and for resolving pin multiplexing conflicts. Once all conflicts have been resolved, the results are output as two C header files. The first header file is device-specific and it specifies all the pad configuration registers that can be programmed on the device. The second header file is board-specific and it specifies all the pin multiplex and pad configuration settings that were generated for your specific system design. The format of the output header files is compatible with the format used by the UBOOT initialization software.

Software Installation

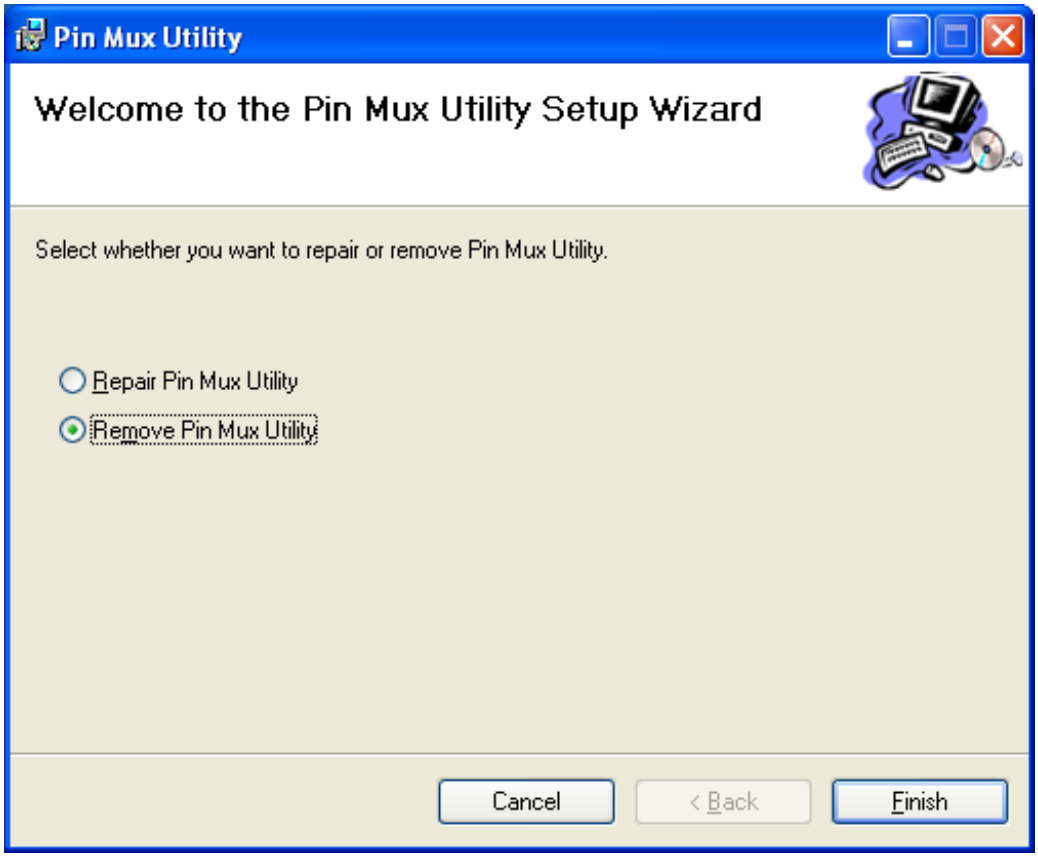
The most recent version of the Pin Mux Utility software is available here [1]. It is also installed automatically as part of the AM37x Software Development Kit [2], but the stand-alone version is recommended to get the most recent updates.

If you are running Linux as a virtual machine on a Windows O/S, the Windows Installer and Pin Mux Utility program can be run on the Linux host PC under the Windows O/S. The generated source files can then be transferred to the Linux virtual machine via shared folders or another available method.

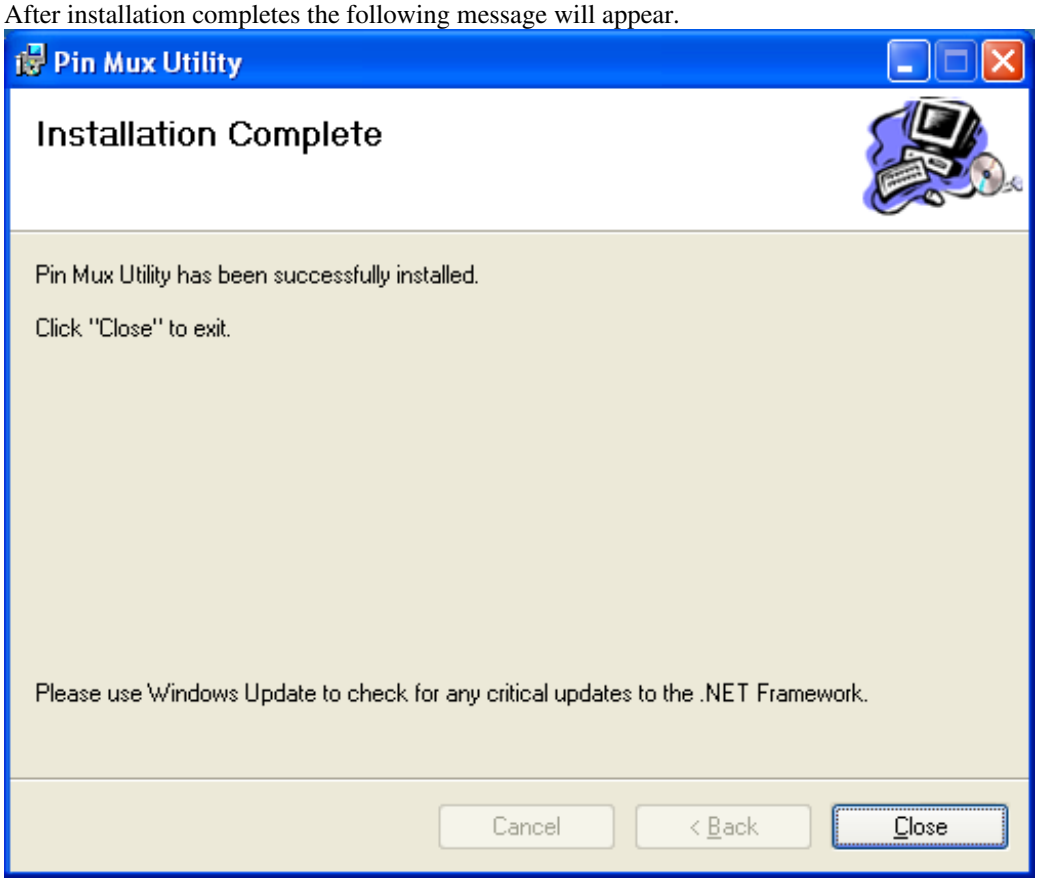
If your Linux O/S is not a virtual machine, the Windows installer and Pin Mux Utility program must be run on a separate PC running the Windows O/S. To run the Windows Installer, run Setup.exe. The installer should start running as shown below.



If a previous version of Pin Mux Utility was installed, the following message will appear. Select "Remove Pin Mux Utility" and click Finish to uninstall the previous version. Then, run Setup.exe again to install the new version.



After installation completes the following message will appear.



Software User's Guide

Program Startup

To see the current release notes, click on this in the Windows Start menu:

All Programs > Pin Mux Utility > Release_Notes.txt

To start the program, double-click the Pin Mux Utility icon on the desktop, or click

All Programs > Pin Mux Utility > Pin Mux Utility

in the Windows Start menu.

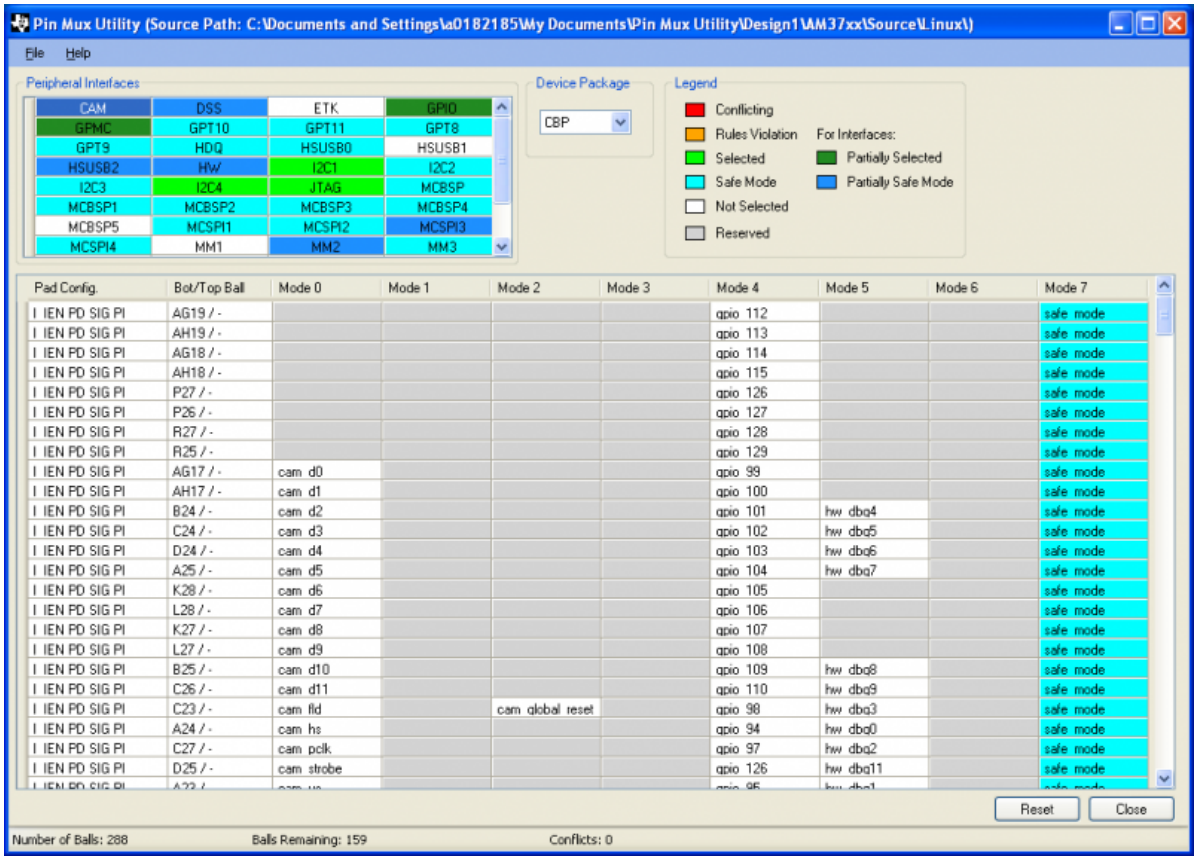
Pin Mux Utility

A dialog box will appear indicating which devices are supported by this installation of the Pin Mux Utility as shown below. Select your device and click OK.

Main Window

The main window opens with device data populated for the selected device type. All pin mux selections and pad configurations are in the power-up-reset state.

- The title bar indicates the path where output source files will be stored.
- The Peripheral Interfaces data grid shows the current mux selection state for the group of balls that make up each peripheral interface.
- The Device Package selector controls which package type is used to populates the "Bot/Top Ball" ball location column.
- The Legend describes the color scheme that is used to give a visual indication of the state of each peripheral interface and the state of each mux item in the main data grid.
- The main data grid represents one device ball in each row and contains columns to display the current pad configuration and ball location(s).
- Columns labeled Mode 0 to Mode 7 represent the pin mux state at each ball. The state may be changed by double clicking on a cell.
- The status bar indicates the total number of device balls, number of balls remaining and the number of balls with conflicting mux settings.
- The Reset button is used to set all mux selections and pad configurations to the power-on-reset state.
- The close button exits the program.



Menu Items

File Menu

- File > Save > Design

Saves current pin mux selections and pad configurations to a data file that can be reloaded at any time or on a subsequent run of the program. This file, by default, will be stored at:
<My Documents Folder>\Pin Mux Utility\Design1\<device_name>\PinMuxDesignState_<device_name>.dat
Where: <device_name> is AM37xx, AM35xx or OMAP35xx

- File > Open > Design

Opens data file stored previously with the File > Save > Design menu item. Must open a file for the same device type that was selected at program startup. This will populate all mux selections and pad configuration parameters from the data file.

- File > Save > Source > Linux

Saves the current pin mux selections and pad configurations to C header files.

- File > Exit

Exits the program. You will be prompted to "Save changes to mux selections and/or pad configurations?". If Yes is clicked, this will open a file save dialog to save the design data file as when the File > Save > Design menu command is used.

Help Menu

- Help > ReadMe

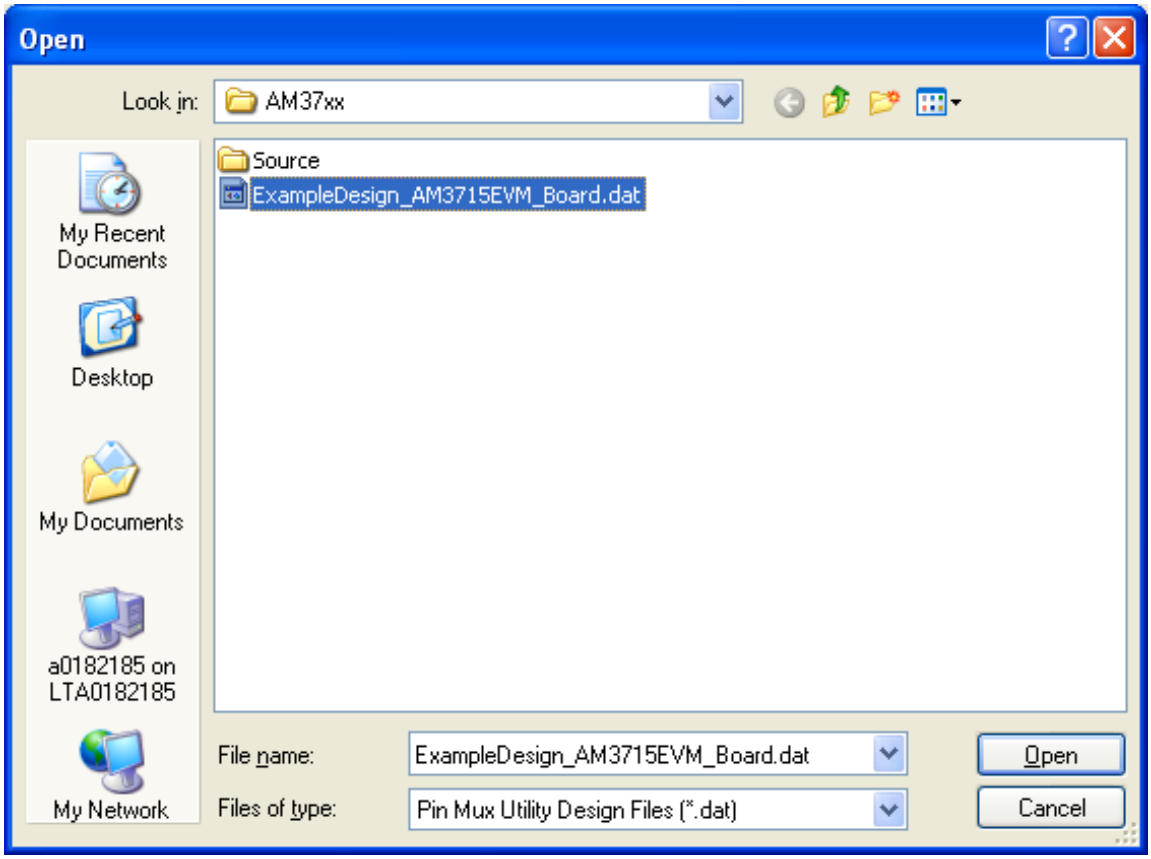
Displays a window describing features of the program.

- Help > About

Displays the program version.

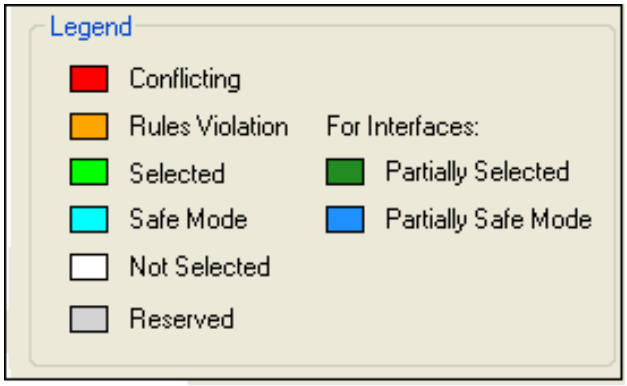
Opening an Example Design Data File

For each device type, a data file containing the pin mux settings and pad configurations for the corresponding EVM board is included. Use File > Open > Design , click on the example file and Open. If your design is similar to the TI EVM board for your device, this can be used as a good starting point.



Legend

The legend below shows the visual indicators that are used to display the state of the pin mux settings at the peripheral interface and individual ball levels. The tables below describe how these states are to be interpreted.



Legend - State of Peripheral Interfaces

Peripheral Interface State	Description
Conflicting	An error condition where at least one ball has more than one signal selected.
Rules Violation	An error condition where at least one set of balls in the interface are selecting mutually-exclusive signals. (not implemented)
Selected	All balls of the interface have a signal selected free-and-clear.
Partially Selected	A subset of the balls of the interface have their signals selected. Remaining balls are set to safe mode or other interfaces.
Safe Mode	All balls of the interface are set to safe mode.
Partially Safe Mode	A subset of the balls of the interface are set to safe mode. Remaining balls have a signal of another interface selected
Not Selected	All balls of the interface have a signal of another interface selected.

Legend - State of Individual Mux Items (Cells in a row)

Mux Item State	Description
Conflicting	An error condition where more than one signal is selected on the same ball.
Rules Violation	An error condition where a mutually exclusive signal selection exists. (not implemented)
Selected	Signal is selected free-and-clear. If all other cells in the row are reserved, the cell cannot be deselected.
Safe Mode	Ball is to be configured as an input with no functional interface mapped to it. Deselect all other cells in the row to select safe mode.
Not Selected	Signal is not selected.
Reserved	This mux mode cannot be selected. Cell cannot be changed.

Main Data Grid

The main data grid provides a view of all multiplexed signals that can be brought out to the pads of the die.
The "Bot / Top Ball" column indicates the ball location(s) that are connected to the die pads. If this column indicates "- / -", the die pad does not connect to a ball of the device (for the selected device package).
Each row represents one pad on the die. Each column labeled Mode 0 to Mode 7 represents one mux mode.
Each row is independent. To scroll to the first signal of a peripheral interface, double click the corresponding cell in the peripheral interfaces grid. To change the state of a cell, double-click the cell.
Changes occur according to the following table.

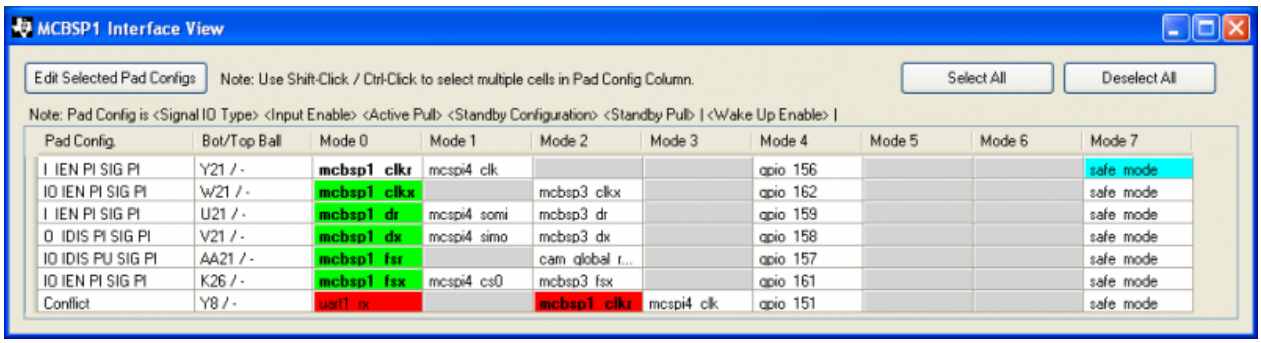
Cell State Change when Double-Clicking a Cell

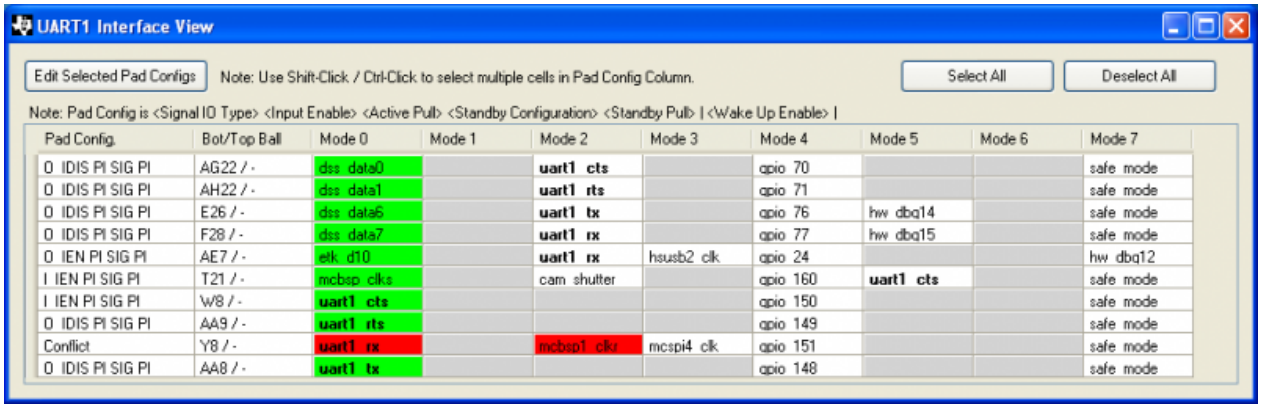
Original Cell State	New Cell State
Conflicting	**Conflicting** if at least one other cell in row is still conflicting. **Selected** if no other cells in row are conflicting
Selected	**Not Selected** if safe mode is not reserved. Safe Mode cell in row will become active. **Selected** if safe mode is reserved. This is the only cell in the row that can be selected - it cannot be deselected.
Safe Mode	**Safe Mode** - Nothing happens when clicking a safe mode cell. Select any other cell in the row to deselect the safe mode cell.
Not Selected	**Selected** if no other cells in row are already selected or conflicting **Conflicting** if any other cell in row is already selected or conflicting **NOTE**: A safe mode cell changes from **Not Selected** to **Safe Mode** if all other cells in the row are set to **Not Selected**
Reserved	**Reserved** - Cannot be changed

Interface Views

The interface view displays only the balls that make up a peripheral interface. To open an Interface View, L-Click once on a cell in the Peripheral Interfaces data grid in the main window to select the interface, then R-Click and select View Pins. Any number of Interface View windows may be opened at the same time. The cells that belong to the peripheral interface are shown in bold text. Double-click a cell in an Interface View to change its state as with the main data grid view. Select All and Deselect All buttons are available to select/deselect all cells that are members of the interface. Select one or more balls in the Pad Config column using Ctrl-Click or Shift-Click, then click the Edit Selected Pad Configs button to open the Pad Config Editor. Below, is an example of MCBSP1 and UART1 interface views. A conflict exists between the uart1_rx and mcbsp1_clkr signals. This example also illustrates a rules violation. Signal mcbsp1_clkr is muxed to multiple balls (Y21 for Mux Mode 0 and Y8 for Mux Mode 2). In the case of multi-muxed signals, the same mux mode that is used for the other signals of

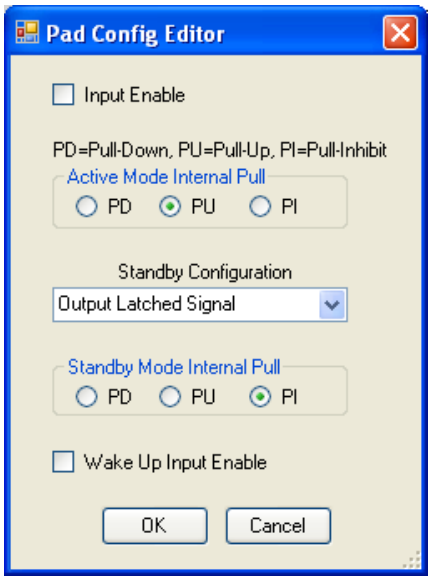
the interface must be used or there is a "rules violation". Detection of the rules violation is not done by the program in the current release, so care must be taken. In the example below, mcbsp1_clkr for Mux Mode 0 must be used.





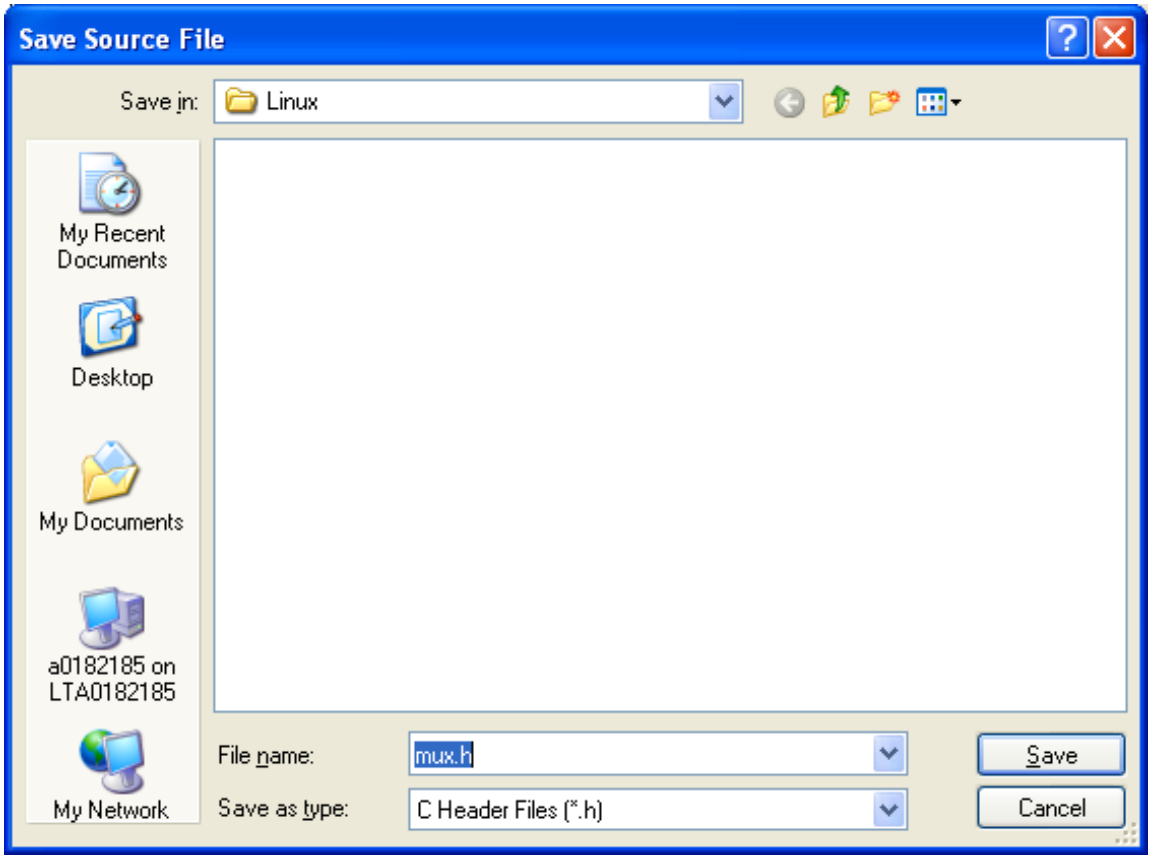
Procedure for Configuring a Peripheral Interface

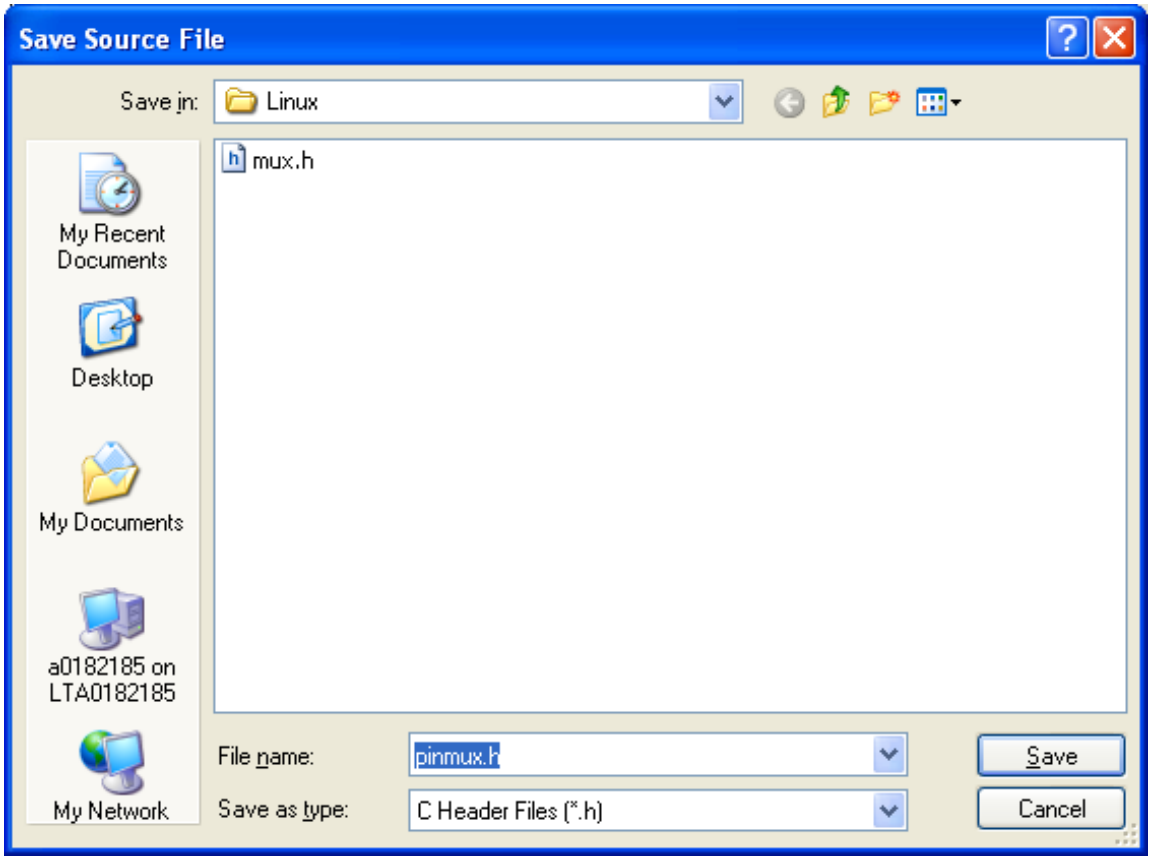
- Usually begin by clicking Select All
- Double-click any members of the interface that will not be used to deselect them
- Shift-Click in Pad Config column to select all balls
- Click Edit Selected Pad Configs and set the Active Configuration of all balls to PI (Pull-up inhibited - most common)
- Ctrl-Click cells in Pad Config column that need pull-up (or pull-down) enabled
- Click Edit Selected Pad Configs and set selected balls to PU (or PD) as required



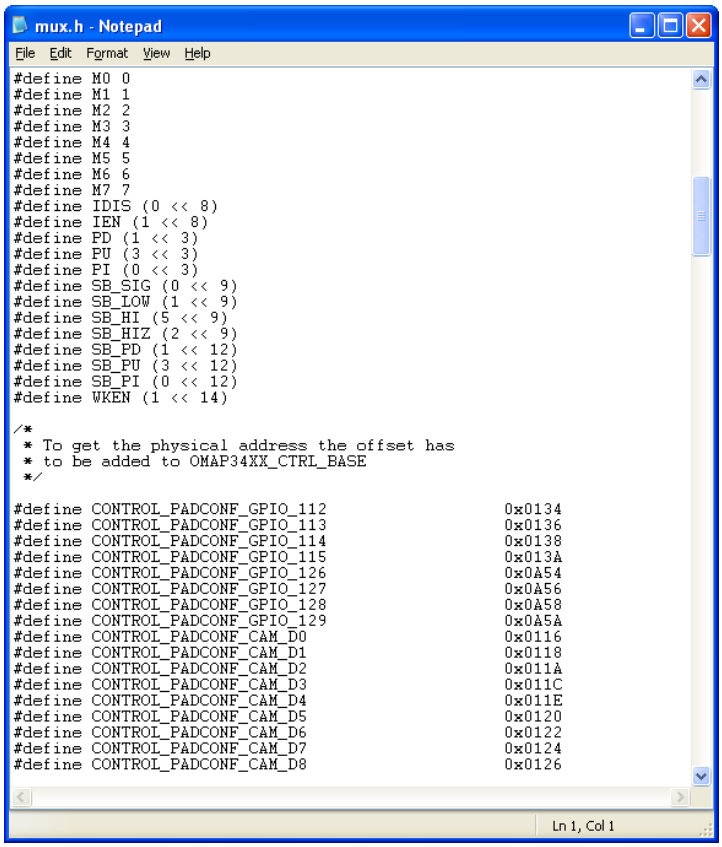
Saving Results to Source Files

Click the "File > Save > Source > Linux" menu item to save results as Linux header files for UBOOT. File save dialogs will open to speciy the file names for a device-dependent header file (default name: mux.h) and a board-dependent header file (default name: pinmux.h).





Examples of the device-dependent and board-dependent header files are shown below.



Using the Generated Header Files

The device-dependent and board-dependent header files are standard C code. Macros are used for compactness and readability. A C function in a higher level source file can make a single macro call (example: MUX_EVM()) to cause all pin configuration register writes to be performed.

The generated pinmux settings may be used to customize the U-Boot source code. Before rebuilding U-BOOT for your system, the following steps are neccessary:

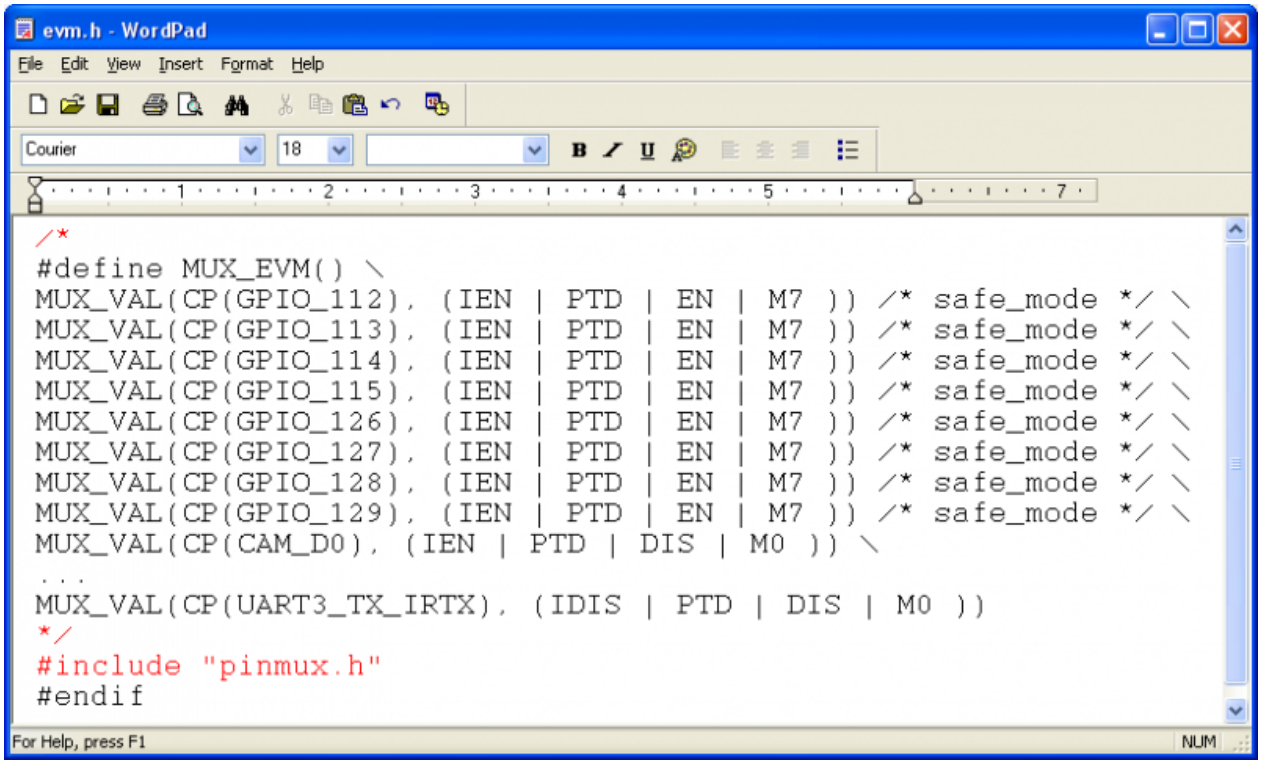
1) Replace the mux.h header file with the mux.h output file from Pin Mux Utility

2) Copy the pinmux.h output file from Pin Mux Utility into the directory containing the evm.h file.

3) Modify the original evm.h file, commenting out or deleting the original section of code that makes the pin mux programming macro calls.

4) Replace this code with #include "pinmux.h"

The following shows modificatios to the original evm.h file to include pinmux settings from a separate header file.



It is planned in future PSP versions to isolate the pin mux macro calls to a separate source file named "pinmux.h". If this is changed in future PSP versions then steps 3 and 4 above will not be neccessary, and it will just be a matter of replacing the old pinmux.h file with the new pinmux.h file.

Locations for U-Boot Linux Source Files

Device Type	Device-Dependent Header File	Board-Dependent Header File
OMAP35xx	include/asm-arm/arch-omap3/mux.h	board/ti/evm/evm.h board/ti/evm/pinmux.h (planned)
AM37xx	include/asm-arm/arch-omap3/mux.h	board/ti/evm/evm.h board/ti/evm/pinmux.h (planned)
AM35xx	include/asm-arm/arch-omap3/mux.h	board/ti/am3517evm/am3517evm.h board/am3517evm/pinmux.h (planned)

The original mux.h file includes register name defines for OMAP35xx, AM35xx and AM37xx. By setting the configuration for the make utility, U-BOOT can be built for any of these platforms. However, the Pin Mux Utility generated mux.h file will only contain register name defines for the device that was selected when Pin Mux Utility was run. So, U-BOOT can be rebuilt for that selected device only.

See also: Pinmux Utilities for Davinci Processors

References

[1] http://focus.ti.com/docs/toolsw/folders/print/pinmuxtool.html
[2] http://focus.ti.com/docs/toolsw/folders/print/linuxsdk-am37x.html

Pin Setup Tool for AM18xx ARM Microprocessors

Introduction

The Pin Setup Tool for AM18xx ARM Microprocessors is a Windows-based software tool for configuring pin multiplexing settings for the AM1802, AM1806, AM1808 and AM1810 devices.

Pin Setup provides a graphical user interface for selecting the peripherals that will be used in the system design and for resolving pin multiplexing conflicts. Results are saved as a C header file that includes a list of all selected signals and settings for the pin mux registers.

Software Installation

The Pin Setup program is a Windows application. It can be obtained from:

1. Download ZIP archive containing the executable and associated data from here Pin Setup for AM18xx [1]. This is most frequently updated.

To install, simply unzip this archive to any directory on a PC running the Windows OS. For this document, we have unzipped this

to a directory named C:\PinSetup.

OR

2. If the above link does not work, you can obtain the same ZIP archive on www.ti.com [2] on any of the AM18xx product pages in the Application Notes

section. Look for "AM18xx Pin Muxing Utility".

OR

3. The program is installed on your Ubuntu Linux host when an AM18x SDK is installed. Copy the entire directory structure from your Ubuntu Linux host

to a PC running the Windows O/S (which could be the same PC if you are running a virtual Linux machine under the Windows O/S). You would need

to transfer the directory structure by copying it under a virtual machine shared folder or by using a Samba server.

Copy the entire directory structure at the location below to any directory on the Windows side (suggest: C:\PinSetup*.*).

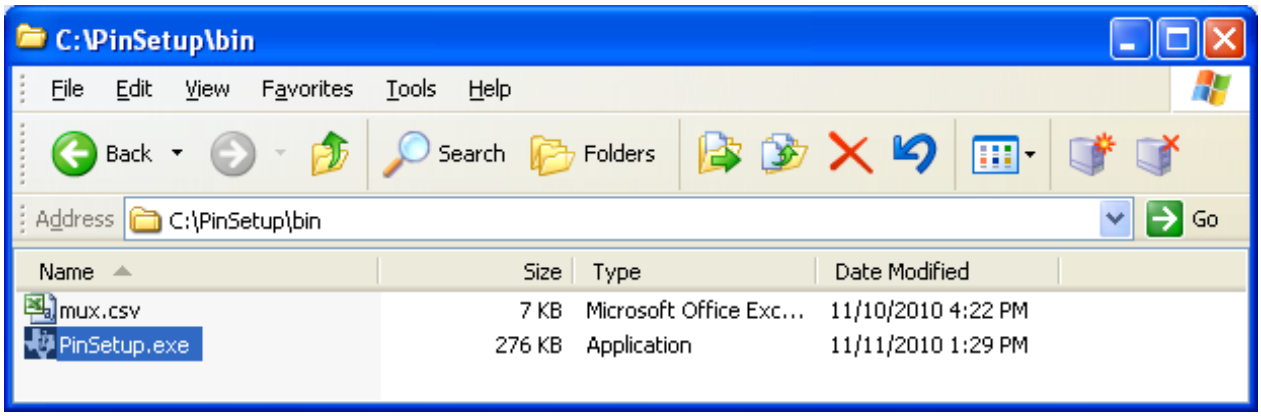
/home/user/ti-sdk-am180x-evm-4.0.1.0/host-tools/pinmux_utils/windows

OR

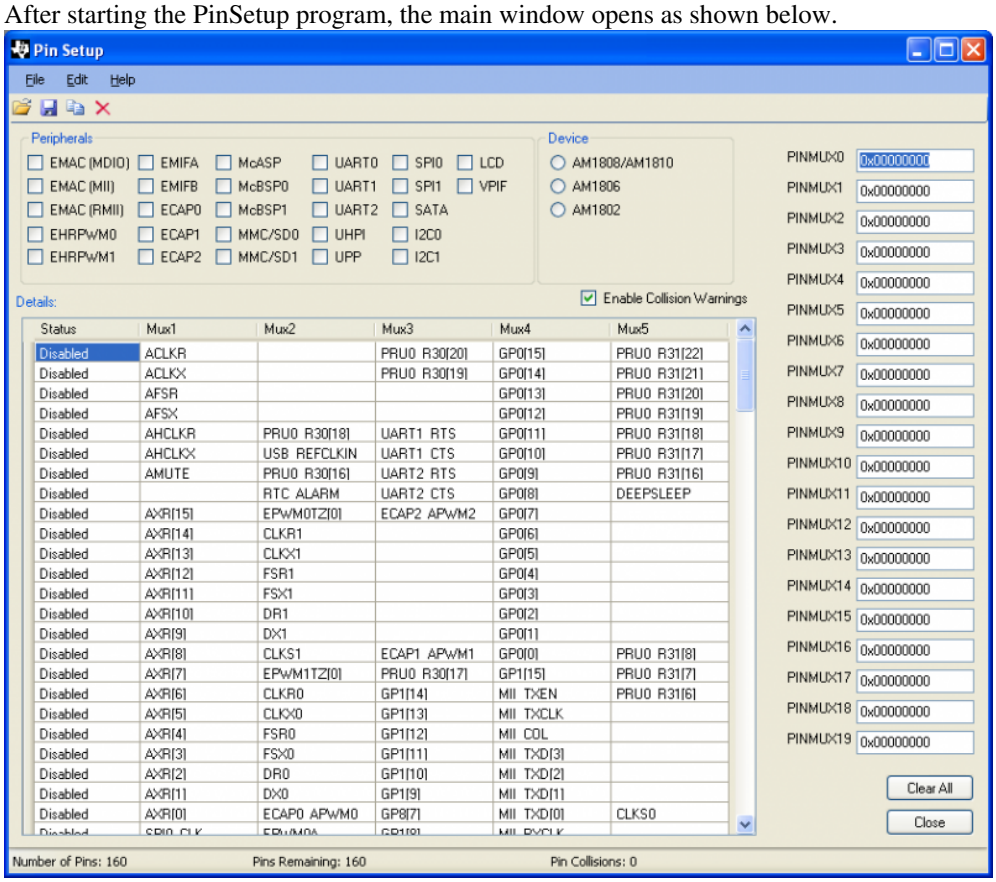
/home/user/ti-sdk-am181x-evm-4.0.1.0/host-tools/pinmux_utils/windows

Starting the PinSetup Program

To start running Pin Setup, open the bin subdirectory and run the program named PinSetup.exe.



After starting the PinSetup program, the main window opens as shown below.



Using the PinSetup Program

Main Window Features

Device Selecter Radio Buttons

The device selecter initially shows nothing selected (it defaults to AM1808 / AM1810). AM1808 and AM1810 have the same pin muxing options and represent the superset device which uses all peripherals in the peripheral check box area. If AM1806 or AM1802 is selected, the peripheral check boxes that do not apply will be greyed out. Also, the signals in the details grid that are associated with the unavailable peripherals will have a grey background and will not be selectable. If there are unsaved pin mux selections, they can be saved using the File - Save - Pin Selections menu item before clicking a Device Selecter Radio Button.

Registers Display

The registers display is specific to the pin mux architecture of the AM18xx (and similar) devices. There are 20 pin mux registers. Each register is 32-bits wide and programs the signal selection for 8 device balls. Each device ball in the details grid has up to 5 different multiplexed signals and controls one nibble of the registers display. The signal selection is programmed with a 4-bit code as shown in the table below.

Pin Mux Selection	Mux1	Mux2	Mux3	Mux4	Mux5
Binary Code	b0001	b0010	b0100	b1000	b0000

Clear All Button

The Clear All Button clears all pin mux selections that have been made. Use File - Save - Pin Selections before doing this to save work, if neccessary. All pin mux registers will be set to zero, and all device balls in the details grid will indicate disabled.

Close Button

The Close Button exits the program. This can also be done using the File - Exit menu item, or by double-clicking the title bar icon, or by clicking the title bar X icon. Unsaved work must be saved first using the File - Save - Pin Selections menu item.

Details Grid

Each row of the details grid represents a programmable device ball. Device balls that only have a single signal definition are not shown in the tool. Each device ball has up to 5 selectable signal definitions shown in the Mux1 - Mux5 columns. The status column indicates "Enabled" if a signal is selected, "Disabled" if no signal is selected or "Conflict" if a previously selected signal was unselected due to a conflict (conflicts are shown with a red background). The details grid view can be scrolled up and down using the vertical scrollbar.

Enable Collision Warnings

If Enable Collision Warnings is checked, a collision warning dialog box will open each time a signal selection is attempted and another signal in the same row of the details grid is already selected. The option is given to either not select the new signal or to unselect the existing signal and select the new signal. In the later case, the newly selected signal changes to green (selected) and the previously selected signal changes to red (conflict) and the status column indicates "Conflict".

If Enable Collision Warnings is not checked, the collision warning dialog box does not appear. When the program attempts to select new signals, the selection will only occur if there is no conflict.

Changing Pin Mux Selections

Pin Mux selections and unselections can be made in several ways.

Using Peripheral Check Boxes

The peripheral check boxes are used to select or unselect a group of signals that comprise a peripheral interface.

When a peripheral is checked, the program attempts to select all member signals of the interface.

When a peripheral is unchecked, the program unselects all member signals of the interface.

Double-Clicking Individual Cells

An individual signal in the details grid can be selected or unselected by double-clicking its cell.

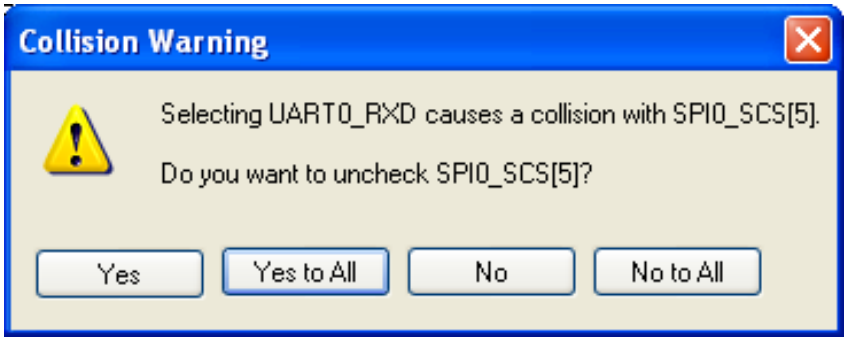
Using the Context Menu

The details grid contains a context menu with the menu items shown below. Click on a cell first, to select it, before using the context menu.

Context Menu Item	Function
Select Pin	Selects the signal
Unselect Pin	Unselects the signal
Clear Collision	Removes collision status and red background.
Find...	Open signal finder dialog box
Clear All	Clears all pin mux selections

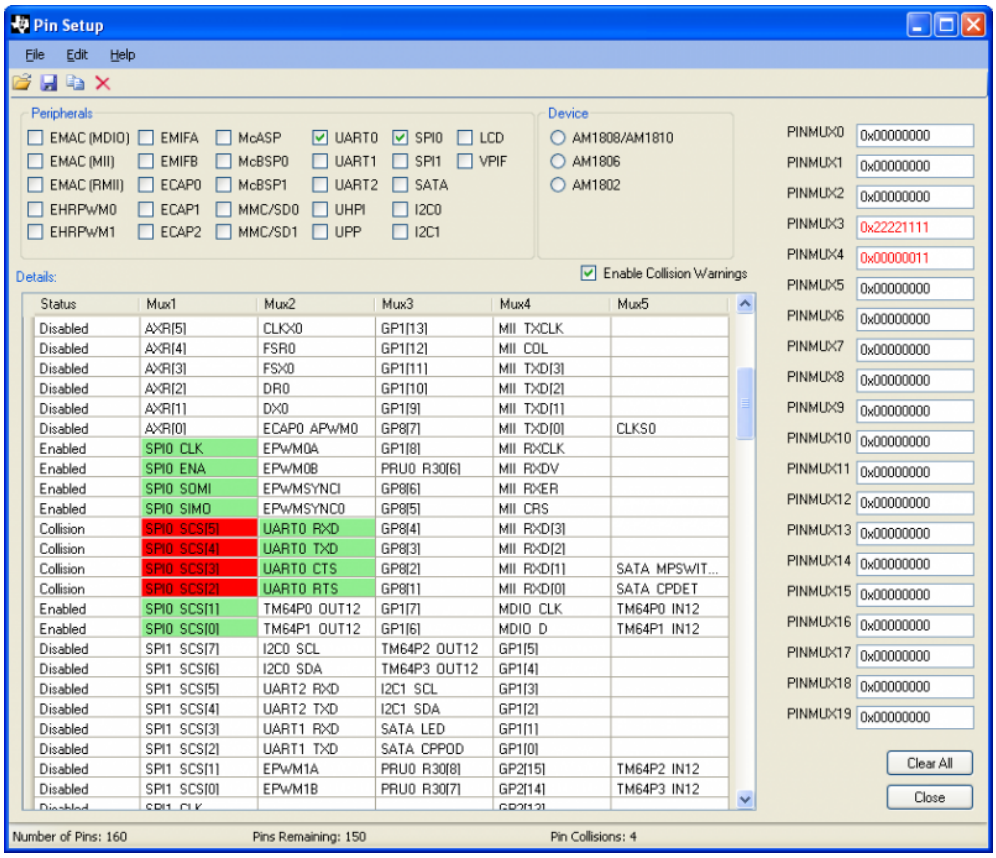
Resolving Conflicts

As an example of conflict resolution, first select the SPI0 peripheral, then the UART0 peripheral. A collision dialog box appears indicating a conflict. The decision to select the new signal or not can be made on a signal-by-signal basis or you can click Yes to All or No to All.

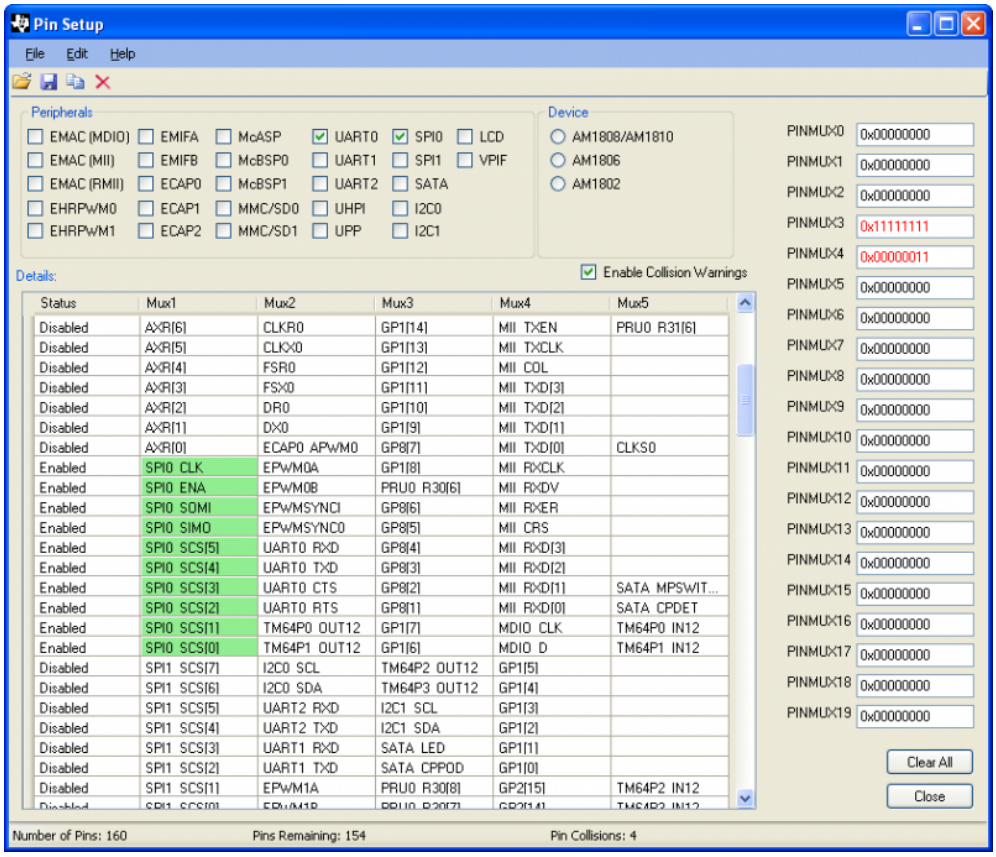


If Yes to All is selected, the UART0 signals will get selected as shown below. Note that the PINMUX3 register is selecting Mux2

for the UART0 signals. The conflicting SPI0 signals are no longer selected but are shown in red to indicate that a conflict occurred there.



If No to All is selected, the UART0 signals do not get selected and no change is made as shown below. Note that the PINMUX3 register is selecting Mux1 for the SPI0 signals.



Menu Items

File - Save - Header File

The File - Save - Header File manu item saves a list of the selected signals and #define constants for the Pin Mux register values. The default location is the My Documents folder. A file save dialog allows you to specifiy where to save the header file and to give the file a name.

File - Save - Pin Selections

The File - Save - Pin Selections menu item is used to save your work. The saved file can be read in using the File - Load - Pin File menu item. This file contains a list of all currently selected signals. saves the list of the selected signals and #define constants. Save your work before exiting the program, changing the device selecter radio buttons or clicking the Clear All button.

File - Load - Pin File

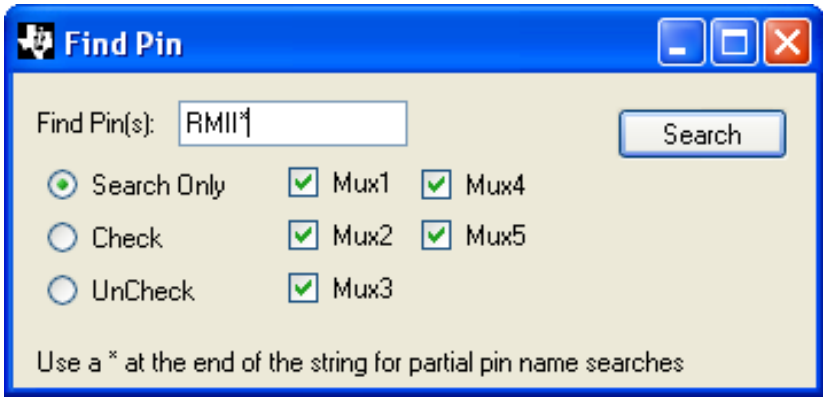
The File - Load - Pin File loads a file that was saved using the File - Save Pin Selections menu item.

File - Exit

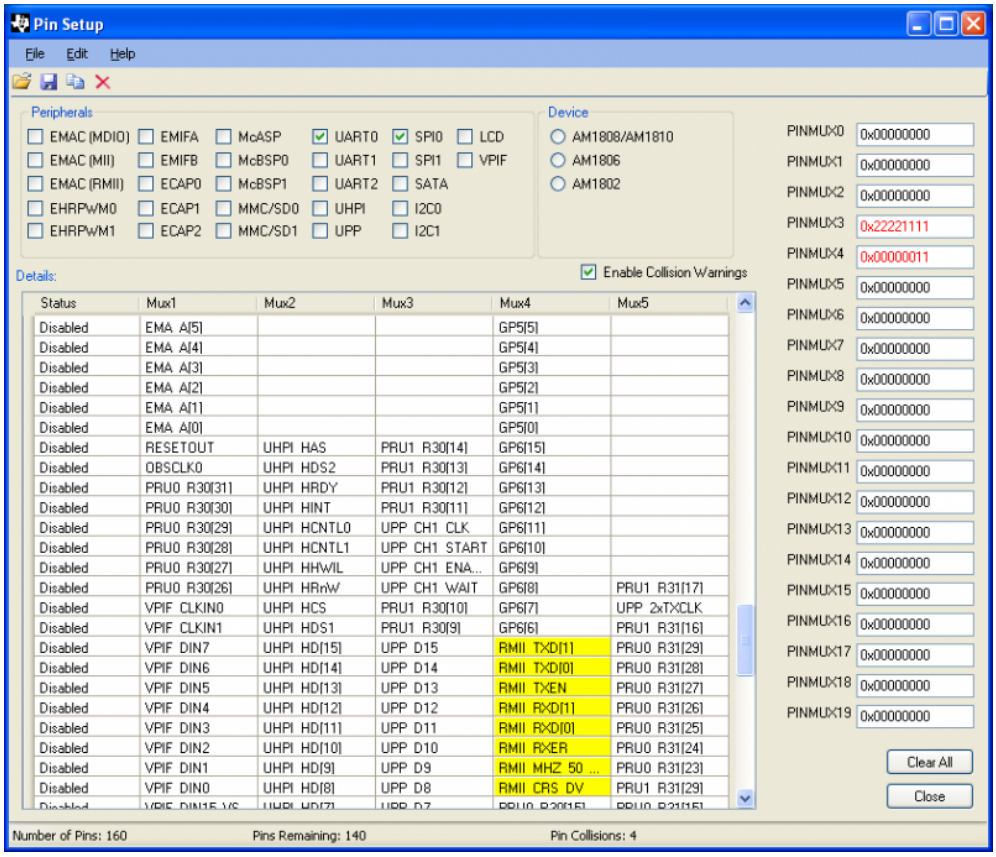
The File - Exit menu item closes the program. Program can also be closed by double-clicking the title bar icon at in upper left corner or by clicking the X icon. Save work using the File - Save - Pin Selections before exiting the program.

Edit - Find

The Edit - Find menu item open the Find Pin dialog. A signal name is entered. A asterisk "*" may be used as a wild card character. The is the option to search for the matching signal(s) or to select or unselect the matching signals.



Below is the result of searching for the RMII* signals.



Help - About

The Help - About menu item displays the program version information.

References

[1] http://www-s.ti.com/sc/techlit/spraba2.zip

[2] http://www.ti.com

Flash v1.3 User Guide

Installation Instructions

1. Locate the Flash application files on the Linux machine where the SDK was installed to. Typically, it will be installed to a folder under your home folder; i.e. /home/user/ti-sdk-x.x/host-tools/Windows/Flash-Utility.
2. Move all the files from /Flash-Utility to a Windows machine. The current version of the tool runs on Windows only. A good method for moving the files is to use a samba server (www.samba.org). (TBD: please add link to more detailed instructions)
3. Run setup.exe (or FlashInstaller.msi) to install the application on your PC. It can be uninstalled at any time by accessing the Add/Remove Programs function of Windows.
4. Follow the installer instructions.
5. Typical installation directory is c:\Program Files\Texas Instruments\Flash v1.3, but this can be modified from the installer's dialog screens.
6. Source code is provided in a zip file in /Flash-Utility/Source. If you want to view the source (not necessary), unzip the file. Later versions will provide complete developers documentation which will instruct users on how to rebuild the sources, which tools are required, etc.

About Flash v1.3

This page contains a description of a tool – Flash v1.3 – that can be used to transfer binary images from a host PC to certain TI ARM-based target platforms. The tool consists of two main components:

```
- a GUI host application, called Flash v1.3
- a CLI host application, called OMAPFlash.
```

It is recommended to use the GUI for performing flashing functions. This documentation covers usage of the GUI. Future documentation will cover the CLI interface in more detail.

This application has been designed with flexibility and portability in mind. It is now possible to modify target register configurations without rebuilding the tool. This allows for easy modifications to various target peripheral configurations (such as SDRC, GPMC, Pad Control, etc.). This capability makes it possible to support new DDR devices and NAND devices without software changes. Check out the section #Porting Guide for more information on this feature.

Internally, the tool makes use of a ROM code mechanism for peripheral boot from UART or USB to transfer compatible drivers to the internal memory of the OMAP device. These drivers provide the mechanism by which the host applications can program binary images into the internal memories (NAND and SDRAM) of the target. All of this operation is hidden from the user.

Release Notes

Version 1.3:

New Features

- GUI updates, including "What's This?" style help, link to user's guide.
- Defined Custom target types for AM37xx and OMAP35xx, to allow customers to more easily make changes for custom boards.
- Enhanced NAND capabilities: SW vs. HW ECC Selection, 1b/4b/8b Error Correction, ONFI NAND selection via check box.

Limitations

- Still no AM35xx support (coming in future release).
- Otherwise, similar functionality to previous releases.

Version 1.2:

New Features

- USB Support
- Can install over previous versions without manual uninstall
- Selectable NAND vs. ONFI NAND mode

Limitations

- Same as before

Version 1.1:

New Features

- 35xx Support.

Limitations

- Same as before

Version 1.0:

Features

- UART support
- 37xx support
- Can support new NAND devices via text config file modification.
- Can modify target registers via text config file modification (for example SDRC, GPMC, pad config settings)
- ONFI NAND Support
- Supported operations: Download, Download and Execute (to Thumb mode code), Erase Region, Erase All.
- Windows GUI (no Linux yet).
- Scriptable Windows CLI (no Linux yet).
- Fully open source, BSD-style license.

Limitations

- Download and execute can only branch to Thumb code.
- Storing images to NAND uses HWECC. Therefore, you cannot use a standard xloader to load uboot from NAND. Xloader uses SWECC when reading from NAND. The workaround is to use the uboot provided on the SD card to flash itself into NAND memory.
- GUI support does not yet exist to define new platforms. However, two memory choices are supplied, Hynix and Micron. For experimenting with modifications to the configuration text file (for instance, to port to a new platform) it is recommended that you modify the Micron files (and save a copy if you need the originals).

Accessing Online Help

There are two methods to get help from the user interface:

1. Click the "What's This Do?" Button. This changes the cursor type. Move the cursor over a widget until you see the cursor change to a question mark. Then, you may click and get context sensitive help.
2. Click the "Open User's Guide" Button. This will open your normal HTML browser to the Flash User's Guide Page (this page).

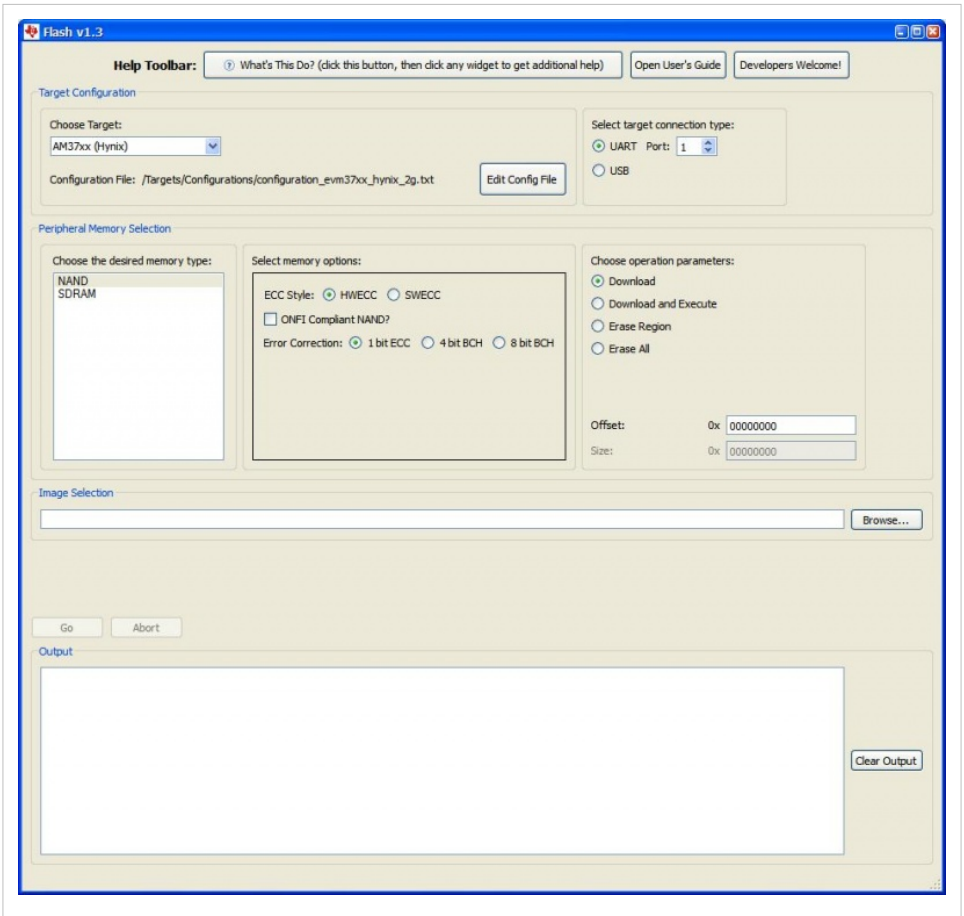
Usage Instructions (USB connection)

(board specific instructions relevant for the standard Mistral OMAP EVM)

1. Using a USB cable, connect the board's USB "On the Go" port to a USB port on the PC.
2. Ensure that your EVM is set up for peripheral boot from USB. This may require changing the setting of SW4 on the EVM. Switch bits 2 and 3 to the ON position. Other bits should be off.
3. After launching Flash app, ensure that the USB box is selected under "Select target connection type".
4. When you connect your board to your PC for the first time, you need to configure the PC to use the correct USB driver.
   - You will see the "Found New Hardware Wizard" dialog box. Select "No, no this time", then "Next".
   - Select "Install from a list or specific location (Advanced)".
   - Use the browse box to select the following folder: <install dir>\usb_drv_windows. Normally, the full path will be c:\Program Files\Texas Instruments\Flash vX.X\usb_drv_windows.
   - Make sure the box "search removable media" is unchecked.
   - Make sure the box "Include this location in the search" is checked.
   - Click Next. The driver should install correctly without error messages.
   - Click Finish.
5. Continue following the instructions below (UART connection) at #4.

Usage Instructions (UART connection)

1. (Using a 9-pin serial cable, physically connect your host PC to the UART3 port on the target board.
2. Ensure that your EVM is set up for peripheral boot from UART. This may require changing the setting of SW4 on the EVM. Switch bits 2 and 4 to the ON position. Other bits should be off.
3. There are three ways to launch the Flash app. You can use the shortcut on the desktop, the entry that was created under Start->Texas Instruments, or directly from the application installation directory. The executable name is Flash.exe. See below for a snapshot of the user interface:



4. Next to the label "Choose target:", select the appropriate target. You will see the associated target configuration file listed just below. You have the option of editing the configuration file by clicking "Edit Config File" button. This is typically useful when selecting one of the "Custom" target types. The Custom Targets have been provided to give customers an easy way to make changes that may be necessary for custom boards. Check out the section #Porting Guide for more information on how to edit configuration files for custom boards.
5. Next, select which peripheral memory you would like to program. Current choices are limited to NAND and SDRAM.
6. If you have selected SDRAM, there are no other memory options to select.
7. However, if you selected NAND, you will see a group of options under "Select memory options".
   - ECC Style: Choose HWECC (when programming X-loader to address 0, to be read by ROM), or SWECC (usually used when programming U-boot if using a standard X-loader which is expected U-boot to be stored with SWECC style).
   - ONFI Compliant NAND?: Select this option if you know your NAND device is ONFI compliant. Otherwise, your configuration file must contain certain NAND paramters (#Memories)
   - Error Correction: This is enabled if you have selected HWECC. Choose 1bit ECC always if you are programming X-loader or some other bootloader to address 0. For other memory regions, you have the option of selecting 4bit BCH or 8bit BCH for stronger error correction. These should only be selected if your NAND device requires greater than 1-bit protection. Also note that your X-loader must know how to load a U-boot image that has been programmed with 4-bit or 8-bit BCH.
8. Choose the UART port that you have connected to on the host PC. Typically this will be 1 (i.e. COM1), or 2 (COM2) if you have two serial ports.
9. Choose the operation that you want to perform:
   - Download - download a binary image to the target

- Download and Execute - download a binary image and execute it. Must be loaded to SDRAM.
- Erase Region - Erase a select region of memory. NOTE: With NAND Memory, REMEMBER TO ERASE IT BEFORE DOWNLOADING!
- Erase All - Erase all of the desired memory. NOTE: With NAND Memory, REMEMBER TO ERASE IT BEFORE DOWNLOADING!

10. Now, if the offset field is enabled, enter an offset from the start of the peripheral memory for your operation. This is a hexadecimal address value. X-loader is generally programmed to offset 0x0. U-boot generally to offset 0x80000. Kernel typically programmed to offset 0x280000.
11. If you have selected the operation "Erase Region", you will see the Size field enabled. Enter a size (hexadecimal, in bytes) that you want to clear. Clearing will begin at the selected offset.
12. For download operations, you must select a binary file that you want to download. Click the "Browse..." button in order to browse the filesystem to select a file. The default folder that is opened is the "test_data" folder under the application installation folder. There are a few example binaries located in that folder for evaluation purposes.
13. Once you have successfully made your selections, the "Go" button will be enabled. Press the "Go" button to execute your operation.
14. While the operation is executing, you will have the option of pressing the "Abort" button in order to cancel the operation and make modifications to your selections.
15. You will see the output of the operation in the Output text box at the bottom of the screen. If errors occur, related information will also be reported within the box.
16. If everything is setup correctly, you will soon see is a line in the output with the text "Awaiting ASIC ID". When you see this line, reset the target platform.

1. After you reset the target, your selected operation will be executed. Progress is reported within the output text box. Download speeds over UART are in the range of 10Kbps, so be patient if you have a large download - it could take several minutes.
2. At any time, you may clear all of the text in the output box by pressing the button "Clear Output".

Flash v1.3 (for developers)

- GForge Page: https://gforge.ti.com/gf/project/flash
- You can link directly to this page from the Flash GUI by clicking the "Developers Welcome!" button.

Feedback

- Several options exist for feedback:
  - Leave comments using the link at the bottom of the page. For any error conditions, please copy and paste the text in the output box into the body of the email. Also, identify your target board, processor, and memory types. Provide a synopsis of what you are trying to do.
  - Join the mailing list (arm_mpu_flash_tool@list.ti.com)
  - Join the development effort via the GForge page, and provide feedback there.

Porting Guide

[Remainder of document intended for advanced users]

This is a small porting guide for Flash. It contains an introduction to the modifications that may be necessary in order to get Flash working on a new board or with a new memory device.

Porting to a new platform

In general, porting to a new platform should not require the modification or recompilation of the code base. The starting point will be to create a board configuration file for the new platform.

Existing board configuration files are found in .\Targets\Configurations. In order to add support for a new platform it is recommended to modify configuration_custom_am37xx.txt (used when selecting target "Custom AM37XX Board") or configuration_custom_omap35xx.txt (used when selecting target "Custom OMAP35xx Board".

Once the board configuration file has been created, it will need to be added to the content of omapflash2nd.txt, present in the application installation folder. (NOTE: in the case of the prebuilt custom files, this is already done for you). This file allows Flash to find the board configuration file. When modifying the file, simply add a new line to it:

```
 <platform> <omap id> <omap version> <omap
type> <second loader file> -pheripheralboot_reopen
-board_config <board configuration file>
```

where <platform> is the platform name (use a new name for the new platform), <omap id> is an id number provided by the OMAP device during peripheral boot over UART or USB along with the <omap version> number, <omap type> specifies whether the OMAP detected should be a High Security 'HS' or General Purpose 'GP' type, <second loader file> specifies the second loader to use with the combination of <platform>, <omap id>, <omap version> and <omap type> and the <board configuration file> is the newly added board configuration.

An example could be:

```
 SDP_MDDR_HYNIX_4G 363007 07 GP
Targets\2nd-Downloaders\dnld_startup_omap3_gp_4g.2nd
-pheriphalboot_reopen
 -board_config
Targets\Configurations\configuration_sdp3630_hynix_4g.txt
```

Note that the installation currently comes with second loaders supporting memory sizes of 2 Gb, 4 Gb and 8 Gb for GP and HS devices:

```
dnld_startup_omap3_gp_2g.2nd – OMAP3 GP w 2 Gb SDRAM
dnld_startup_omap3_gp_4g.2nd – OMAP3 GP w 4 Gb SDRAM
dnld_startup_omap3_gp_8g.2nd – OMAP3 GP w 8 Gb SDRAM
dnld_startup_omap3_hs_2g.2nd – OMAP3 HS w 2 Gb SDRAM
dnld_startup_omap3_hs_4g.2nd – OMAP3 HS w 4 Gb SDRAM
dnld_startup_omap3_hs_8g.2nd – OMAP3 HS w 8 Gb SDRAM
dnld_startup_omap4_gp_2g.2nd – OMAP4 GP w 2 Gb SDRAM
dnld_startup_omap4_gp_4g.2nd – OMAP4 GP w 4 Gb SDRAM
dnld_startup_omap4_gp_8g.2nd – OMAP4 GP w 8 Gb SDRAM
dnld_startup_omap4_hs_2g.2nd – OMAP4 HS w 2 Gb SDRAM
dnld_startup_omap4_hs_4g.2nd – OMAP4 HS w 4 Gb SDRAM
dnld_startup_omap4_hs_8g.2nd – OMAP4 HS w 8 Gb SDRAM
```

The reason for this is that there is a link-time dependency on the placement of the heap and external memory for the memory device drivers in SDRAM and on the location of the second loader components in internal memory between HS and GP devices. Pick the right one - e.g. if you have a 4 Gbit memory on an OMAP3 GP based board, use 'dnld_startup_omap3_gp_4g.2nd'.

Board configuration

In order to create the new file it is often useful to start with a copy of one of the existing files. The file has three main sections:

1. 'use' directive, pointing to a definition file listing a number of OMAP registers and their addresses
2. 'memory' directives, specifying the memories on the platform
3. initialization commands, modifying registers to control the configuration of the OMAP to match the platform

Definitions

A 'use' directive can be used to point to a device definition file, e.g.:

```
use definitions_omap3.txt
```

Only one definition file can be used and it must be indicated before the first element using its definitions occurs in the configuration file.

The device definition file basically holds a set of paired of labels and values. The labels can be used in the device configuration file in place of the values in order to make the device configuration file more readable. The syntax is as follows:

```
PRM_CLKSRC_CTRL                          0x48307270
CM_CLKEN_PLL                             0x48004D00
PRM_CLKSEL                              0x48306D40
CM_CLKSEL1_PLL                           0x48004D40
CM_CLKSEL2_PLL                           0x48004D44
CM_CLKSEL3_PLL                           0x48004D48
```

A maximum of 1000 definition pairs can be present in a definition file.

Memories

Memories are specified using the 'memory' directive:

```
 memory NAME [driver DRIVER] [parameters PARAMETER1 VALUE1 PARAMETER2
VALUE2 ... PARAMETERN VALUEN]
```

An example of a memory specification could be:

```
 memory NAND driver Targets\Flash-Drivers\nand_onfi_16bit_8bit.bin
parameters gpmc 0x6E000000 cs 1 address 0x28000000 bberase 0
```

where the device name is NAND and the driver required to access it is present in the binary file nand_onfi_16bit_8bit.bin (part of this distribution). The driver needs a number of configuration parameters for correct operation. These are passed directly to driver as written on the line following the 'parameters' keyword. This distribution contains a number of driver binaries for various memory types. At present these are:

```
 File      : nand_onfi_16bit_8bit.bin
 Type      : NAND
 Parameters: gpmc      (mandatory)               Base address of the
GPMC in the OMAP
             cs        (mandatory)               Chip select where the
device is present (or GPMC-config index)
             address   (mandatory)               Address of the device
as mapped in the GPMC
             bberase   (mandatory)               Erase bad blocks in
the device (0 for no, 1 for yes). Caution: erasing bad blocks may cause
 an irreversible loss of manufacturing information.
             onfi      (optional)                Read and use ONFI
device description from the device (0 for no, 1 for yes).
             bpp       (mandatory if onfi = 0)   Bytes per page if ONFI
 information is not used.
             sbpp      (mandatory if onfi = 0)   Spare bytes per page
if ONFI information is not used.
             ppb       (mandatory if onfi = 0)   Pages per block if
ONFI information is not used
             bpl       (mandatory if onfi = 0)   Blocks per logical
unit if ONFI information is not used
             l         (mandatory if onfi = 0)   Logical unit count
(only 1 supported by the driver)
             acv       (mandatory if onfi = 0)   Address cycle values -
 8 bit value with lower 4 bits for row and upper 4 bits for
                                                 column.
             f         (mandatory if onfi = 0)   Features - 16 bit
value with bit 0 = 16 bit data operation, rest are don't-care
 Examples  : memory NAND driver
Targets\Flash-Drivers\nand_onfi_16bit_8bit.bin parameters gpmc
0x6E000000 cs 1 address 0x28000000
             bberase 0
             memory NAND driver
Targets\Flash-Drivers\nand_onfi_16bit_8bit.bin parameters gpmc
```

```
0x6E000000 cs 1 address 0x28000000
             bberase 0 onfi 0 bpp 2048 sbpp 64 ppb 64 bpl 4096 l 1 acv
0x23 f 0x0019
```

For SDRAM, no driver is required, but the memory type must be specified with one parameter stating the base address in the memory map, e.g.:

```
 memory SDRAM parameters address 0x80000000
```

Initialization of Target Device

A number of register operation commands can be used to configure the target device:

- WRITE - Write a value to a register
- MODIFY - Modify the value of a register
- POLL_ZERO - Poll a register value until zero
- POLL_NZERO - Poll a register value until not zero
- POLL_VALUE - Poll a register until value
- WAIT_N - Loop n times in a simple while-loop (where n is a value from 0x0000 to 0xFFFF)
- SPIN - Loop forever. This may be used for debugging.
- MODE_16 - Use 16 bit register access mode
- MODE_32 - Use 32 bit register access mode

The command structures are:

- WRITE : WRITE REGISTER VALUE
- MODIFY : MODIFY REGISTER MASK VALUE
- POLL_ZERO : POLL_ZERO REGISTER MASK
- POLL_NZERO : POLL_NZERO REGISTER MASK
- POLL_VALUE : POLL_VAL REGISTER MASK VALUE
- WAIT_N : WAIT_N N
- SPIN : SPIN
- MODE_16 : MODE_16
- MODE_32 : MODE_32

The definitions included from a definition file specified in a 'use' directive can be used with the commands. Definitions can be used for registers, values and masks, e.g.:

```
 MODIFY         CM_CLKEN_PLL_MPU     EN_XXX_DPLL_MODE_MASK
EN_XXX_DPLL_LOCK_MODE
 WRITE          CM_CLKSEL3_PLL       0x00000009
 POLL_ZERO      CM_IDLEST_CKGEN      ST_PERIPH_CLK_DPLL4_LOCKED
```

AM18x Flash Tool User's Guide

The followingdetails the procedures required to Flash the AM18x EVM.

In future releases AM18x supported will be inlcuded in the → Flash Tool

1. Download serial flashing tool here : http://sourceforge.net/projects/dvflashutils/files/OMAP-L138/ [1]. Get the latest version, 2.3.0.
2. Extract the serial flashing tool onto your Windows machine. It creates a file structure that begins with the folder **OMAP-L138_FlashAndBootUtils_2_30**.
3. Download Microsoft .NET 4.0 framework [2]
4. Install .NET 4.0 framework. Reboot machine if it asks you.
5. Get the PSP (03.30.00.02 is the latest [3]) Extract it to your Windows machine.
6. Copy u-boot.bin from the psp into the OMAP-L138_FlashAndBootUtils_2_30\OMAP-L138\GNU folder. The U-boot file is located here: C:\OMAPL138-PSP-SDK\DaVinci-PSP-SDK-03.30.00.02\images\u-boot\omapl1x8\u-boot.bin, assuming you extracted the PSP directly into C:\.
7. Open a command prompt window and navigate to OMAP-L138_FlashAndBootUtils_2_30\OMAP-L138\GNU.
8. Set the boot pins to UART2 boot mode. This is done by setting switch S7 on the AM18x EVM according to the following table:

AM180x EVM (Logic)

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	**OFF**	**ON**	ON

AM1810 EVM (Spectrum Digital)

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	**ON**	**OFF**	ON

9. Flash uboot using one of the following commands: (Please note, if you are using Windows command prompt, do not copy and past the command, you must type it at the command prompt)

For AM180x EVM (Logic): Execute the command “sfh_OMAP-L138.exe –flash ubl\ubl_OMAPL138_SPI_MEM.bin u-boot.bin”.

For AM1810 EVM (Spectrum Digital): Execute the command “sfh_OMAP-L138.exe –targetType INTDEV0 –flash ubl\ubl_INTDEV0_SPI_MEM.bin u-boot.bin”.

10. Press S5 to reset the target board when the serial flasher program requests it.

11. Set the boot pins to SPI1 boot mode. This is done by setting switch S7 on the AM18x EVM according to the following table:

AM180x EVM (Logic)

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	**OFF**	**OFF**	OFF

AM1810 EVM (Spectrum Digital)

Pin#	1	2	3	4	5	6	7	8
Position	OFF	OFF	OFF	OFF	OFF	**ON**	**ON**	OFF

12. Reset the platform to obtain the uboot prompt

References

[1] http://sourceforge.net/projects/dvflashutils/files/OMAP-L138/
[2] http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en
[3] ftp://ftp.india.ti.com/PSP/Releases/ODC/DaVinci-PSP-SDK/DaVinci-PSP-SDK-03.30.00.02.tgz

Error Correction User Guide

Overview

Beginning with SDK 4.01, NAND error correction greater than 1-bit has been added to Xloader, Uboot, and the kernel. This document is intended to briefly describe the usage of the 4-bit and 8-bit error correction capabilities.

The 4-bit and 8-bit capabilities are based on a BCH software correction algorithm, and rely on GPMC hardware for BCH computations.

Be careful to be consistent when moving to a system with higher error correction capability. For example, if you desire 4-bit error correction, be sure to build a 4-bit capable Xloader based on the instructions below. Also, be sure to compile a 4-bit version of the kernel (again, see instructions below). Finally, be sure to program all binaries (Xloader/Uboot/kernel) to NAND using 4-bit correction via either Flash tool or Uboot.

X-Loader

X-loader has compile time selectable support for 4-bit and 8-bit error correction.

When building X-loader, use the following command (for 1 bit ECC):

make CROSS_COMPILE=arm-none-linux-gnueabi- ARCH=arm PLATFORM_RELFLAGS+=-DONE_BIT_ERROR_CORRECT

1. For 4-bit error correction, use -DFOUR_BIT_ERROR_CORRECT instead.
2. For 8-bit error correction, use -DEIGHT_BIT_ERROR_CORRECT instead.

U-boot

Error correction mode is controlled by U-boot 'nandecc' command. Options for 'nandecc' command:

- nandecc hw - This configures for 1-bit HWECC (ROM Style, for programming images intended for ROM to boot, such as X-loader).
- nandecc sw - This configures for 1-bit SWECC.
- nandecc bch4_sw - This configures for 4-bit BCH error correction.
- nandecc bch8_sw - This configures for 8-bit BCH error correction.

The default mode is 4-bit error correction. This is set at the end of board_nand_init().

Kernel

You must choose 1bit ECC, 4bit BCH, or 8-bit BCH at compile time.

To change error correction mode, you must modify omap2.c and uncomment either CONFIG_MTD_NAND_OMAP_BCH_4 (to enable 4-bit BCH) or CONFIG_MTD_NAND_OMAP_BCH_8 (to enable 8-bit BCH). The default is 4-bit ECC correction. Comment out both of them if you want 1-bit ECC mode.

Flash Tool

The Flash Tool has been updated with support for programming images to NAND flash using 1bit SWECC, 1bit HWECC, 4bit BCH, or 8bit BCH. You may use Flash Tool for NAND programming or you may use U-boot, as they have similar functionality.

For more information please consult the Flash User Guide: [Flash v1.3 User's Guide [1]]

References

[1] http://processors.wiki.ti.com/index.php/Flash_v1.3_User_Guide

Article Sources and Contributors

Sitara Linux Software Developer's Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=51351 *Contributors*: Gregturne, Kevinsc, Mike Tadyshak

How to Build a Ubuntu Linux host under VMware *Source*: http://processors.wiki.ti.com/index.php?oldid=50377 *Contributors*: Gregturne, Kevinsc

Installing CodeSourcery Toolchain *Source*: http://processors.wiki.ti.com/index.php?oldid=48549 *Contributors*: Kevinsc, SanjeevPremi, SchuylerPatton

Sitara SDK Installer *Source*: http://processors.wiki.ti.com/index.php?oldid=50935 *Contributors*: Kevinsc, Mike Tadyshak

SDK Setup Script *Source*: http://processors.wiki.ti.com/index.php?oldid=50540 *Contributors*: Kevinsc

Code Composer Studio v5 Preview Edition Users Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=50924 *Contributors*: Alexander.stohr, John, Kevinsc, Mike Tadyshak

How to start Code Composer Studio v5 *Source*: http://processors.wiki.ti.com/index.php?oldid=48323 *Contributors*: Mike Tadyshak

How to Import Matrix GUI into CCSv5 *Source*: http://processors.wiki.ti.com/index.php?oldid=48321 *Contributors*: ClayMontgomery, Jefflance01, John, Mike Tadyshak

How to Build the Matrix GUI *Source*: http://processors.wiki.ti.com/index.php?oldid=50140 *Contributors*: Alexander.stohr, ClayMontgomery, Jefflance01, Kevinsc, Mike Tadyshak

How to Import Matrix Projects into CCSv5 *Source*: http://processors.wiki.ti.com/index.php?oldid=50168 *Contributors*: Jefflance01, Mike Tadyshak

How to setup Remote System Explorer plug-in *Source*: http://processors.wiki.ti.com/index.php?oldid=49655 *Contributors*: ChrisRing, Mike Tadyshak

How to Run GDB on CCSv5 *Source*: http://processors.wiki.ti.com/index.php?oldid=49931 *Contributors*: Alexander.stohr, Jefflance01, John, Kevinsc, Mike Tadyshak, SchuylerPatton

Matrix Users Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=50819 *Contributors*: Alexander.stohr, BernieThompson, ChrisRing, ClayMontgomery, Gregturne, Jefflance01, Kevinsc, Mike Tadyshak, Stevek

How to use a Mouse instead of the Touchscreen with Matrix *Source*: http://processors.wiki.ti.com/index.php?oldid=49993 *Contributors*: ClayMontgomery, Jefflance01

How to Switch Display From Touchscreen to DVIout with Matrix *Source*: http://processors.wiki.ti.com/index.php?oldid=42880 *Contributors*: Gregturne

How to Make a Hello World Matrix Application *Source*: http://processors.wiki.ti.com/index.php?oldid=39865 *Contributors*: BernieThompson

Cryptography Users Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=51342 *Contributors*: Gregturne, Kevinsc

Pin Mux Utility for ARM MPU Processors *Source*: http://processors.wiki.ti.com/index.php?oldid=51216 *Contributors*: Alexander.stohr, Erives, Kevinsc, Kevinvap, Mike Tadyshak

Pin Setup Tool for AM18xx ARM Microprocessors *Source*: http://processors.wiki.ti.com/index.php?oldid=49972 *Contributors*: Mike Tadyshak

Flash v1.3 User Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=48941 *Contributors*: Gguyotte

AM18x Flash Tool User's Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=50912 *Contributors*: Gguyotte, Kevinsc

Error Correction User Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=50665 *Contributors*: Gguyotte

Image Sources, Licenses and Contributors

Image:TIBanner.png *Source*: http://processors.wiki.ti.com/index.php?title=File:TIBanner.png *License*: unknown *Contributors*: Nsnehaprabha
Image:AM37xSW.png *Source*: http://processors.wiki.ti.com/index.php?title=File:AM37xSW.png *License*: unknown *Contributors*: Kevinsc
Image:Linux dev environ.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Linux_dev_environ.jpg *License*: unknown *Contributors*: Kevinsc
Image:Ubuntu com.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Ubuntu_com.png *License*: unknown *Contributors*: Kevinsc
Image:Vmware sh1.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_sh1.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware sh2.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_sh2.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware sh4.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_sh4.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware sh6.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_sh6.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware sh7.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_sh7.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware sh8.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_sh8.jpg *License*: unknown *Contributors*: Kevinsc
Image:Network proxy setup.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:Network_proxy_setup.JPG *License*: unknown *Contributors*: Kevinsc
Image:Network proxy preferences.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:Network_proxy_preferences.JPG *License*: unknown *Contributors*: Kevinsc
Image:Network proxy password.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:Network_proxy_password.JPG *License*: unknown *Contributors*: Kevinsc
Image:Vmware kb1.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_kb1.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware kb2.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_kb2.jpg *License*: unknown *Contributors*: Kevinsc
Image:Vmware kb3.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Vmware_kb3.jpg *License*: unknown *Contributors*: Kevinsc
Image:AM37x Env ov 1 KC.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:AM37x_Env_ov_1_KC.jpg *License*: unknown *Contributors*: SchuylerPatton
Image:AM37x Toolchain info req.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:AM37x_Toolchain_info_req.JPG *License*: unknown *Contributors*: SchuylerPatton
Image:AM37x Toolchain dir selection.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:AM37x_Toolchain_dir_selection.JPG *License*: unknown *Contributors*: SchuylerPatton
Image:AM37x Toolchain found.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:AM37x_Toolchain_found.JPG *License*: unknown *Contributors*: SchuylerPatton
Image:AM37x Sdk dir selection.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:AM37x_Sdk_dir_selection.JPG *License*: unknown *Contributors*: SchuylerPatton
Image:CCSv5 Icon 001.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:CCSv5_Icon_001.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:Eclipse Splash 002.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Eclipse_Splash_002.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:PickWorkspace.png *Source*: http://processors.wiki.ti.com/index.php?title=File:PickWorkspace.png *License*: unknown *Contributors*: Jefflance01
Image:Welcome-CCSv5 003.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Welcome-CCSv5_003.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:Import Qt 006.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Import_Qt_006.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:ImportQTWizard.png *Source*: http://processors.wiki.ti.com/index.php?title=File:ImportQTWizard.png *License*: unknown *Contributors*: Jefflance01
Image:Select-Qt-Project-File 007.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Select-Qt-Project-File_007.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:OpenC++perspective.png *Source*: http://processors.wiki.ti.com/index.php?title=File:OpenC++perspective.png *License*: unknown *Contributors*: Jefflance01
Image:NeedtoSetupQT.png *Source*: http://processors.wiki.ti.com/index.php?title=File:NeedtoSetupQT.png *License*: unknown *Contributors*: Jefflance01
Image:AddNewQT.png *Source*: http://processors.wiki.ti.com/index.php?title=File:AddNewQT.png *License*: unknown *Contributors*: Jefflance01
Image:Qt-add-version 008.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Qt-add-version_008.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui select build.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_select_build.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui clean.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_clean.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui building.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_building.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui build output.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_build_output.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui local files.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_local_files.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui paste.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_paste.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui delete.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_delete.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui rename.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_rename.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Matrix gui permissions.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_gui_permissions.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Import C projects-1.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Import_C_projects-1.png *License*: unknown *Contributors*: Mike Tadyshak
Image:SelectBrowse.png *Source*: http://processors.wiki.ti.com/index.php?title=File:SelectBrowse.png *License*: unknown *Contributors*: Jefflance01
Image:Import-projects.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Import-projects.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Import-projects-list.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Import-projects-list.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Imported-projects.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Imported-projects.png *License*: unknown *Contributors*: Mike Tadyshak
Image:New-connection.png *Source*: http://processors.wiki.ti.com/index.php?title=File:New-connection.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Remote-system-type.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Remote-system-type.png *License*: unknown *Contributors*: Mike Tadyshak
Image:X-New Connection.png *Source*: http://processors.wiki.ti.com/index.php?title=File:X-New_Connection.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Ssh-files.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Ssh-files.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Processes .png *Source*: http://processors.wiki.ti.com/index.php?title=File:Processes_.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Shells.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Shells.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Terminals.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Terminals.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Show-view-remote-systems.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Show-view-remote-systems.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Open-RSE-Detached.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Open-RSE-Detached.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:X-change-user-id.png *Source*: http://processors.wiki.ti.com/index.php?title=File:X-change-user-id.png *License*: unknown *Contributors*: Mike Tadyshak
Image:X-network-Connections-top.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:X-network-Connections-top.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:X-network-Connections-bottom.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:X-network-Connections-bottom.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:Setup-ssh-editted-1.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Setup-ssh-editted-1.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Nasty-Warning-2.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Nasty-Warning-2.PNG *License*: unknown *Contributors*: Mike Tadyshak
Image:X-login.png *Source*: http://processors.wiki.ti.com/index.php?title=File:X-login.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Expand-root-small.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:Expand-root-small.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:MyTerminalView-small.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:MyTerminalView-small.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:MyShellView-small.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:MyShellView-small.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:MyProcessList-small.jpeg *Source*: http://processors.wiki.ti.com/index.php?title=File:MyProcessList-small.jpeg *License*: unknown *Contributors*: Mike Tadyshak
Image:Debug-Config-1.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-1.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Debug-Config-2.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-2.png *License*: unknown *Contributors*: Mike Tadyshak
Image:Debug-Config-3.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-3.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Debug-Config-4.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-4.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Debug-Config-6.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-6.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Debug-Config-7.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-7.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Debug-Config-8.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Debug-Config-8.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Shell-processes.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Shell-processes.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Kill-gdbserver.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Kill-gdbserver.png *License*: unknown *Contributors*: Mike Tadyshak

Image:Matrix latest.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_latest.png *License*: unknown *Contributors*: Jefflance01

Image:Desc_benchmarks.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Desc_benchmarks.png *License*: unknown *Contributors*: Jefflance01

Image:Matrix Main Menu Small.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_Main_Menu_Small.jpg *License*: unknown *Contributors*: ClayMontgomery

Image:Matrix full size.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Matrix_full_size.PNG *License*: unknown *Contributors*: Jefflance01

Image:MatrixMain.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:MatrixMain.PNG *License*: unknown *Contributors*: Gregturne

Image:MatrixCryptos.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:MatrixCryptos.PNG *License*: unknown *Contributors*: Gregturne

Image:SecureServer.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:SecureServer.PNG *License*: unknown *Contributors*: Gregturne

Image:Pin Mux Utility v1.0 Installer Start.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_Installer_Start.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin_Mux_Utility_v1.0_Installer_Remove-2.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_Installer_Remove-2.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 Installer Complete.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_Installer_Complete.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 TI Icon.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_TI_Icon.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image: Pin_Mux_Utility_v1.0_Device_Type_Selection_3.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_Device_Type_Selection_3.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Main Window 3.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Main_Window_3.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Open Design.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Open_Design.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 Legend.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_Legend.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Conflict MCBSP1 3.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Conflict_MCBSP1_3.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Conflict UART1 3.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Conflict_UART1_3.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Pad Config Editor.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Pad_Config_Editor.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Save Device Header 3.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Save_Device_Header_3.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Save Board Header 4.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Save_Board_Header_4.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Device Header File 4.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Device_Header_File_4.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Board Header File 4.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Board_Header_File_4.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Pin Mux Utility v1.0 AM37xx Modified Board Header File.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Pin_Mux_Utility_v1.0_AM37xx_Modified_Board_Header_File.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Bin folder.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Bin_folder.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Program Startup.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Program_Startup.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Collision Warning.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Collision_Warning.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:After Selecting UART0.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:After_Selecting_UART0.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:After Not Selecting UART0.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:After_Not_Selecting_UART0.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Saved Header File.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Saved_Header_File.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Find Pin.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Find_Pin.PNG *License*: unknown *Contributors*: Mike Tadyshak

Image:Find Pin Results.PNG *Source*: http://processors.wiki.ti.com/index.php?title=File:Find_Pin_Results.PNG *License*: unknown *Contributors*: Mike Tadyshak

File:TIBanner.png *Source*: http://processors.wiki.ti.com/index.php?title=File:TIBanner.png *License*: unknown *Contributors*: Nsnehaprabha

File:Flash_1.3_gui.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Flash_1.3_gui.jpg *License*: unknown *Contributors*: Gguyotte

File:Evm_revg_reset.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:Evm_revg_reset.JPG *License*: unknown *Contributors*: Gguyotte

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.
BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

License

1. Definitions

1. "**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
2. "**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
3. "**Creative Commons Compatible License**" means a license that is listed at http://creativecommons.org/compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
4. "**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
5. "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
6. "**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
7. "**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
8. "**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
9. "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
10. "**Publicly Perform**" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
11. "**Reproduce**" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
2. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
3. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
4. to Distribute and Publicly Perform Adaptations.
5. For the avoidance of doubt:

    1. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
    2. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
    3. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
2. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.
3. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Ssection 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
4. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
2. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
6. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.