Development Environment User Guide

For C/C++ applications in GE863-PRO³ with Linux

1vv0300780 Rev.2 -21/05/09

Making machines talk.

Disclaimer

The information contained in this document is the proprietary information of Telit Communications S.p.A. and its affiliates (“TELIT”).

The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Telit, is strictly prohibited.

Telit makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Telit does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information.

Telit disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules.

Telit reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice.

Such changes will, nevertheless be incorporated into new editions of this document.

Applicable Products

GE863-PRO3	GE863PR3***-*** GE863P3********

The suffix “***-***” and “********” depends on the module configuration and packaging.
Please contact your Telit representative for details

Linux SW Version	Recommended U-Boot version
05.0005	20.00.0000

Contents

1 Introduction

1.1 Scope

This user guide serves the following purpose:

- Describes how to install Telit Development Environment.
- Describes how to develop, deploy and debug a simple application on the PRO^{3} with Linux operating system.

1.2 Audience

This User Guide is intended for software developers who develop applications in Linux on the GE863-PRO^{3} module.

1.3 Contact Information, Support

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

For general contact, technical support, report documentation errors and to order manuals, contact Telit's Technical Support Center at:

TS-EMEA@telit.com or http://www.telit.com/en/products/technical-support-center/contact.php

Telit appreciates feedback from the users of our information.

1.4 Open Source Licenses

Telit Development Environment consists of different Open Source Software licensed as described in the following paragraphs.

1.4.1 Linux

Linux is an operating system for computers; it was originally created starting in 1991 by Finnish programmer Linus Torvalds with the assistance of developers from around the globe. It is licensed under the Free Software Foundation's GNU Project's GNU General Public License, version 2.

1.4.2 coLinux

Cooperative Linux (short-named coLinux) is a port of the Linux kernel that allows it to run cooperatively alongside another operating system on a single machine. It is released under GNU General Public License.

1.4.3 Xming

Xming is the leading free unlimited X Window Server for Microsoft Windows. It is released under MIT/X11 license.

1.4.4 Eclipse IDE

Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. Its projects are released under the Eclipse Public License.

1.5 Product Overview

The GE863-PRO^3 is an innovation to the quad-band, RoHS compliant GE863 product family which includes a powerful ARM9^{TM} processor core exclusively dedicated to customer applications. The concept of collocating a powerful processor core with the GSM/GPRS engine allows developers to host their application directly. The PRO^3 incorporates much of the necessary hardware for communicating microcontroller solutions, including the critical element of memory, significant simplification of the bill of material, vendor management, and logistics effort are achieved.

1.6 Document Organization

This manual contains the following chapters:

- “Chapter 1, Introduction” provides a scope for this manual, target audience, technical contact information, and text conventions.

- “Chapter 2, PRO^3 Hardware interfaces” describes the various hardware interfaces of the PRO^3.
- “Chapter 3, Development Environment” describes the files provided with the installer, the hardware requirements and the installation process.
- “Chapter 4, Connecting the Target to the Host” describes how to setup the various possible connection configurations between the target and the host.
- “Chapter 5, Starting Eclipse” describes the first steps with the IDE.
- “Chapter 6, Creating a Project” describes how to create a C/C++ project and how to deploy it to the target.
- “Chapter 7, Debugging an Application” describes how to debug your programs.
- “Chapter 8, General Suggestion“ contains some practical advices
- Various appendixes with useful information for other tasks.

1.7 Text Conventions

This section lists the paragraph and font styles used for the various types of information presented in this user guide.

Format	**Content**
`Arial monospaced`	Linux shell commands at command prompt.

1.8 Related Documents

The following documents are related to this user guide:

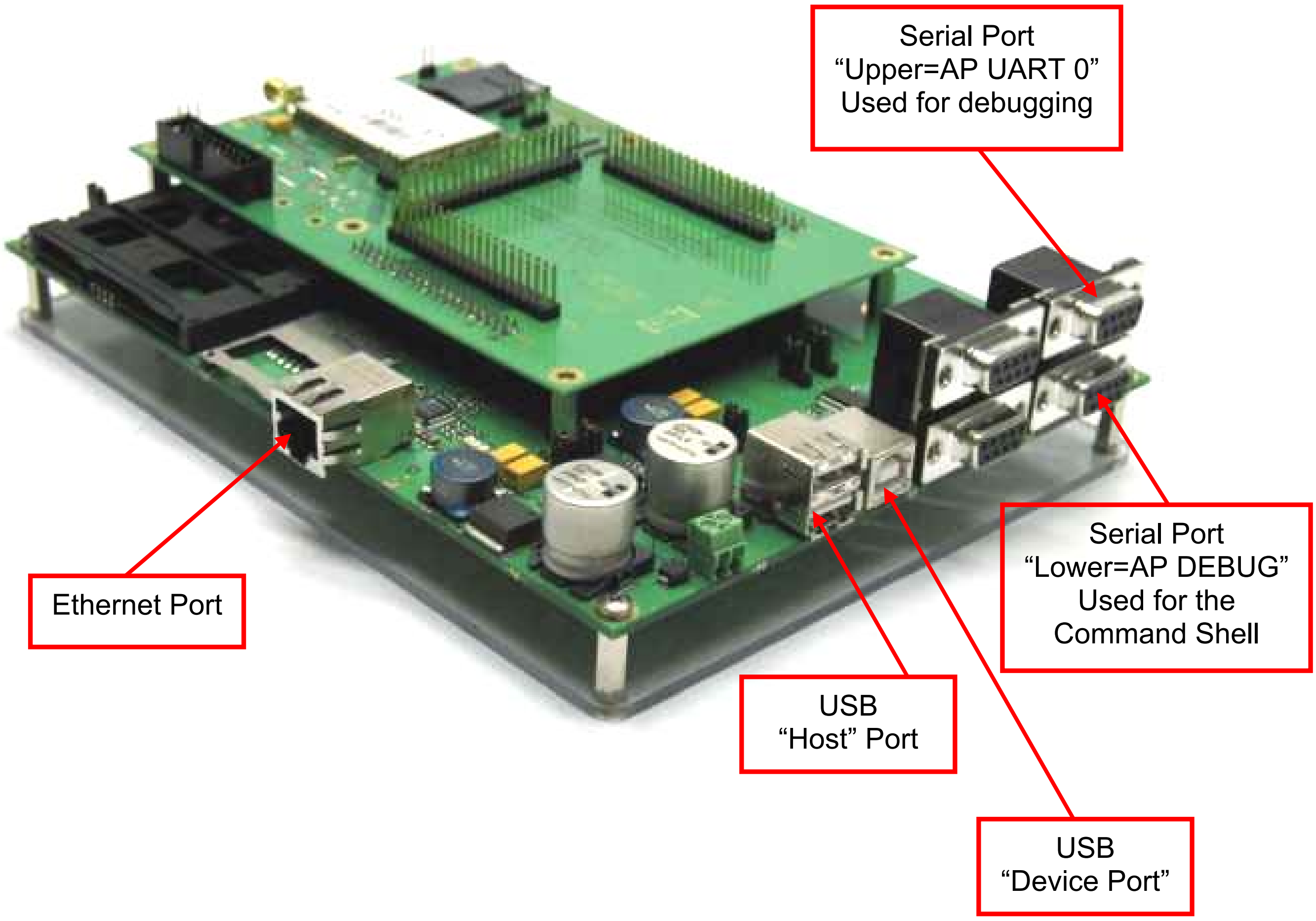
[1] TelitGE863PRO3 Hardware User Guide 1vv0300773a
[2] TelitGE863PRO3 EVK User Guide 1VV0300776
[3] TelitGE863PRO3 Linux SW User Guide 1vv0300781
[4] TelitGE863PRO3 Linux GSM Library User Guide 1vv0300782
[5] TelitGE863PRO3 Product Description 80285ST10036a
[6] TelitGE863PRO3 U-boot Software User Guide 1vv0300777

All documentation can be downloaded from Telit’s official web site www.telit.com if not otherwise indicated.

1.9 Document History

Revision	Date	Changes
ISSUE #0	26/05/08	First Release
ISSUE #1	31/07/08	Added new paragraph: 8 General Suggestion
ISSUE #2	21/05/09	6.2 Linking an application against a library: Added Appendix E: Upgrading the firmware kernel sources… modified Appendix F: Upgrading U-boot… added

2 PRO^3 Hardware interfaces



For further details about hardware interfaces refer to document [1] and document [2].

3 Development environment

This section contains information about the development environment for the host. The development environment is based on coLinux and Xming that permit you to work in Linux under Microsoft Windows XP.

3.1 Available Files

The development environment consists of the following file:

- setupDevEnvironment_C_Cpp.exe: the development environment C/C++ version.

This file is available on the Telit web site www.telit.com in the download zone.

3.2 System Requirements

To install the development environment, a 32-bit x86 platform with the following characteristics is required:

- Microsoft Windows XP installed
- At least 1GB of Ram
- 5,1 GB of hard disk space

3.3 Installation

This paragraph shows how to install and configure the development environment on a Windows XP-based host PC. Note that the development environment must be installed with an administrator account.

For a correct installation and usage of Telit Development Environment it is strongly suggested to turn off any software firewall (especially Windows Firewall service) active in the host system. Otherwise an aimed configuration of the firewall is needed in order to allow the communications between Windows and coLinux.

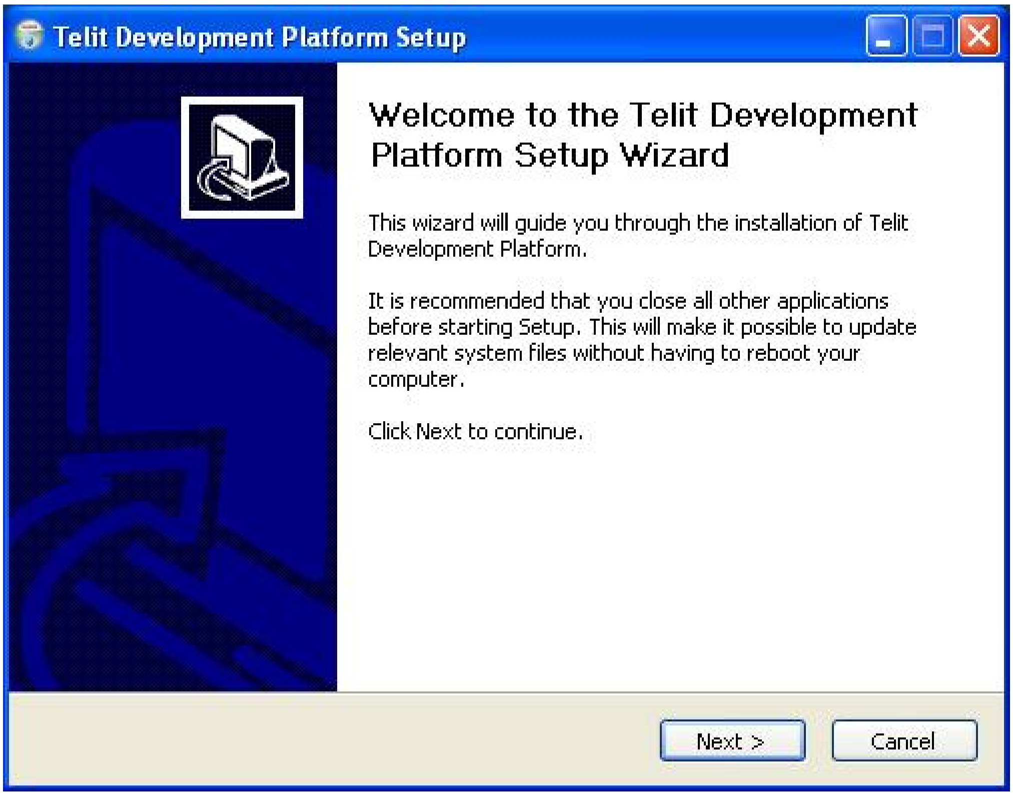
Note that the installation will create a C class network 192.168.121.0: if you already have in your system such a network you have to change manually, after the

installation, all the 192.168.121.x addresses to a free address space in your network. Refer to Appendix A for details.

Note that a new installation of the development environment will completely destroy a previous installation, if present. Refer to Appendix B to learn how to make a back-up.

3.3.1 Installation of the development environment

Run the installation of the development environment by clicking twice on the setupDevEnvironment_C_Cpp.exe file and follow the instructions on the screenshots:



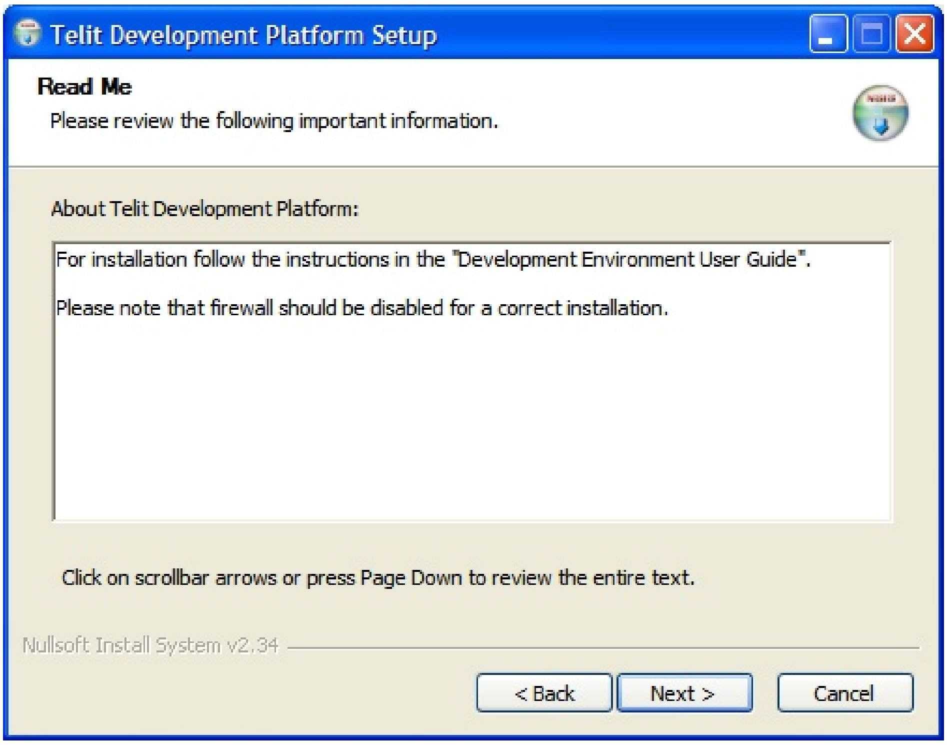
Welcome to the Telit Development Platform Setup Wizard

This wizard will guide you through the installation of Telit Development Platform.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click **Next**.



Telit Development Platform Setup

Read Me

Please review the following important information.

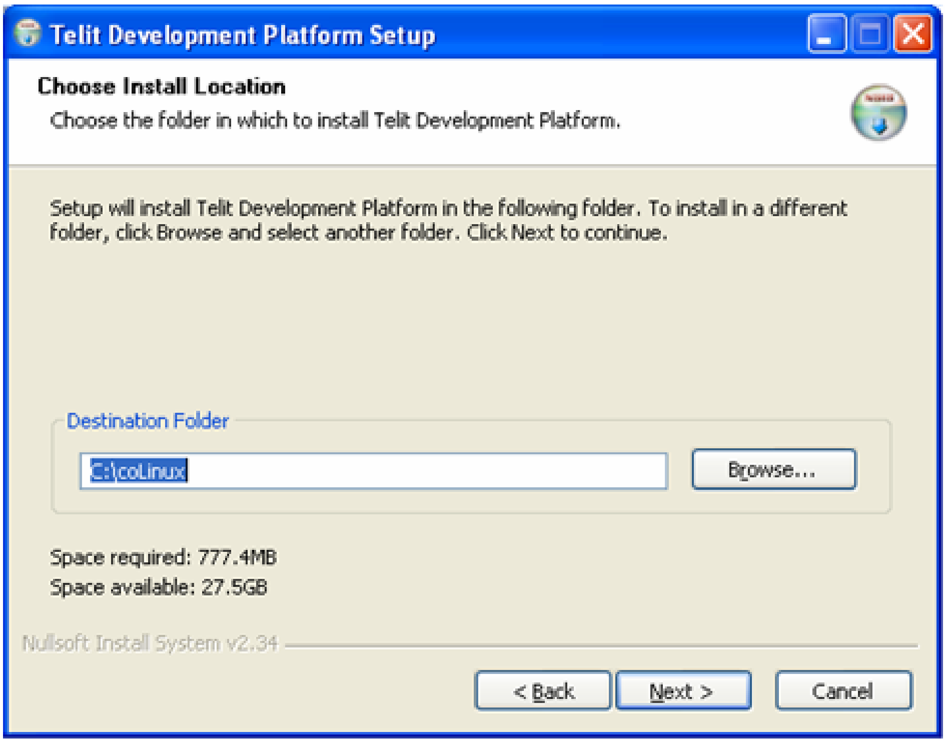
About Telit Development Platform:

For installation follow the instructions in the "Development Environment User Guide".

Please note that firewall should be disabled for a correct installation.

Click on scrollbar arrows or press Page Down to review the entire text.

Carefully read the information then click **Next**.



Telit Development Platform Setup

Choose Install Location

Choose the folder in which to install Telit Development Platform.

Setup will install Telit Development Platform in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\coLinux

Space required: 777.4MB
Space available: 27.5GB

Choose the directory, where Telit Development Platform should be installed, according to the following rule:

- The directory tree must end with the directory coLinux in such a way:

[device letter]:[path]\colinux

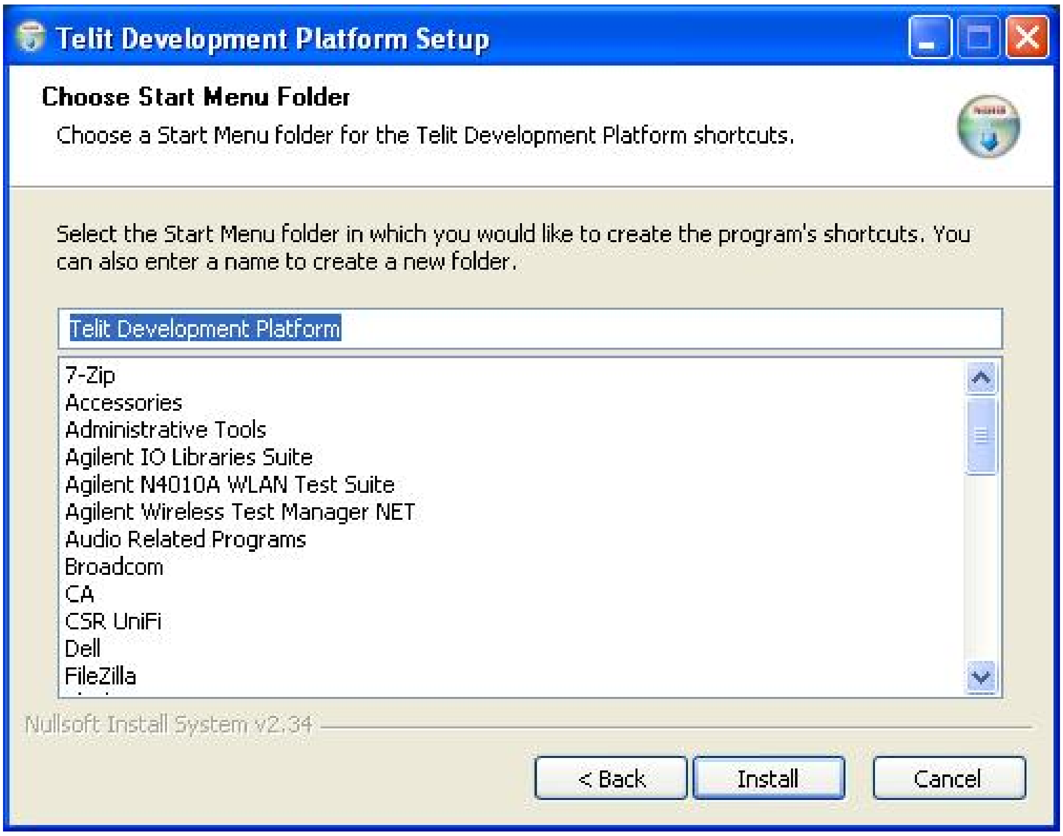
where [path] is a filesystem path preferably without spaces. Some examples are:

c:\colinux (PREFERRED)
c:\telit\colinux

Note that all the other packages should be installed inside this directory. For example, considering Xming:

c:\colinux\Xming
c:\telit\colinux\Xming

After choosing the directory click **Next**.


Telit Development Platform Setup

Choose Start Menu Folder

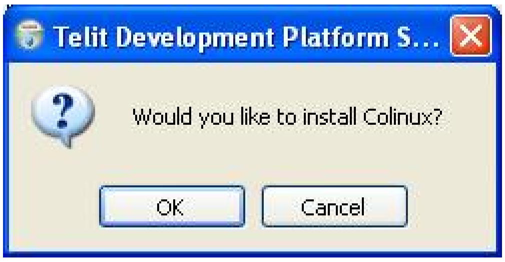
Choose a Start Menu folder for the Telit Development Platform shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

Telit Development Platform

7-Zip
Accessories
Administrative Tools
Agilent IO Libraries Suite
Agilent N4010A WLAN Test Suite
Agilent Wireless Test Manager NET
Audio Related Programs
Broadcom
CA
CSR UniFi
Dell
FileZilla

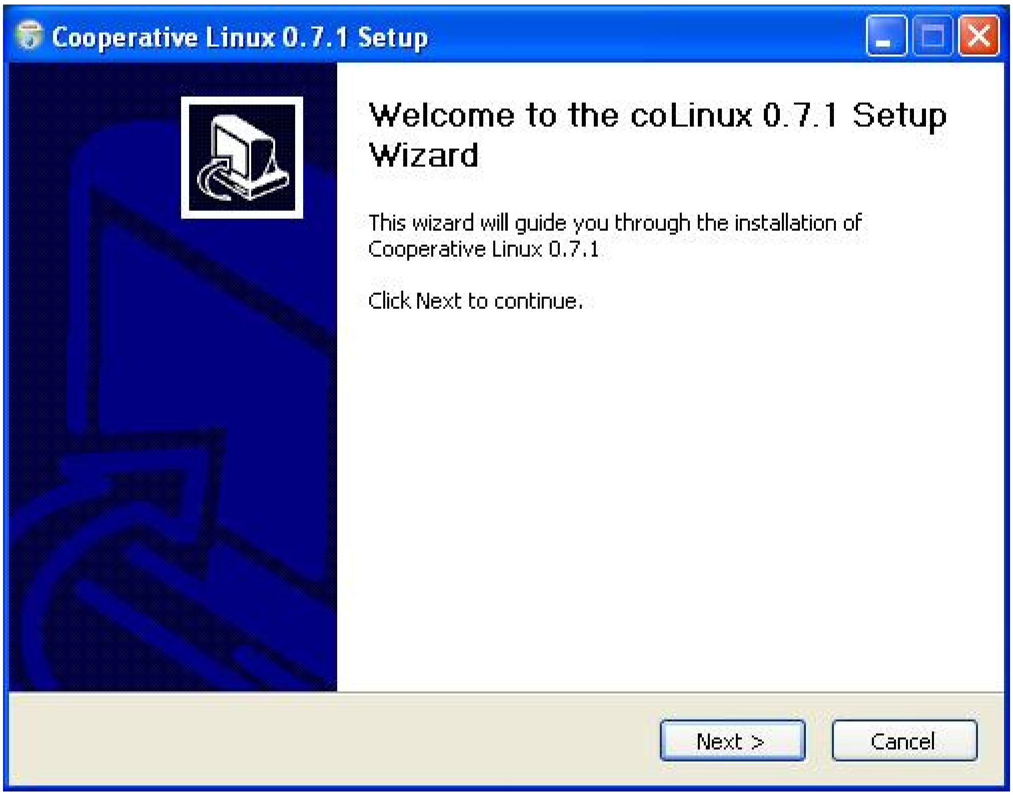
Click **Install**.



Would you like to install Colinux?

You will be asked to install coLinux: if you haven't previously installed it click **Ok**, otherwise click **Cancel** and skip to Xming installation.

If you have chosen to install coLinux follow the next screenshots:

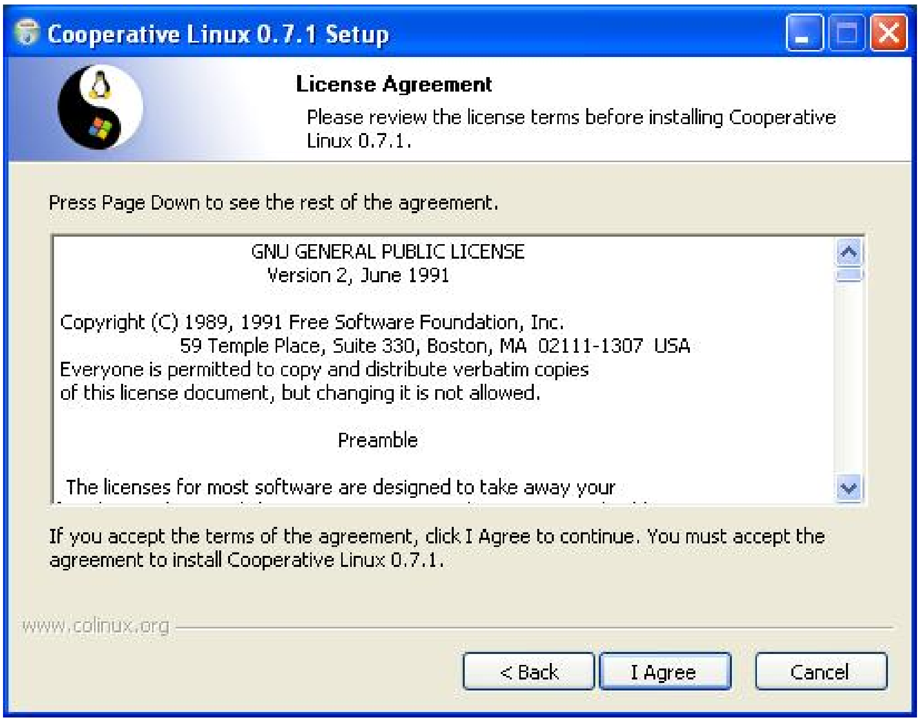


Welcome to the coLinux 0.7.1 Setup Wizard

This wizard will guide you through the installation of Cooperative Linux 0.7.1

Click Next to continue.

Click **Next**.



Cooperative Linux 0.7.1 Setup

License Agreement

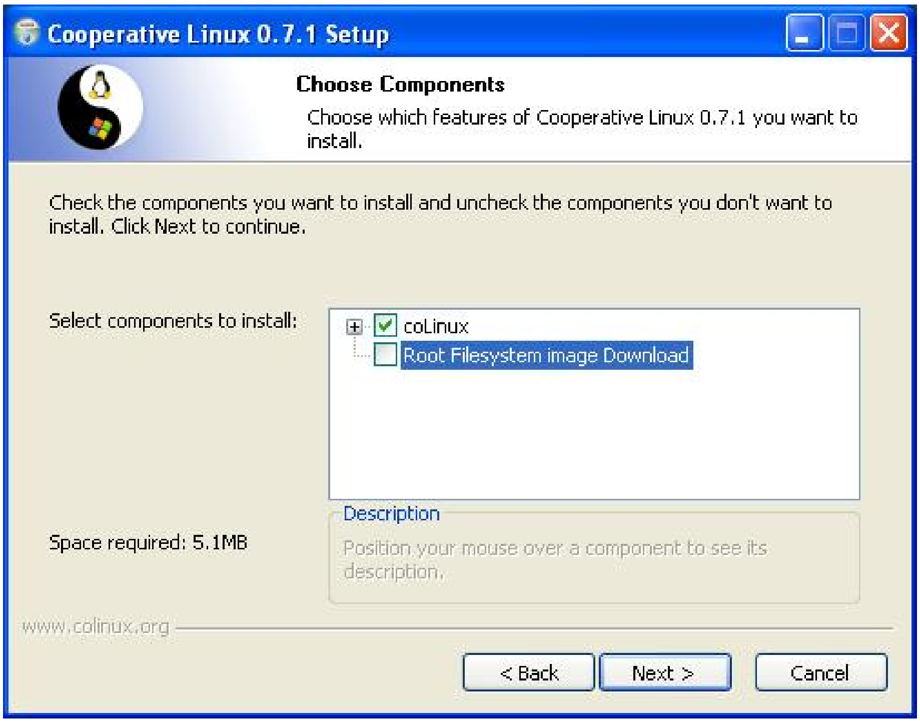
Please review the license terms before installing Cooperative Linux 0.7.1.

Press Page Down to see the rest of the agreement.

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
               59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

                            Preamble

 The licenses for most software are designed to take away your
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Cooperative Linux 0.7.1.

Click I **Agree**.



Cooperative Linux 0.7.1 Setup

Choose Components

Choose which features of Cooperative Linux 0.7.1 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ coLinux

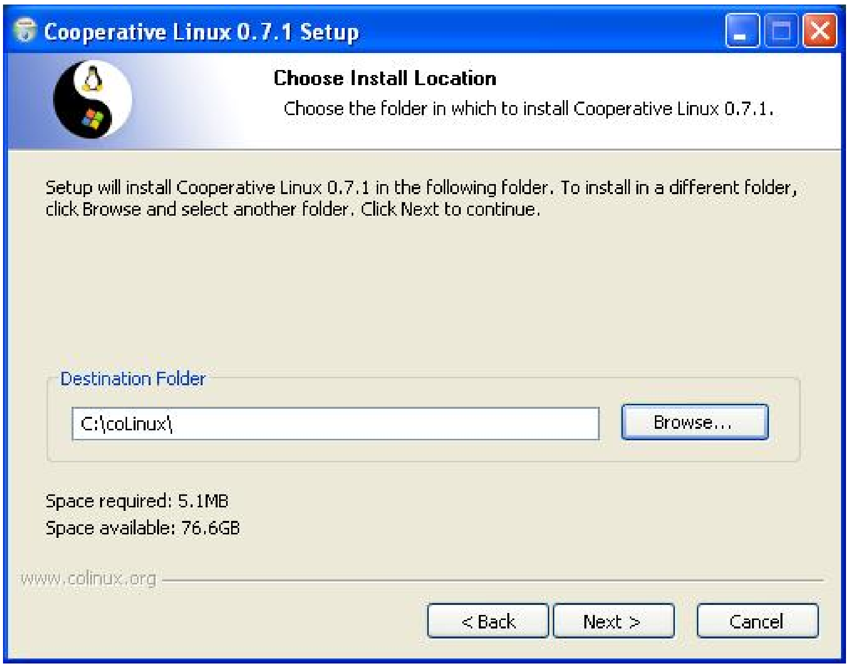
□ Root Filesystem image Download

Space required: 5.1MB

Description

Position your mouse over a component to see its description.

Uncheck the voice **Root Filesystem image Download** and click **Next**.



Cooperative Linux 0.7.1 Setup

Choose Install Location

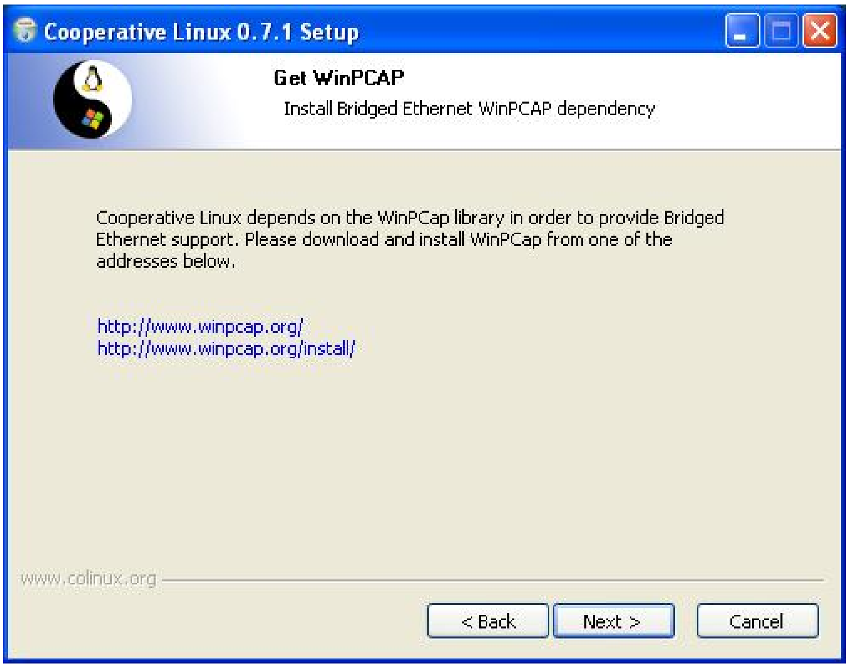
Choose the folder in which to install Cooperative Linux 0.7.1.

Setup will install Cooperative Linux 0.7.1 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\coLinux\

Space required: 5.1MB
Space available: 76.6GB

Select the previously chosen Telit Development Installation directory (for example c:\colinux or c:\telit\colinux).



Cooperative Linux 0.7.1 Setup

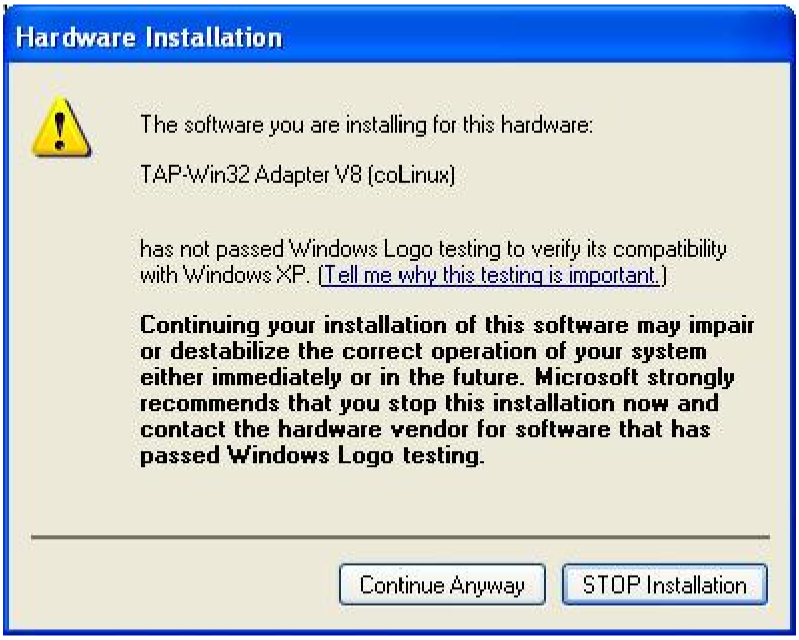
Get WinPCAP

Install Bridged Ethernet WinPCAP dependency

Cooperative Linux depends on the WinPCap library in order to provide Bridged Ethernet support. Please download and install WinPCap from one of the addresses below.

http://www.winpcap.org/
http://www.winpcap.org/install/

Click **Next**.



Hardware Installation

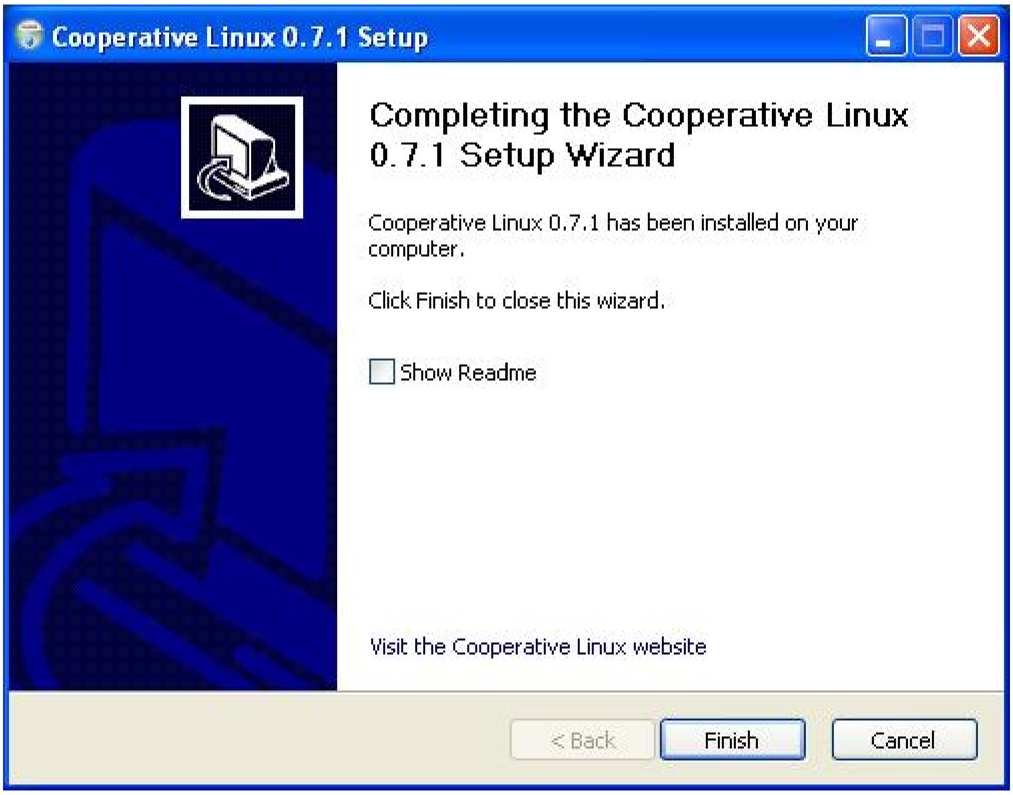
The software you are installing for this hardware:

TAP-Win32 Adapter V8 (coLinux)

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Click **Continue Anyway**.



Cooperative Linux 0.7.1 Setup

Completing the Cooperative Linux 0.7.1 Setup Wizard

Cooperative Linux 0.7.1 has been installed on your computer.

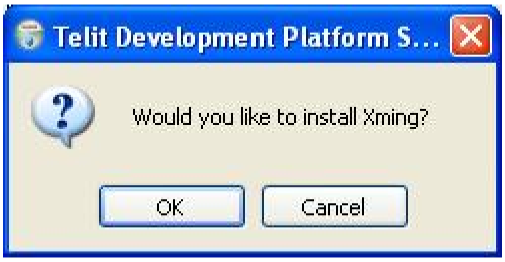
Click Finish to close this wizard.

□ Show Readme

Visit the Cooperative Linux website

Uncheck the voice **Show Readme** and click **Finish**.

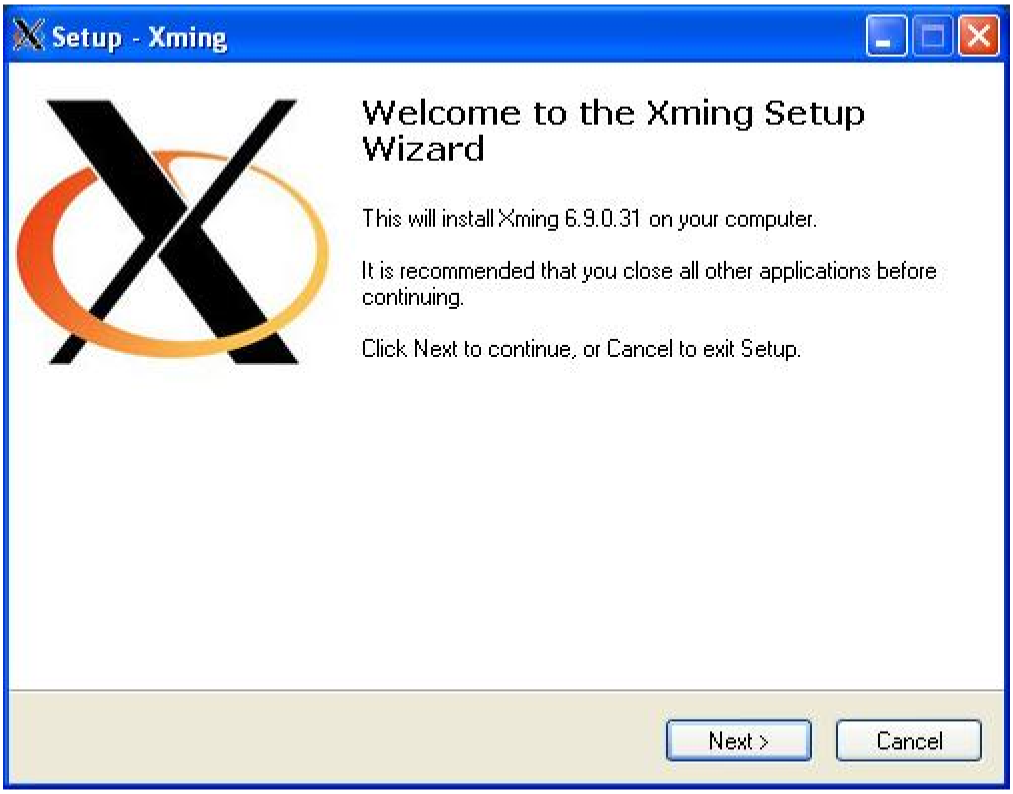
After installing coLinux (or choosing not to install it) you will be asked to install Xming.


Telit Development Platform S...

Would you like to install Xming?

If you haven't previously installed it click **Ok**, otherwise **Cancel** and skip to the file system image installation.

If you have chosen to install Xming follow the next screenshots:


Setup - Xming

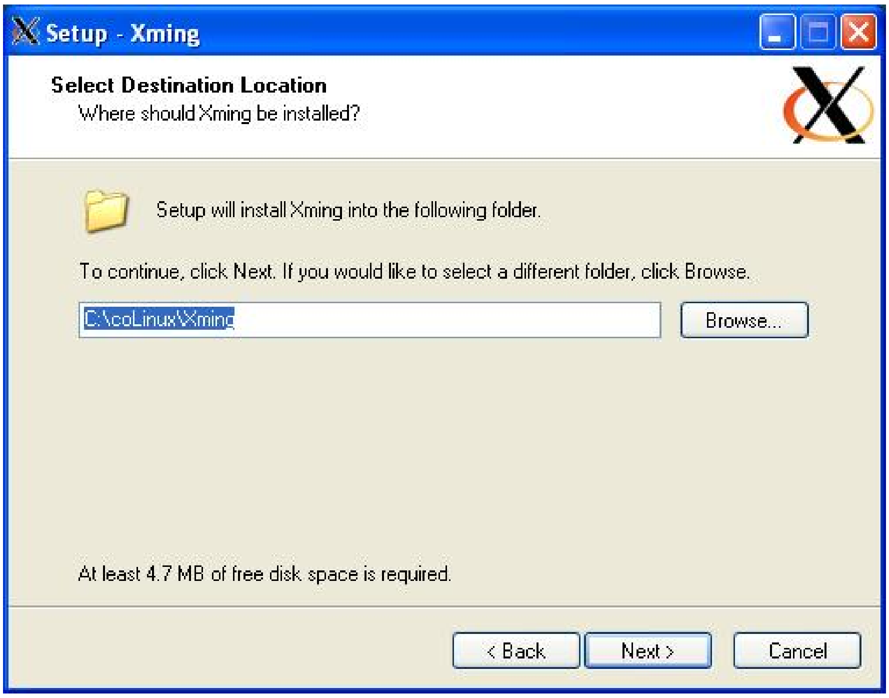
Welcome to the Xming Setup Wizard

This will install Xming 6.9.0.31 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Click **Next**.



Setup - Xming

Select Destination Location
Where should Xming be installed?

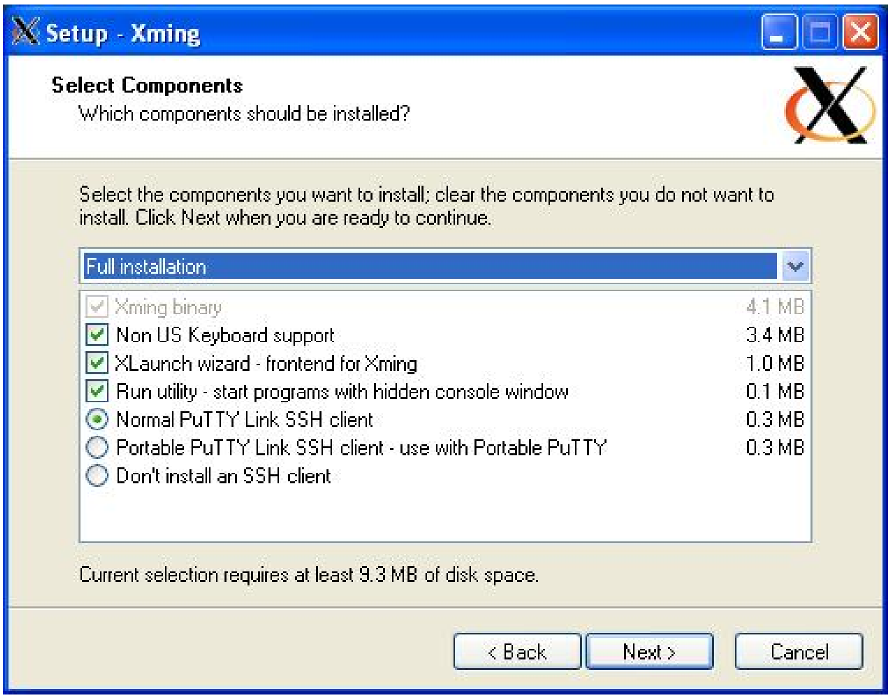
Setup will install Xming into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\coLinux\Xming

At least 4.7 MB of free disk space is required.

Click **Browse** to select the previously chosen Telit Development Platform installation directory. The installer will automatically add the path part for Xming (according to the previous examples the path will become c:\colinux\Xming or c:\telit\colinux\Xming).



Setup - Xming

Select Components
Which components should be installed?

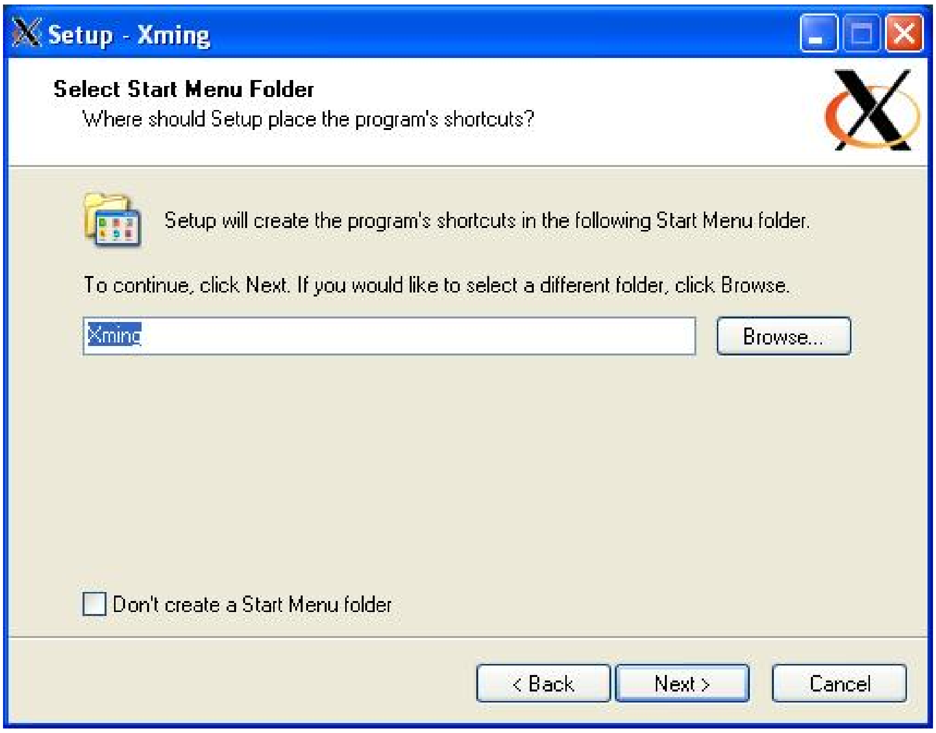
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

☑ Xming binary 4.1 MB
☑ Non US Keyboard support 3.4 MB
☑ XLaunch wizard - frontend for Xming 1.0 MB
☑ Run utility - start programs with hidden console window 0.1 MB
● Normal PuTTY Link SSH client 0.3 MB
○ Portable PuTTY Link SSH client - use with Portable PuTTY 0.3 MB
○ Don't install an SSH client

Current selection requires at least 9.3 MB of disk space.

Click **Next**.



Setup - Xming

Select Start Menu Folder

Where should Setup place the program's shortcuts?

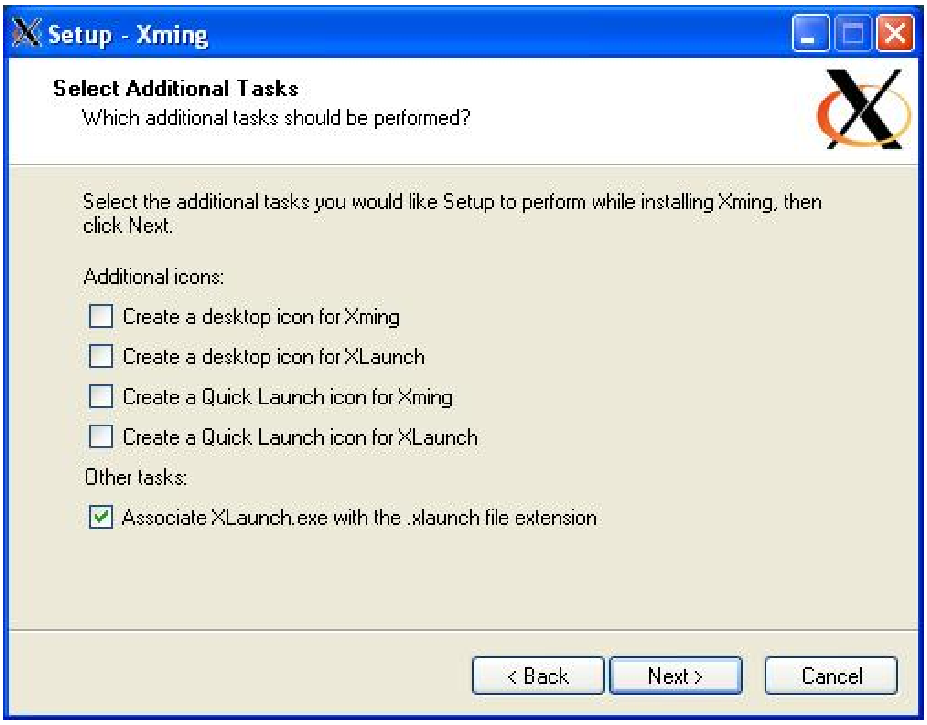
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Xming

□ Don't create a Start Menu folder

Click **Next**.



Setup - Xming

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Xming, then click Next.

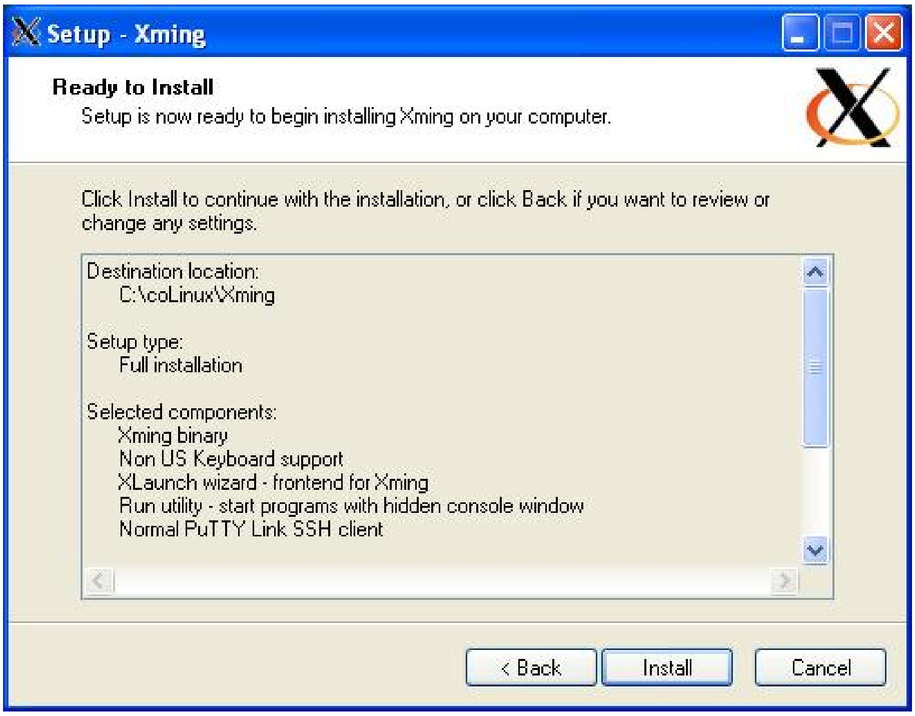
Additional icons:

□ Create a desktop icon for Xming
□ Create a desktop icon for XLaunch
□ Create a Quick Launch icon for Xming
□ Create a Quick Launch icon for XLaunch

Other tasks:

☑ Associate XLaunch.exe with the .xlaunch file extension

Click **Next**.



Setup - Xming

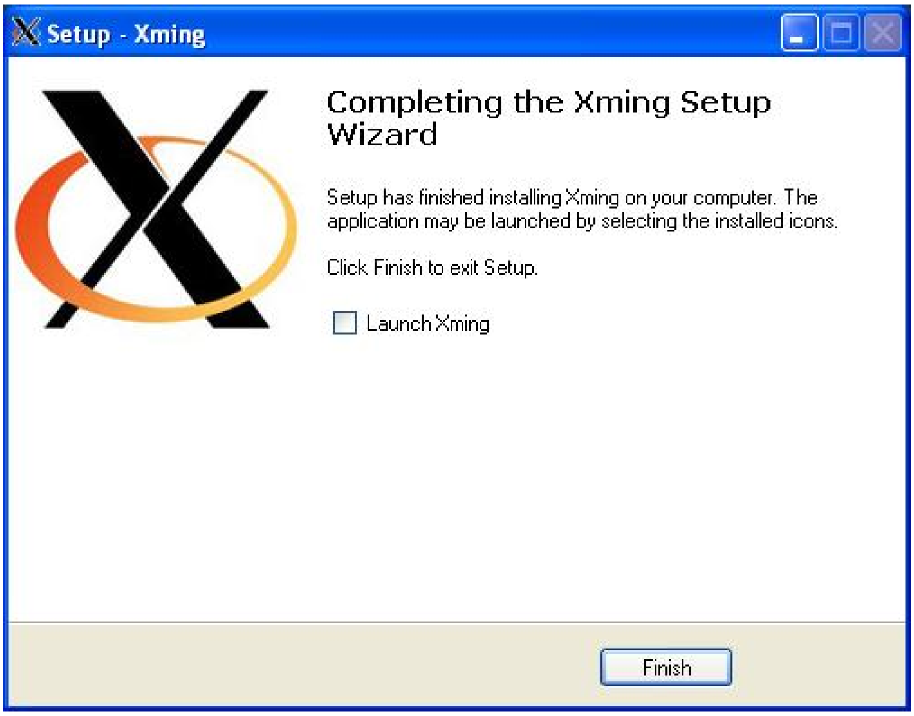
Ready to Install

Setup is now ready to begin installing Xming on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
      C:\coLinux\Xming

Setup type:
      Full installation

Selected components:
      Xming binary
      Non US Keyboard support
      XLaunch wizard - frontend for Xming
      Run utility - start programs with hidden console window
      Normal PuTTY Link SSH client
```

Click **Install**.



Setup - Xming

Completing the Xming Setup Wizard

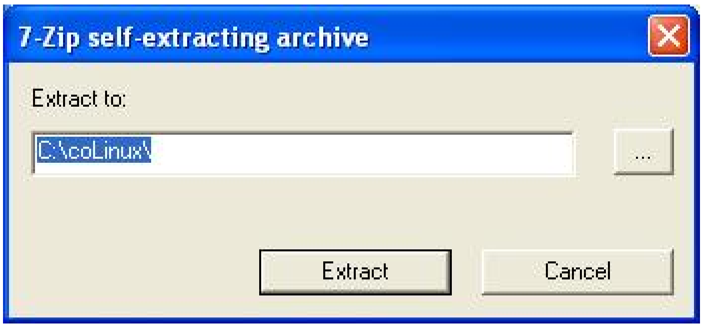
Setup has finished installing Xming on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

□ Launch Xming

Uncheck the voice **Launch Xming** and click **Finish**.

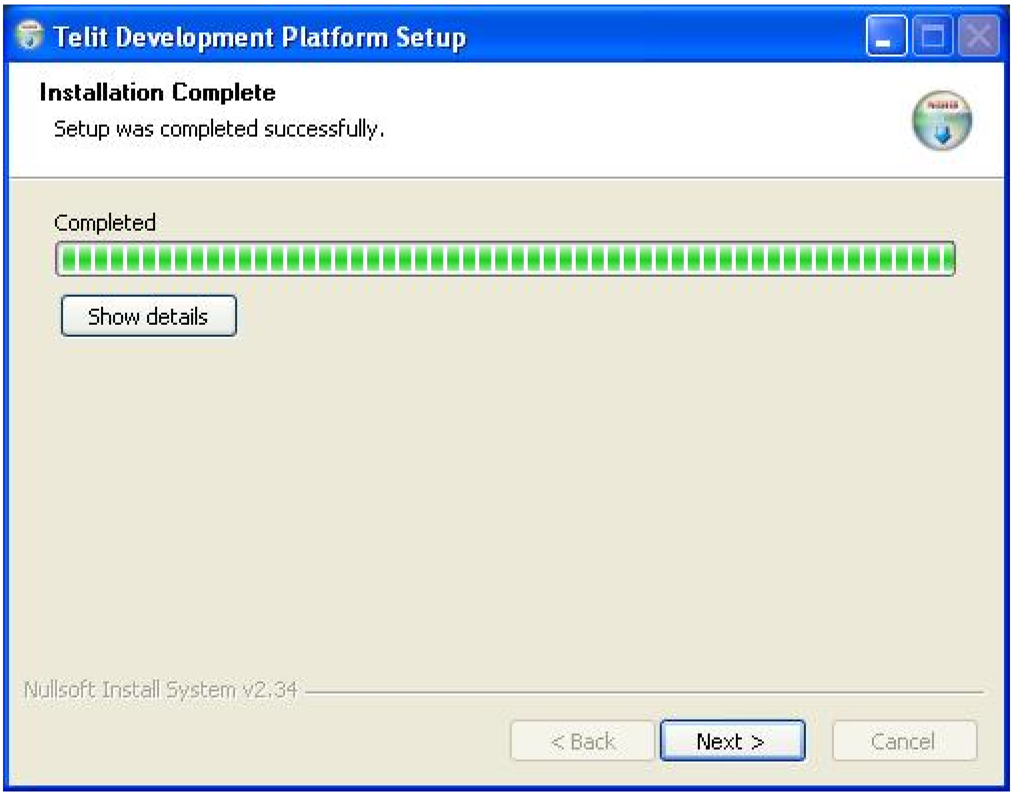
After installing Xming (or choosing not to install it) you will be prompted with the following window:


7-Zip self-extracting archive

Extract to:

C:\coLinux\

Select the previously chosen Telit Development Platform installation directory and click **Extract** (normally the directory is already correct and you have only to click Extract). Please note that this step can last several minutes.

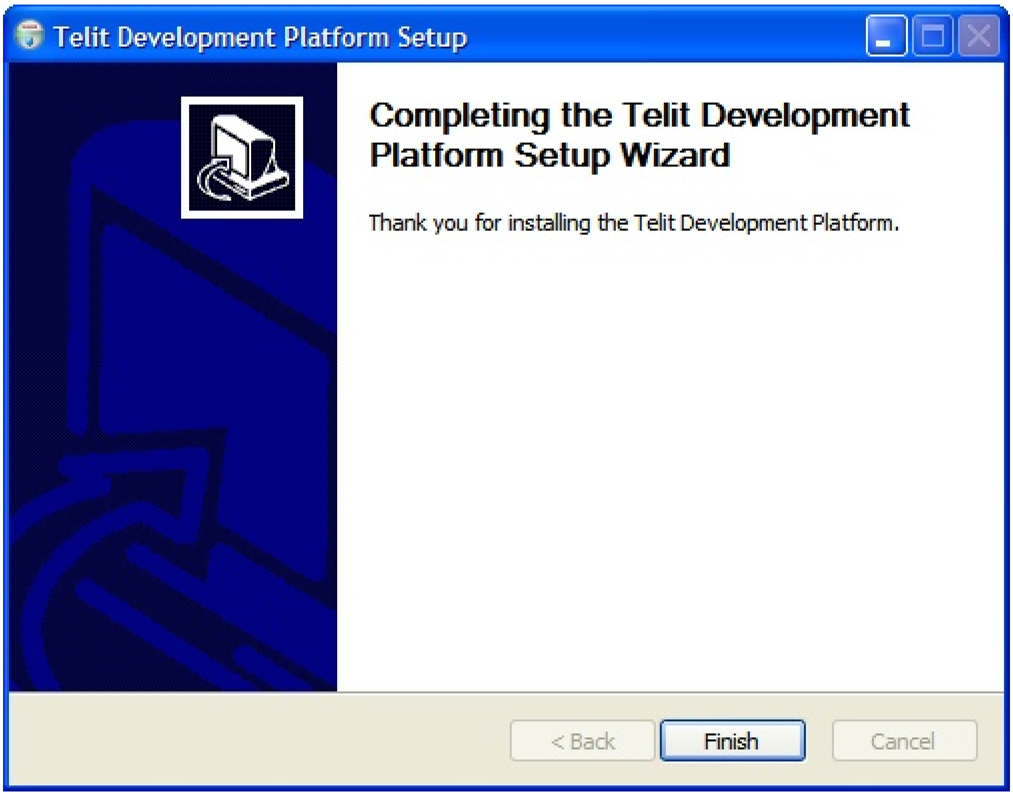

Telit Development Platform Setup

Installation Complete

Setup was completed successfully.

Completed

Click **Next**.



Telit Development Platform Setup

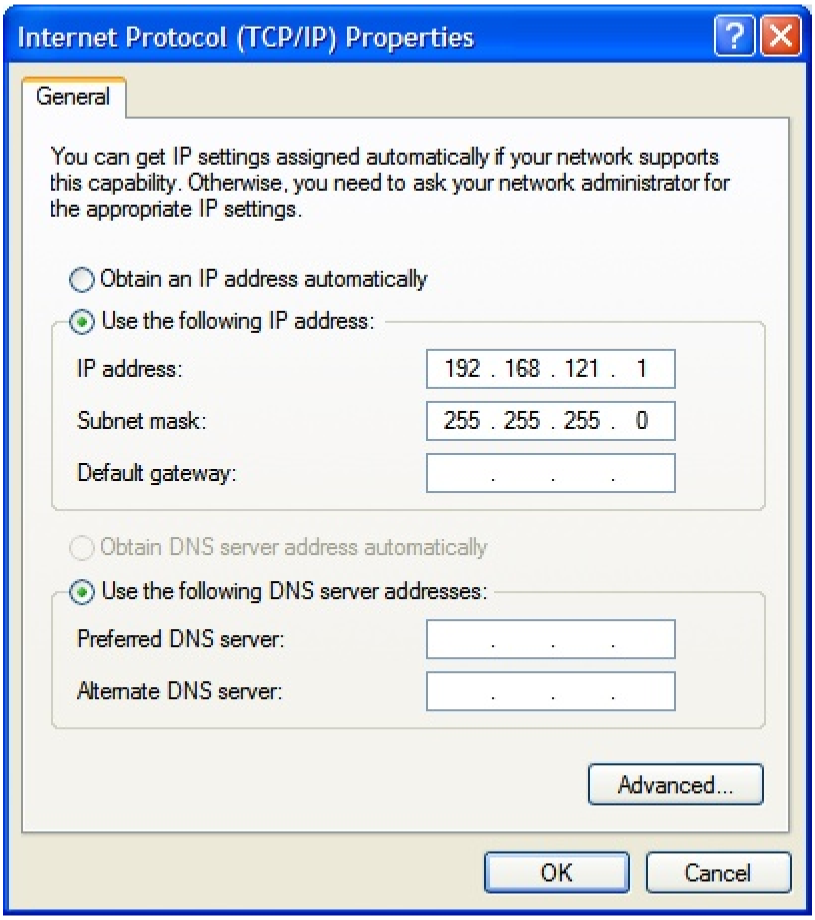
Completing the Telit Development Platform Setup Wizard

Thank you for installing the Telit Development Platform.

Click **Finish**.

As previously explained, if you already have in your system the 192.168.121.0 network it is time to change the IP addresses of the development environment: refer to Appendix A for further details.

Go to **Network Connections**, right click on the device called **TAP-Win32 Adapter V8** and choose the menu voice **Properties**. Select the **Internet Protocol (TCP/IP)** and click on the button properties; you should see the following window:



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 121 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

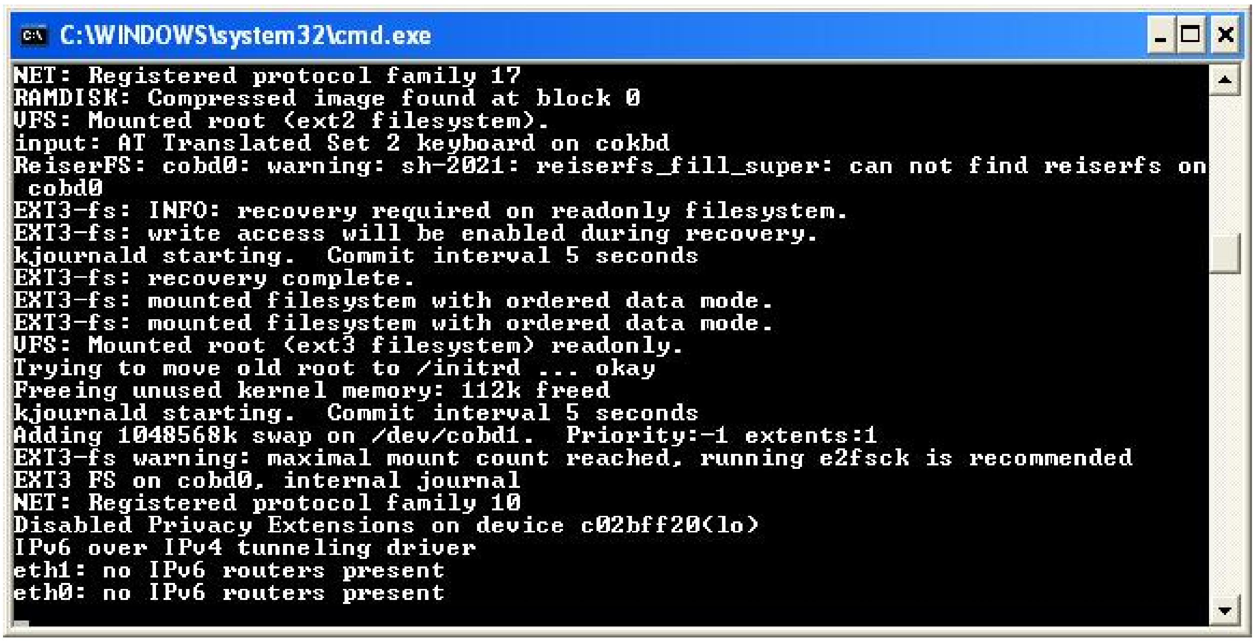
Alternate DNS server: . . .

If this configuration is not available, you have to manually insert the following values:

IP address: 192.168.121.1
Subnet mask: 255.255.255.0

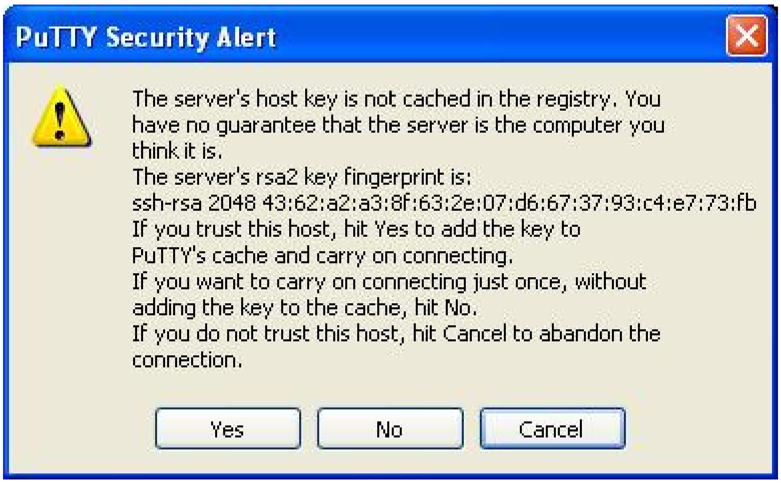
then click **Ok**.

Go to Start menu → All Programs → Telit Development Platform → Start Colinux.



```
NET: Registered protocol family 17
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
input: AT Translated Set 2 keyboard on cokbd
ReiserFS: cobd0: warning: sh-2021: reiserfs_fill_super: can not find reiserfs on
 cobd0
EXT3-fs: INFO: recovery required on readonly filesystem.
EXT3-fs: write access will be enabled during recovery.
kjournald starting.  Commit interval 5 seconds
EXT3-fs: recovery complete.
EXT3-fs: mounted filesystem with ordered data mode.
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem) readonly.
Trying to move old root to /initrd ... okay
Freeing unused kernel memory: 112k freed
kjournald starting.  Commit interval 5 seconds
Adding 1048568k swap on /dev/cobd1.  Priority:-1 extents:1
EXT3-fs warning: maximal mount count reached, running e2fsck is recommended
EXT3 FS on cobd0, internal journal
NET: Registered protocol family 10
Disabled Privacy Extensions on device c02bff20(lo)
IPv6 over IPv4 tunneling driver
eth1: no IPv6 routers present
eth0: no IPv6 routers present
```

After loading, go to Start menu → All Programs → Telit Development Platform → Telit Customized Eclipse. The first time you launch the application you will be prompted with the following window:



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 43:62:a2:a3:8f:63:2e:07:d6:67:37:93:c4:e7:73:fb
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

Click **Yes**.

Congratulations, you have installed Telit Development Platform.

3.3.2 Starting coLinux

Before using any component of the development environment you have to start coLinux going to Start menu → All Programs → Telit Development Platform → Start Colinux.

4 Connecting the target to the host

To get the most out of the development environment it is necessary to create an IP connection through the target and the host. There are two options: using a normal Ethernet connection via RJ45 port or using an Ethernet on USB connection via USB port. The first step is to configure the target.

Please note that if you have executed the procedure in Appendix A, in the following paragraphs you have to consider the new IP addresses values instead of the written ones.

4.1.1 Configuring the target: connection via RJ45

Turn on the target, connect the Ethernet cable and open a terminal; after the boot type:

```
modprobe macb
ifconfig eth0 hw ether AA:BB:AA:BB:AA:BB
ifconfig eth0 192.168.121.3 netmask 255.255.255.0
```

to configure the network interface; then, type:

```
telnetd
```

to start the telnet daemon.

For further details about Ethernet configuration refer to document [3].

Note that you have to repeat these configuration steps each time the target is started. It is possible to make the process automatic: refer to 4.1.4 for details.

4.1.2 Configuring the target: connection via USB

Turn on the target and open a terminal; after the boot type:

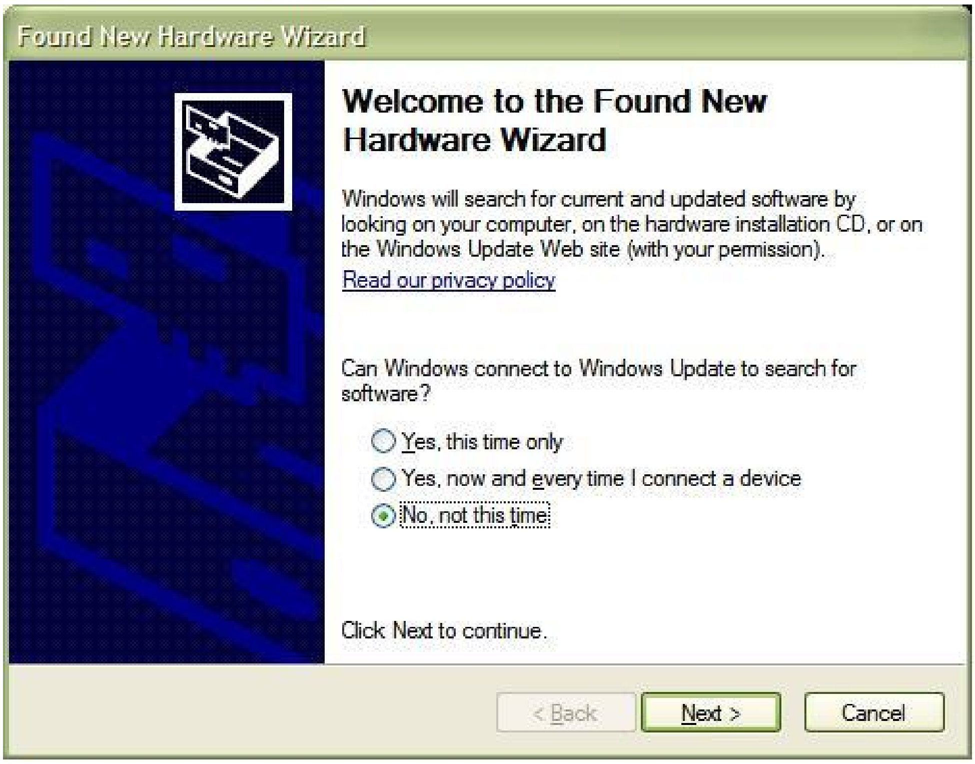
```
modprobe g_ether
```

then connect the USB cable.

If this is the first time you use the connection via USB you have to configure Windows.

Note that Windows driver installation is needed only if it is the first time you use the USB connection or if you change USB port. Otherwise you can skip over this part and go directly to the IP configuration with `ifconfig` (page 31).

Plug the USB cable into the target’s USB Device port (B-type) and into the PC (A-type): you will be asked for a driver "Linux USB Ethernet/RNDIS Gadget". Follow the actions described by the screenshots:



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

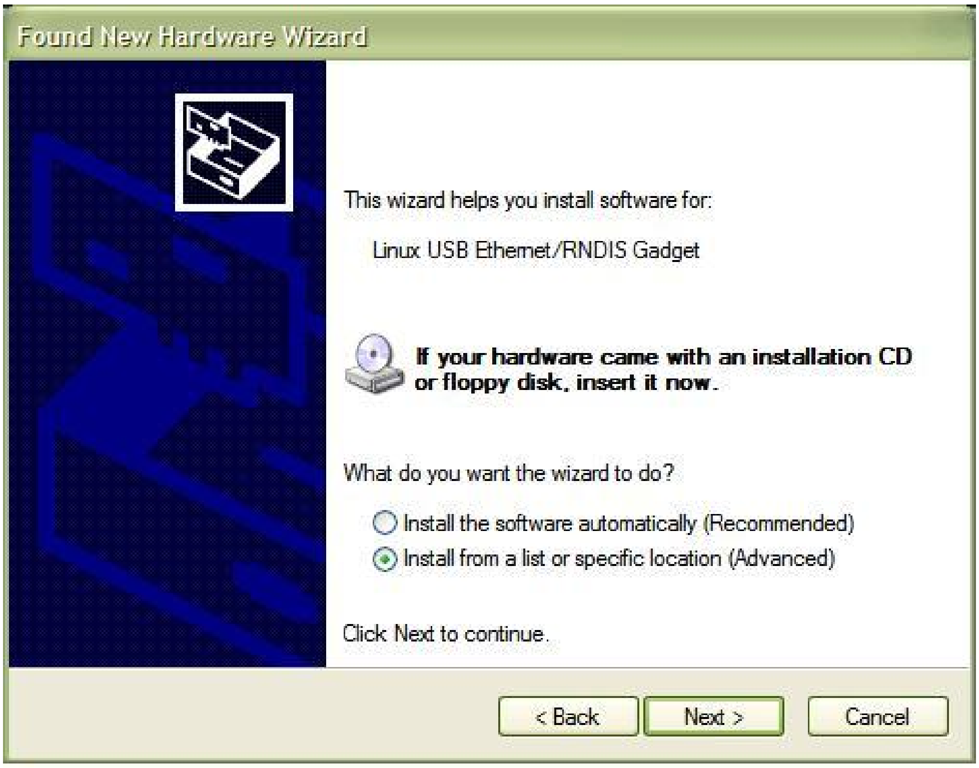
Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.



Found New Hardware Wizard

This wizard helps you install software for:

Linux USB Ethernet/RNDIS Gadget

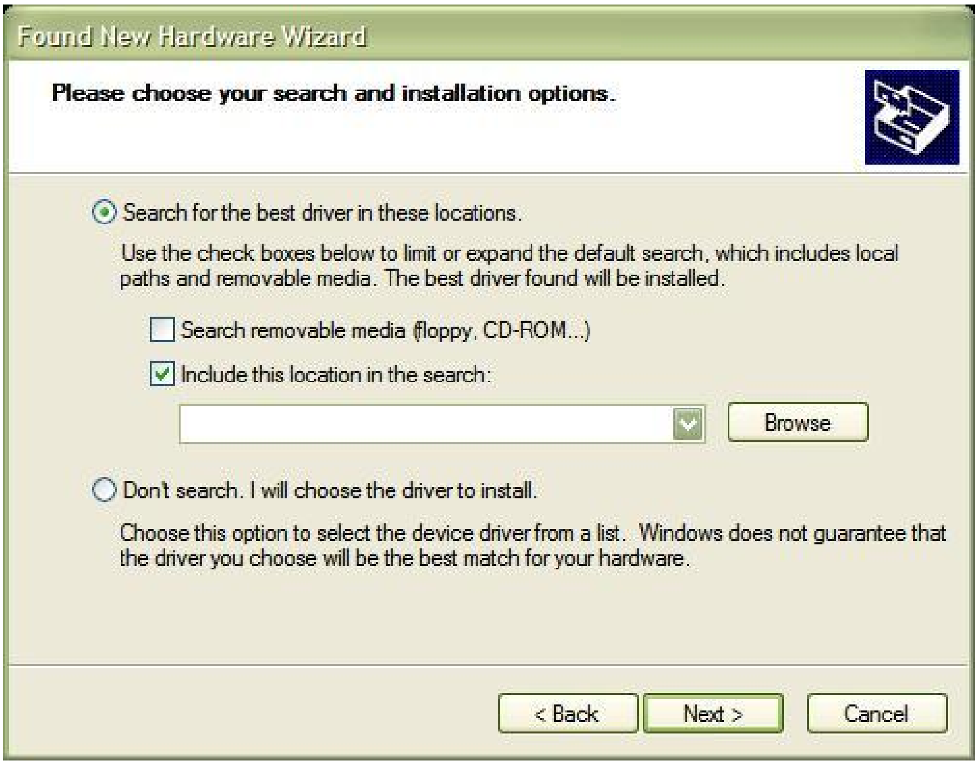
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)

● Install from a list or specific location (Advanced)

Click Next to continue.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

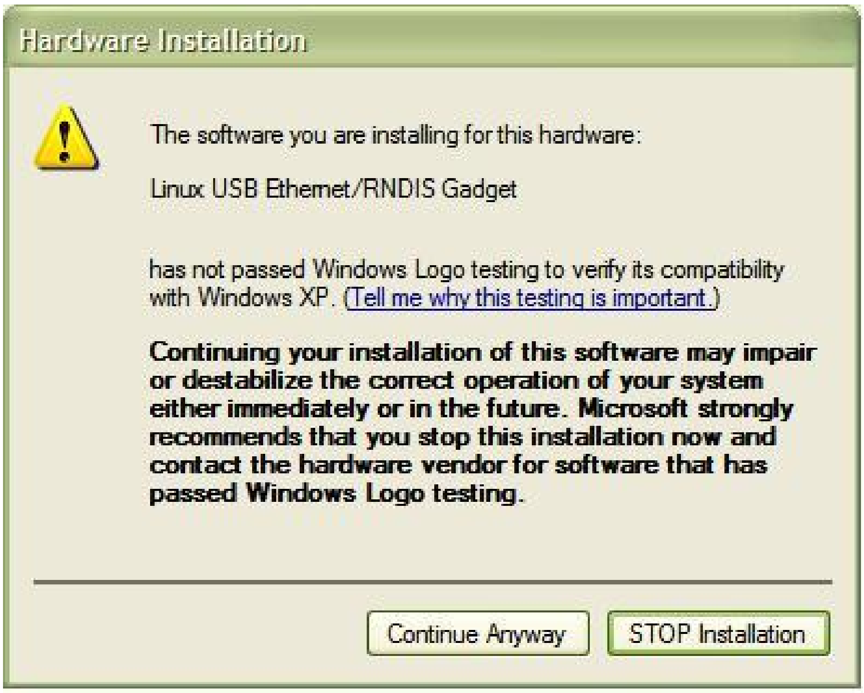
□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Click the **Browse** button and search for the file linux.inf, given with Telit package inside coLinux installation directory.



Hardware Installation

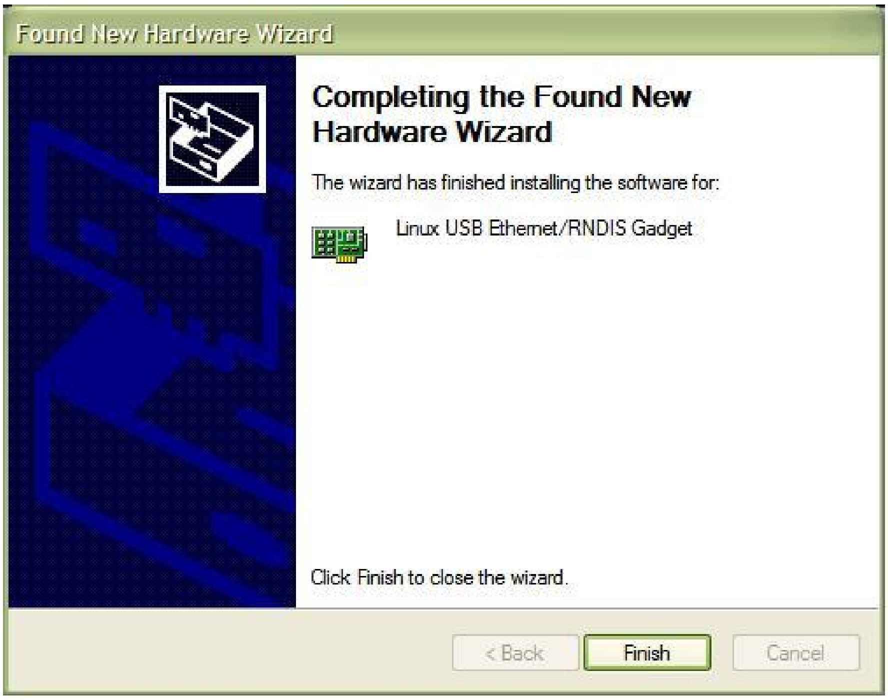
The software you are installing for this hardware:

Linux USB Ethernet/RNDIS Gadget

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Click **Continue Anyway**.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

Linux USB Ethernet/RNDIS Gadget

Click Finish to close the wizard.

Now, type in the terminal:

```
ifconfig usb0 192.168.121.3 netmask 255.255.255.0
```

to configure the IP address. Then, type:

```
telnetd
```

to start the telnet daemon.

For further details about Ethernet on USB configuration refer to document [3].

Note that you have to repeat these configuration steps (excluding the Windows driver installation) each time the target is started. It is possible to make the process automatic: refer to 4.1.4 for details.

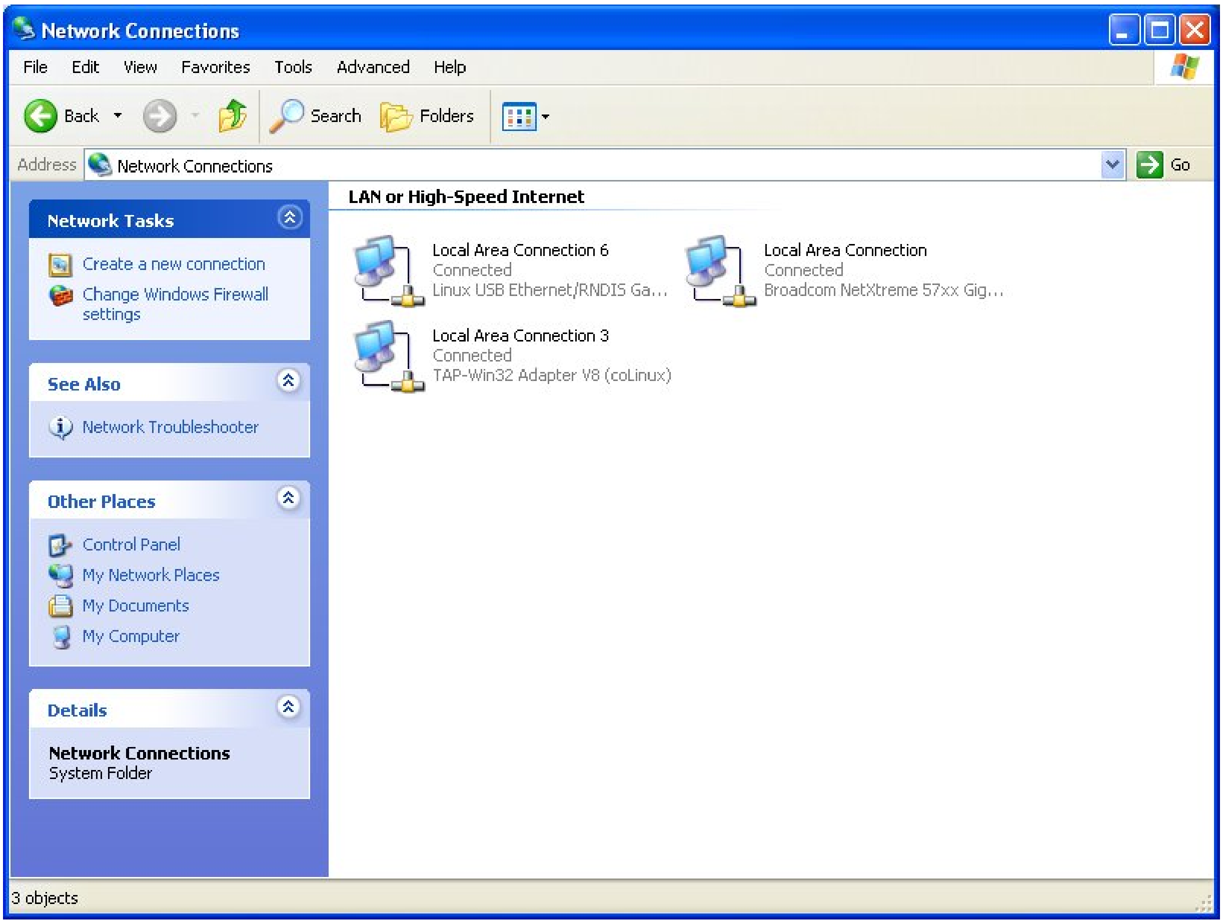
Note that after a reboot of the target you have to un-plug and plug the USB cable to make the connection work.

4.1.3 Configuring the host Ethernet connection

Close all the applications opened in the development environment and coLinux itself. Go to My Network Places → View Network Connections and identify the following two connections:

- The virtual connection used for coLinux whose Device Name is "TAP-Win32 Adapter V8 (coLinux)".
- The connection used for the target (it could be a normal Ethernet link or the Ethernet on USB link).

You can find an example on the image below:



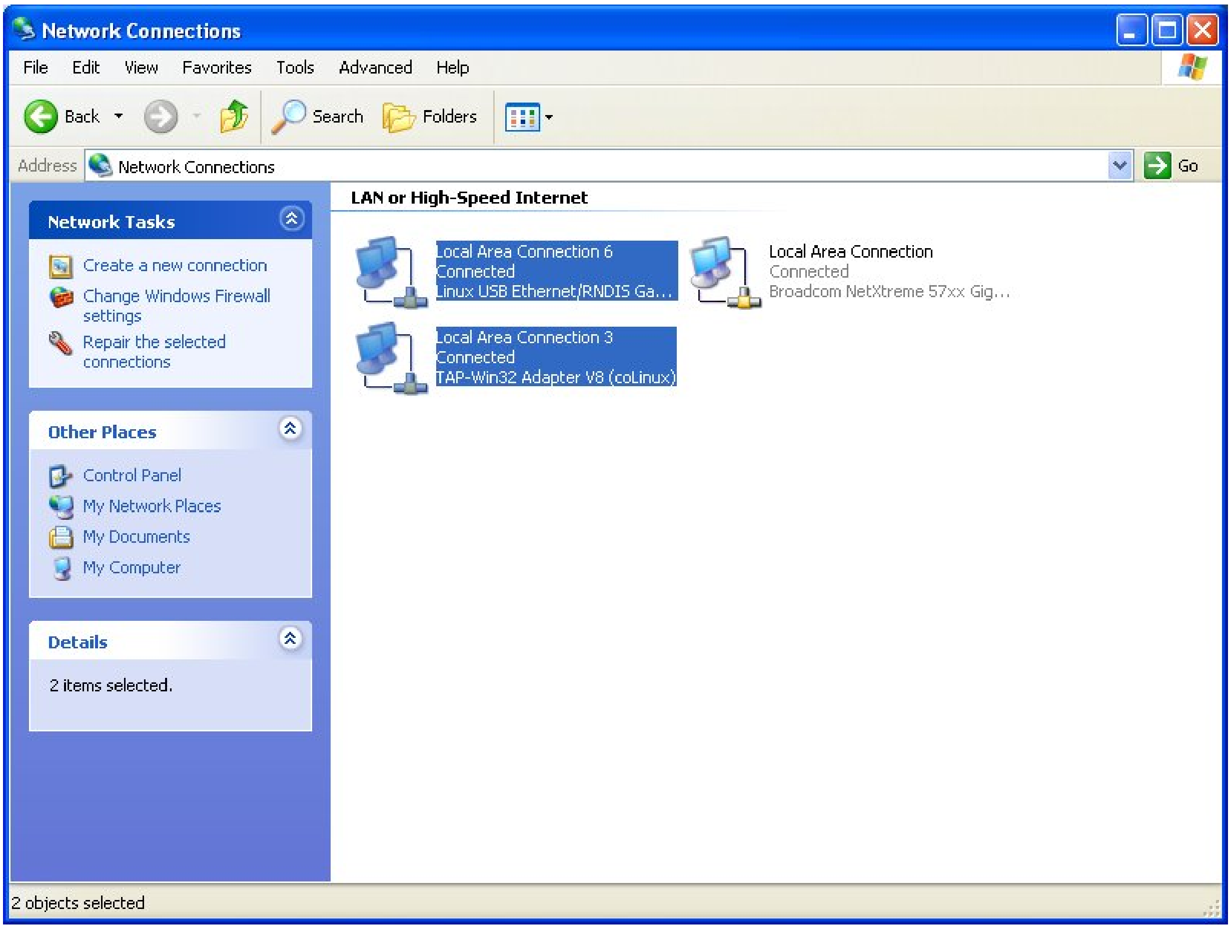
LAN or High-Speed Internet

Local Area Connection 6
Connected
Linux USB Ethernet/RNDIS Ga...

Local Area Connection
Connected
Broadcom NetXtreme 57xx Gig...

Local Area Connection 3
Connected
TAP-Win32 Adapter V8 (coLinux)

Select the two connections.



2 items selected.

LAN or High-Speed Internet

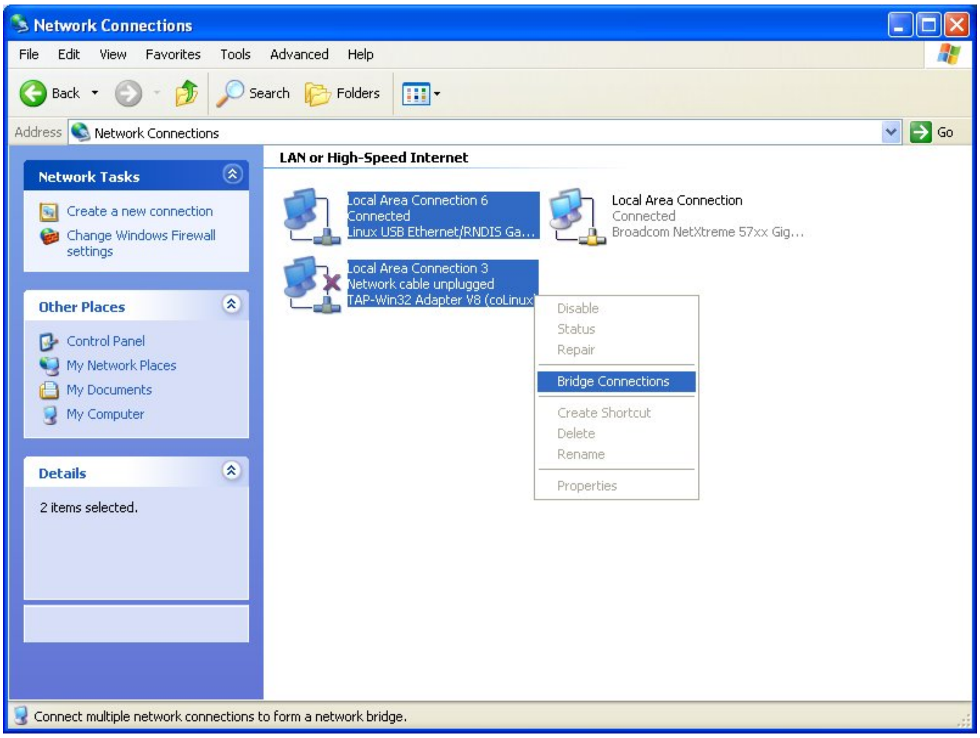
Local Area Connection 6
Connected
Linux USB Ethernet/RNDIS Ga...

Local Area Connection
Connected
Broadcom NetXtreme 57xx Gig...

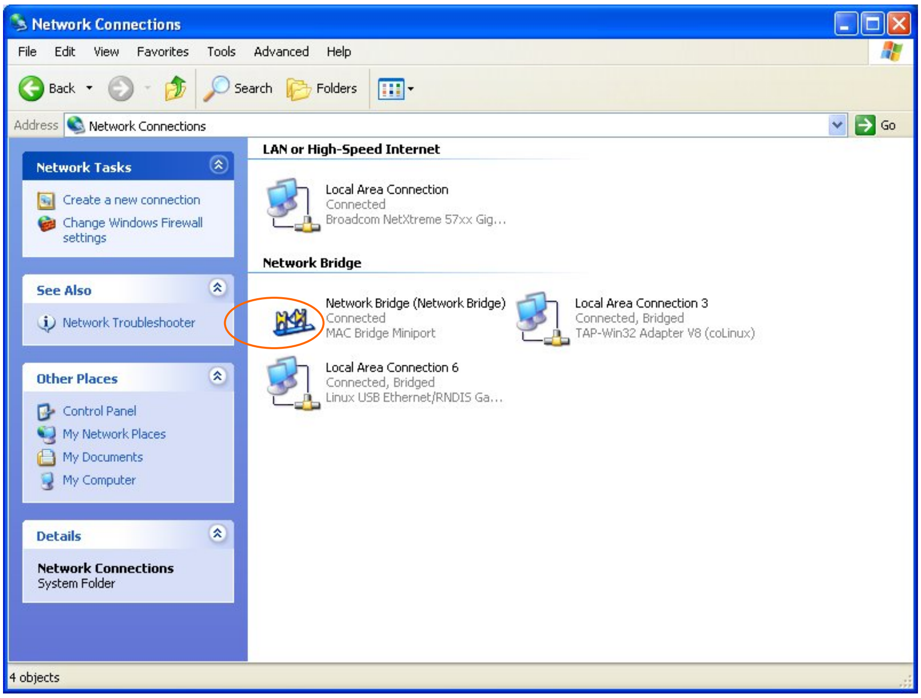
Local Area Connection 3
Connected
TAP-Win32 Adapter V8 (coLinux)

2 objects selected

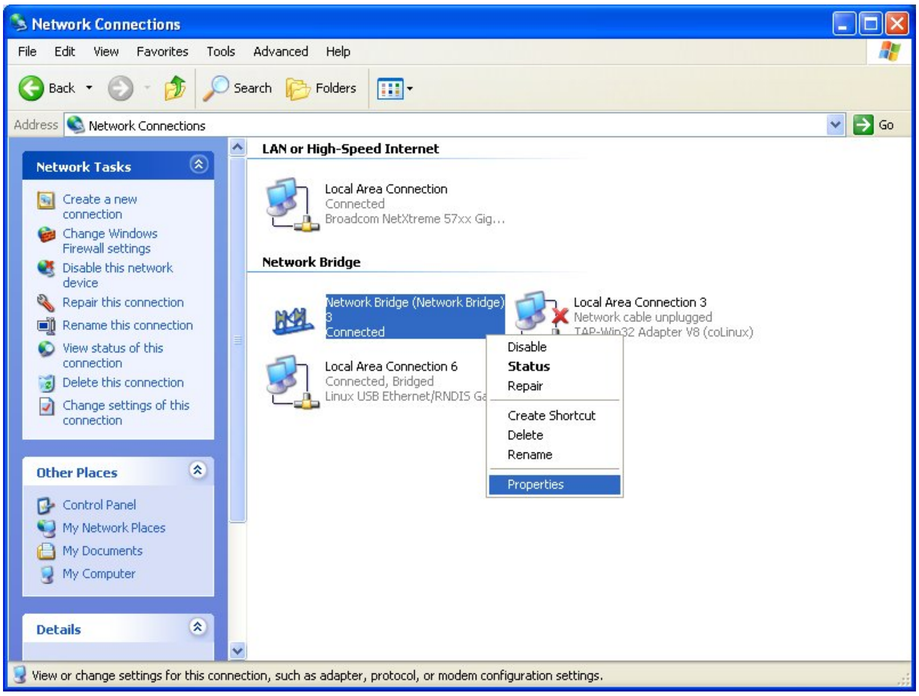
Once selected, right click and choose the voice **Bridge Connections**.



If the operation is successful you should see a new network bridge like pictured in the following image (if the bridge doesn’t show up reboot your Windows system):

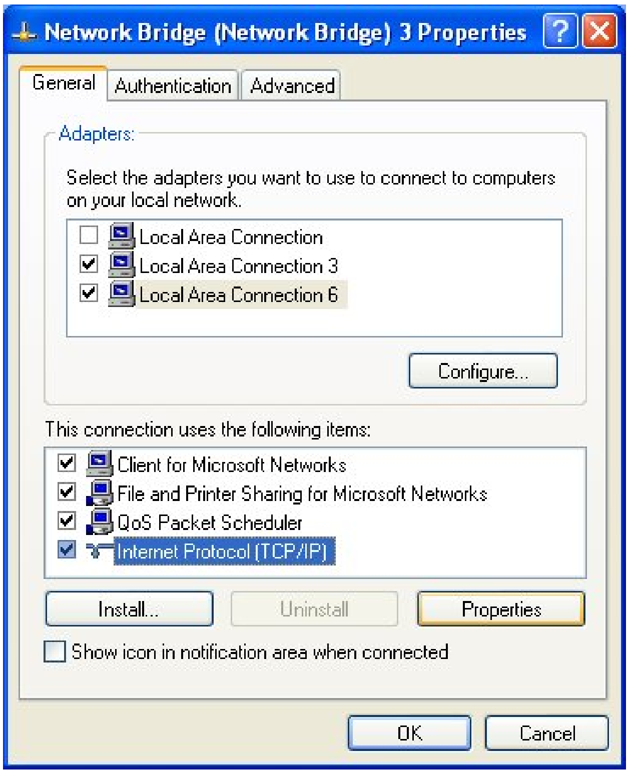


Right click on the Network Bridge and choose the voice **Properties**



View or change settings for this connection, such as adapter, protocol, or modem configuration settings.

Select the voice Internet Protocol (TCP/IP) and click the button **Properties**.


Network Bridge (Network Bridge) 3 Properties
General Authentication Advanced
Adapters:
Configure...
Install... Uninstall Properties
OK Cancel

Select the adapters you want to use to connect to computers on your local network.

- □ Local Area Connection
- ☑ Local Area Connection 3
- ☑ Local Area Connection 6

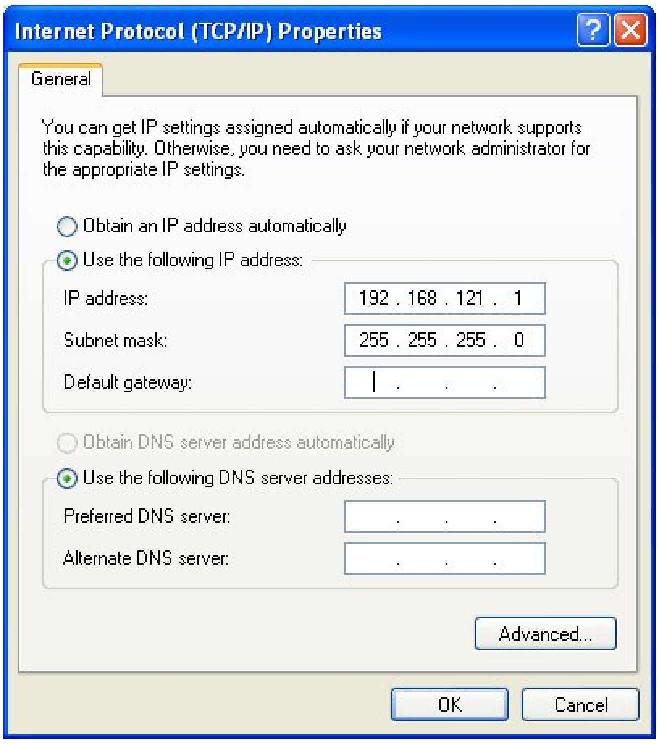
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol (TCP/IP)

□ Show icon in notification area when connected

Check the radio button **Use the following IP address** and insert these data:

- IP address: 192.168.121.1
- Subnet mask: 255.255.255.0



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 121 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Click the **Ok** button, then the **Close** button.

The host configuration is complete. To test its working, start coLinux and open a windows shell; type:

ping 192.168.121.2

to check for the connection to coLinux. Type:

ping 192.168.121.3

to check for the connection to the target.

Summarizing network configuration after default installation:

Windows IP address: 192.168.121.1
coLinux IP address: 192.168.121.2
GE863-PRO^3 IP address: 192.168.121.3

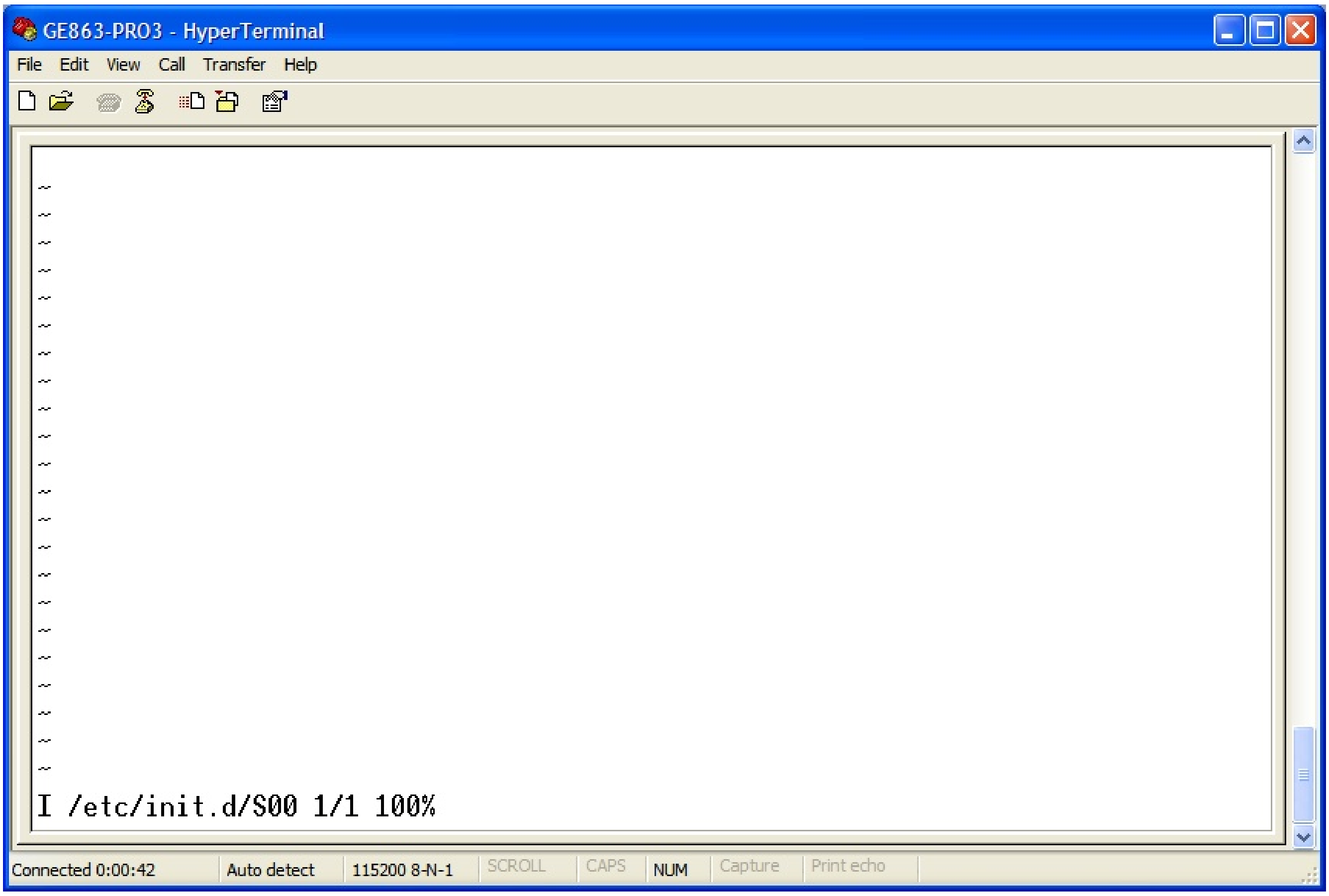
4.1.4 Making the target IP configuration automatic

In order to avoid manual IP configuration of the target each time it is rebooted you can follow these steps:

- Type in the terminal:

```
vi /etc/init.d/S00
```

- At vi prompt, press the ‘i’ key on the keyboard to enter in edit mode.



```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
I /etc/init.d/S00 1/1 100%
```

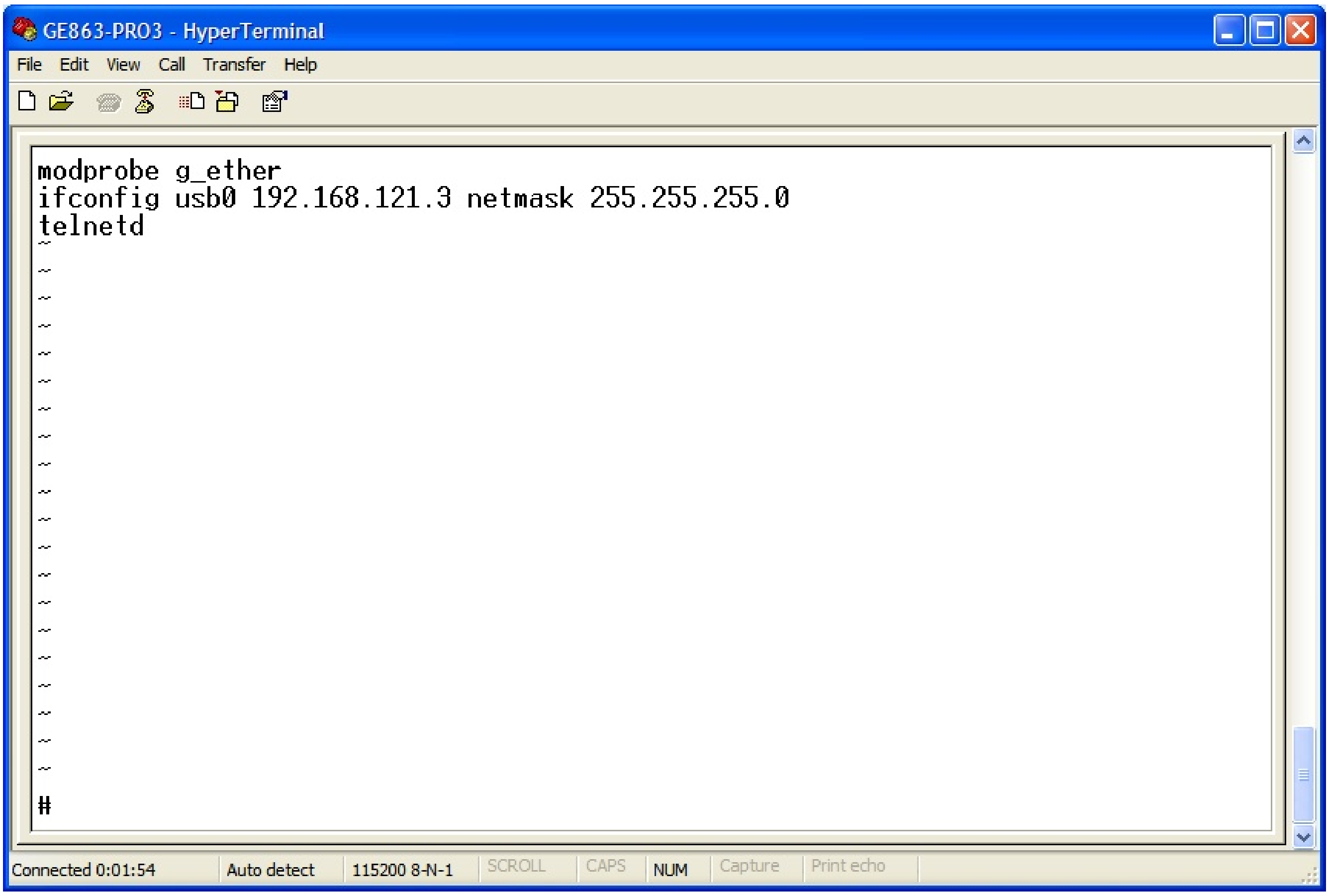
- For Ethernet RJ45 connection type:

```
modprobe macb
ifconfig eth0 hw ether AA:BB:AA:BB:AA:BB
ifconfig eth0 192.168.121.3 netmask 255.255.255.0
telnetd
```

- For Ethernet on USB connection type:

```
modprobe g_ether
ifconfig usb0 192.168.121.3 netmask 255.255.255.0
telnetd
```

- Press the ‘Esc’ key, then press the following keys on the keyboard in succession: ‘:wq’ and ‘Enter’. You should see again the shell prompt



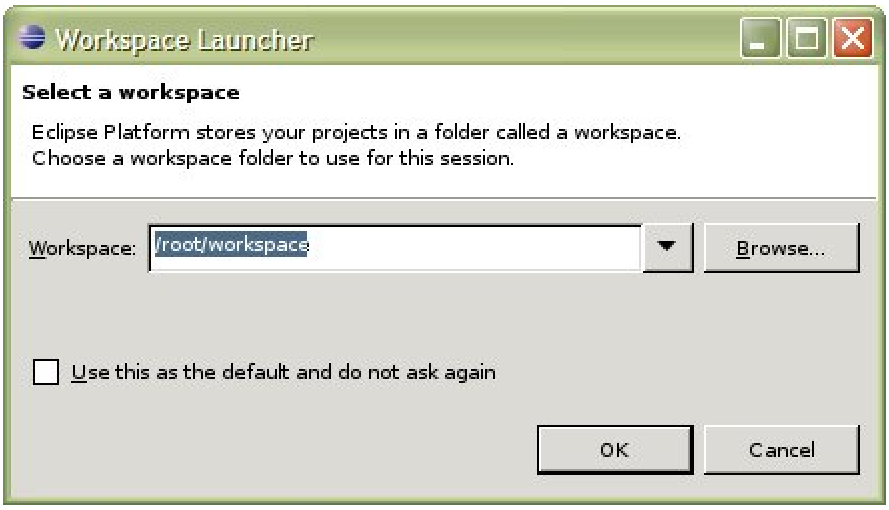
```
modprobe g_ether
ifconfig usb0 192.168.121.3 netmask 255.255.255.0
telnetd
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
#
```

After reboot the IP configuration should be working.

5 Starting Eclipse

To start the IDE follow these steps:

- Be sure that Colinux has been started (to start it go to Start menu → All Programs → Telit Development Platform → Start Colinux).
- After loading, go to Start menu → All Programs → Telit Development Platform → Telit Customized Eclipse.
- You will be prompted to choose a location in the filesystem for creating the workspace. A path without spaces is recommended. Note that you can find your Windows filesystem in `/mnt/windows/`.



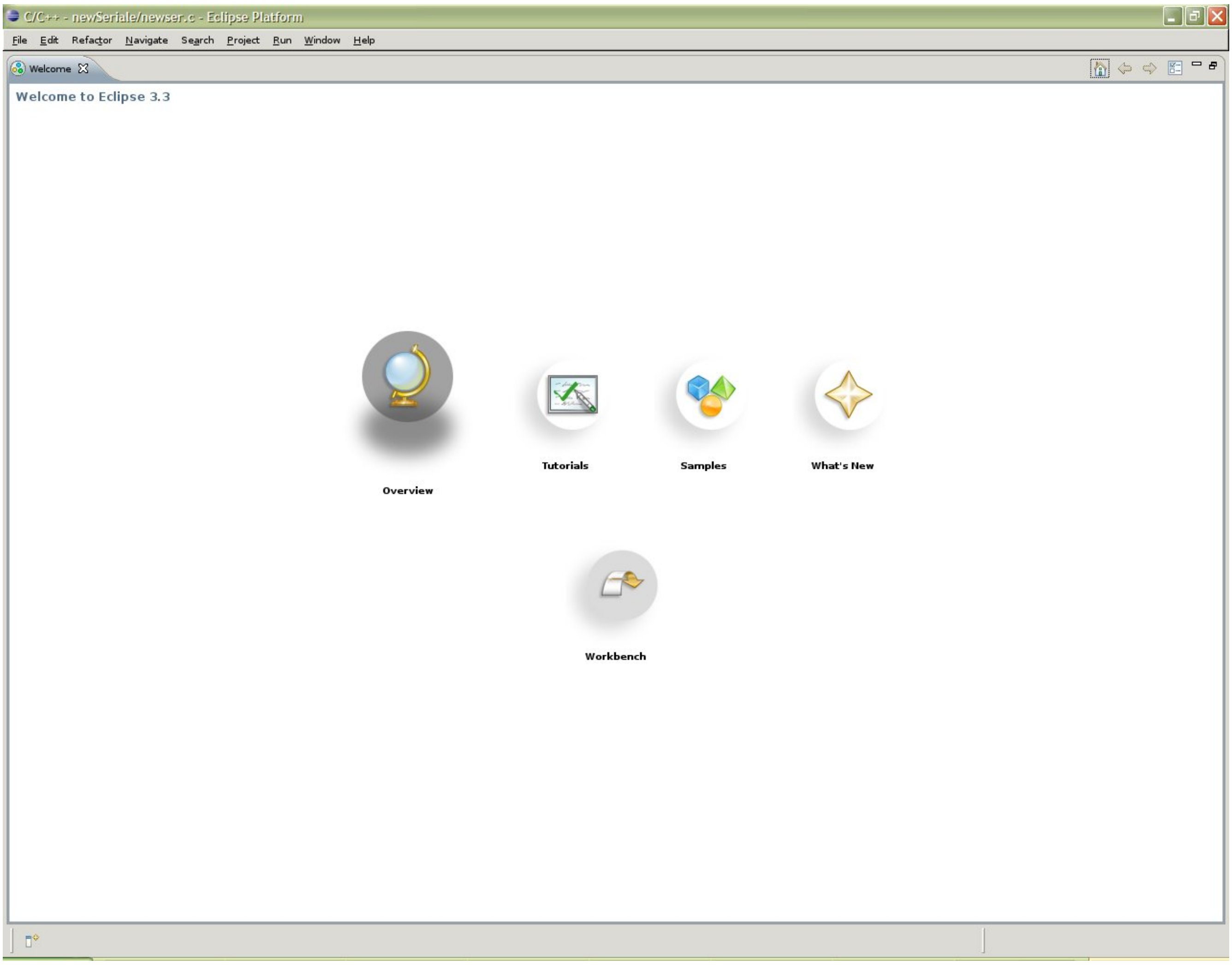
Select a workspace

Eclipse Platform stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /root/workspace

□ Use this as the default and do not ask again

- When the workspace has been created click on the **Workbench** icon.


C/C++ - newSeriale/newser.c - Eclipse Platform
File Edit Refactor Navigate Search Project Run Window Help
Welcome
Welcome to Eclipse 3.3
Overview
Tutorials
Samples
What's New
Workbench

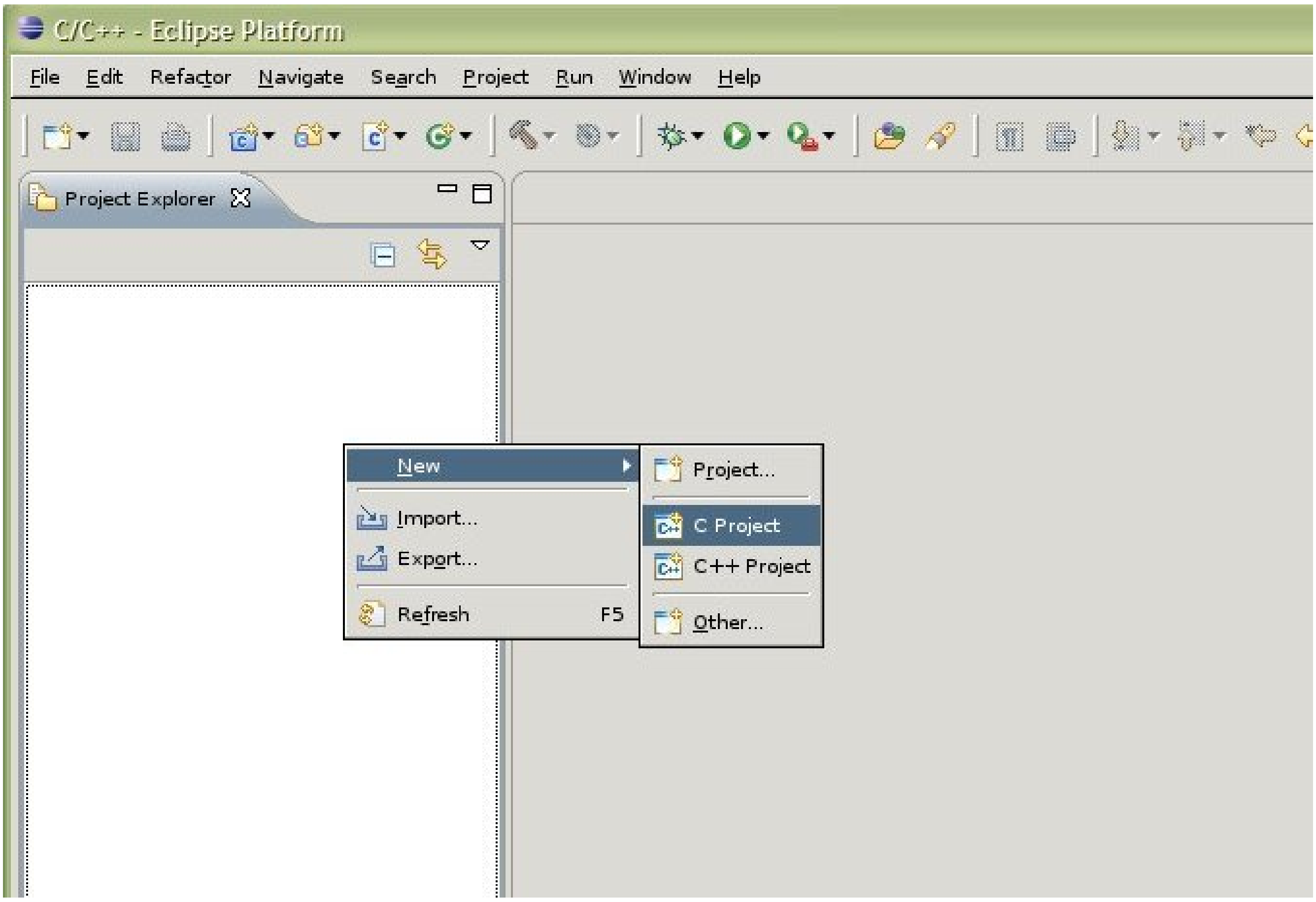
Now you are ready to develop your first application for the Pro^3.

6 Creating a project

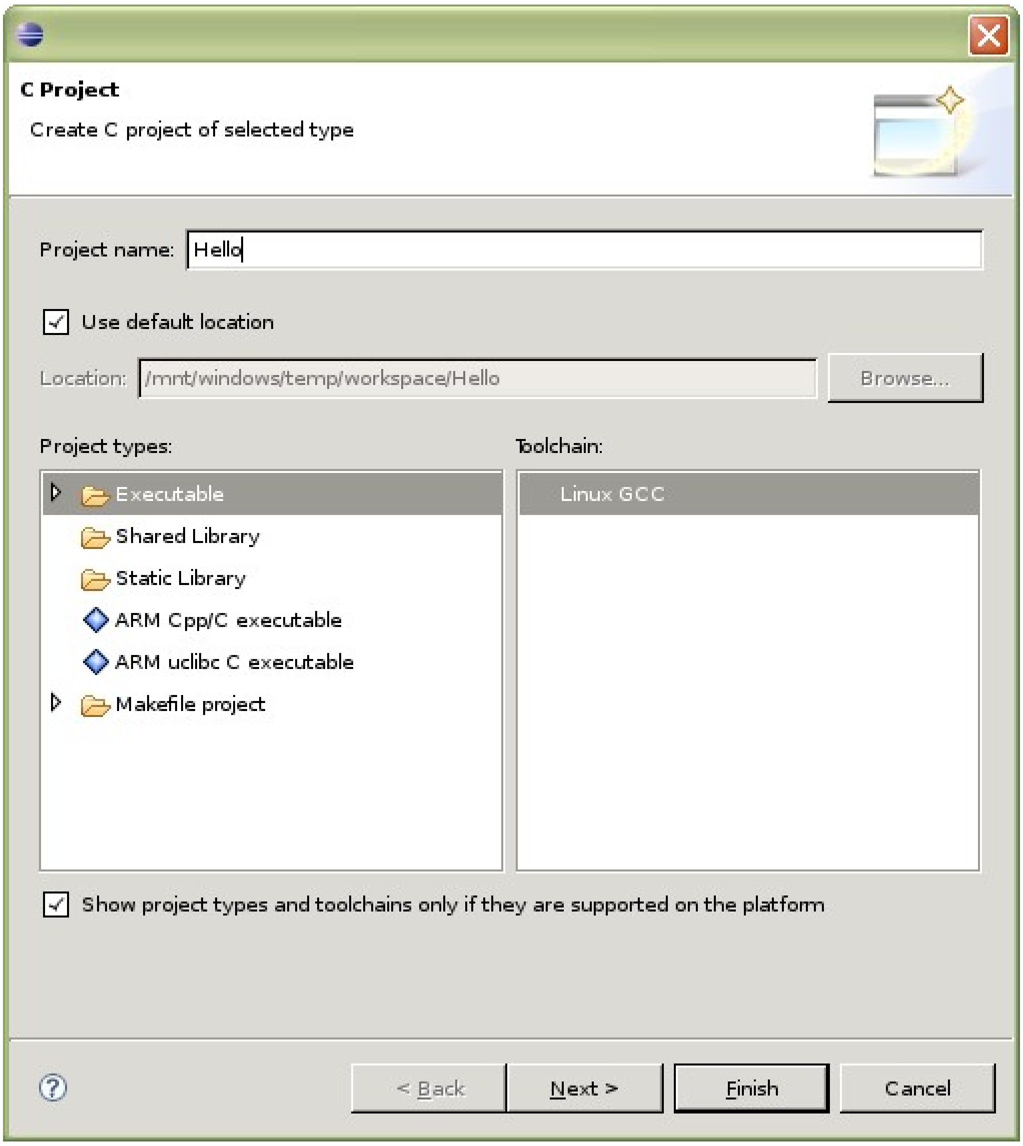
6.1 C or C++ project

To create a project, follow these steps:

- Open the IDE (Start menu → All Programs → Telit Development Platform → Telit Customized Eclipse).
- Open the menu **File** → **New** → **C Project**. As an alternative, you can right click on the Project Explorer frame and select the C Project like shown in the picture below:



Choose a name for the project and its location, and then select the **ARM uclibc C executable** for C projects or **Arm Cpp/C executable** for C++ or mixed C++/C projects. Click **Next**.



C Project

Create C project of selected type

Project name: Hello

☑ Use default location

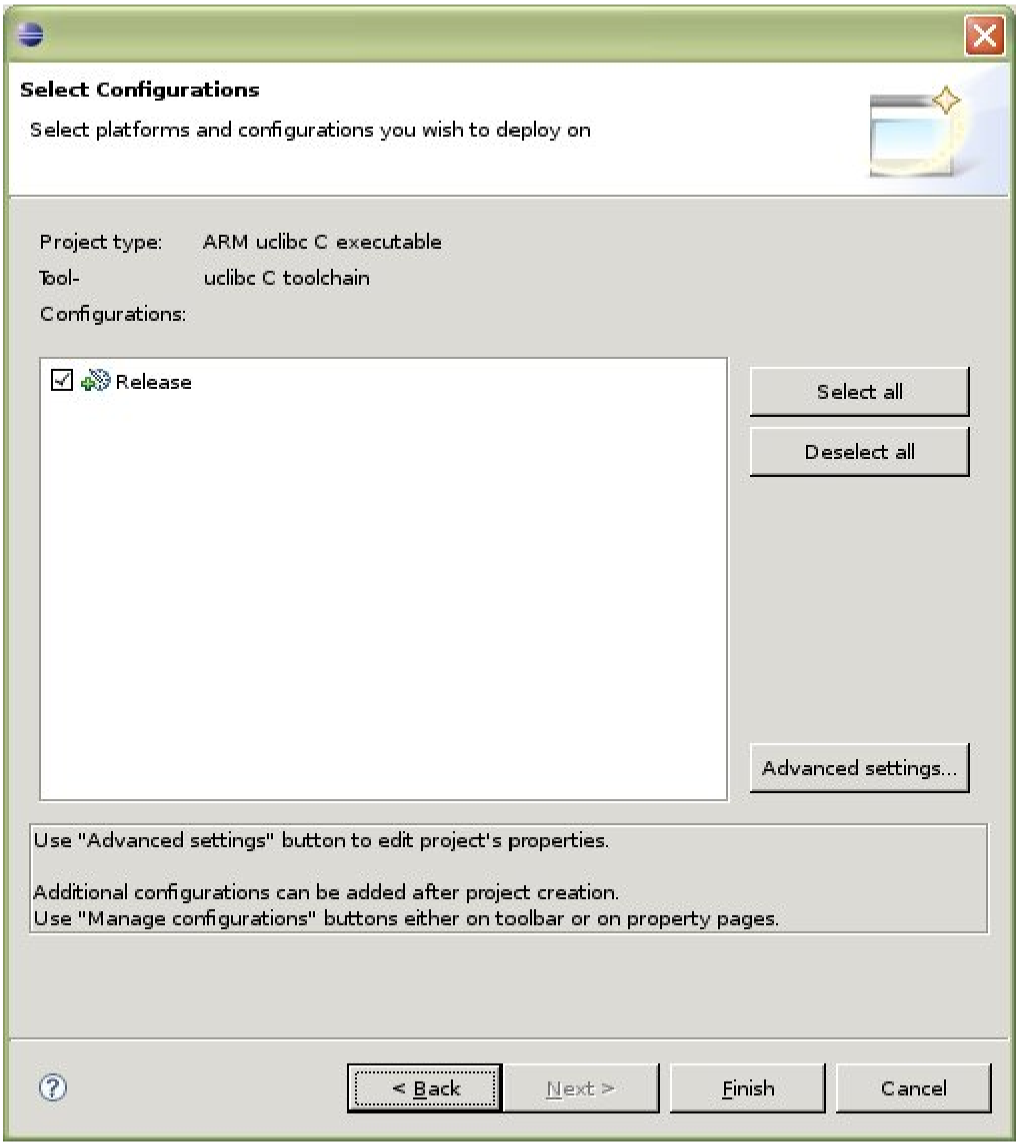
Location: /mnt/windows/temp/workspace/Hello

Project types:
- Executable
- Shared Library
- Static Library
- ARM Cpp/C executable
- ARM uclibc C executable
- Makefile project

Toolchain:
- Linux GCC

☑ Show project types and toolchains only if they are supported on the platform

Now you can choose the configuration for your project. Make sure that at least one configuration is selected. Click **Finish**.



Select Configurations

Select platforms and configurations you wish to deploy on

Project type: ARM uclibc C executable

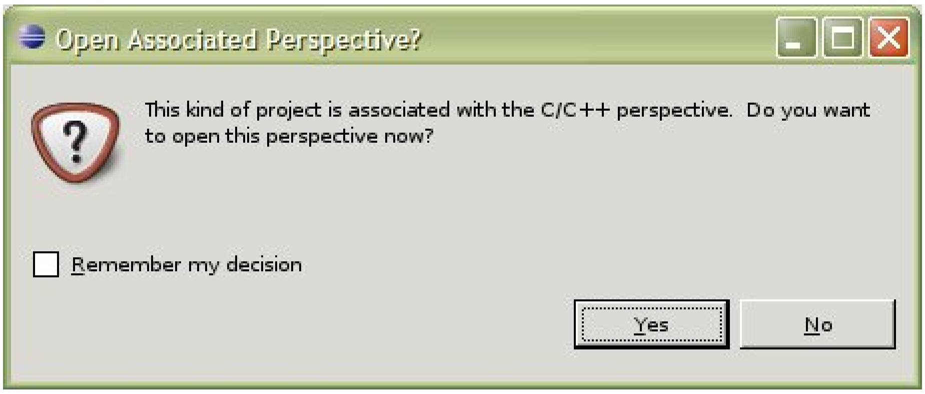
Tool-Configurations: uclibc C toolchain

☑ Release

Use "Advanced settings" button to edit project's properties.

Additional configurations can be added after project creation.
Use "Manage configurations" buttons either on toolbar or on property pages.

The following alert may appear:

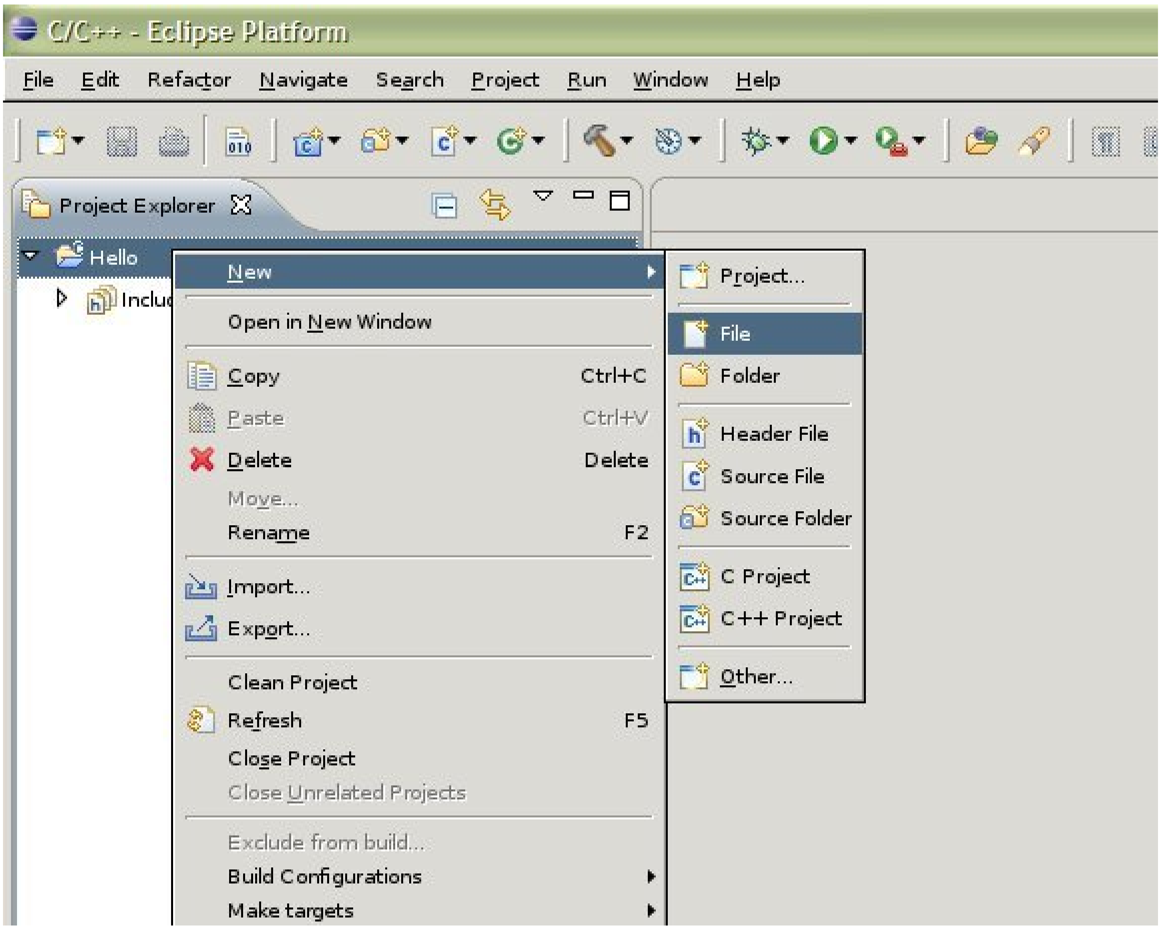


This kind of project is associated with the C/C++ perspective. Do you want to open this perspective now?

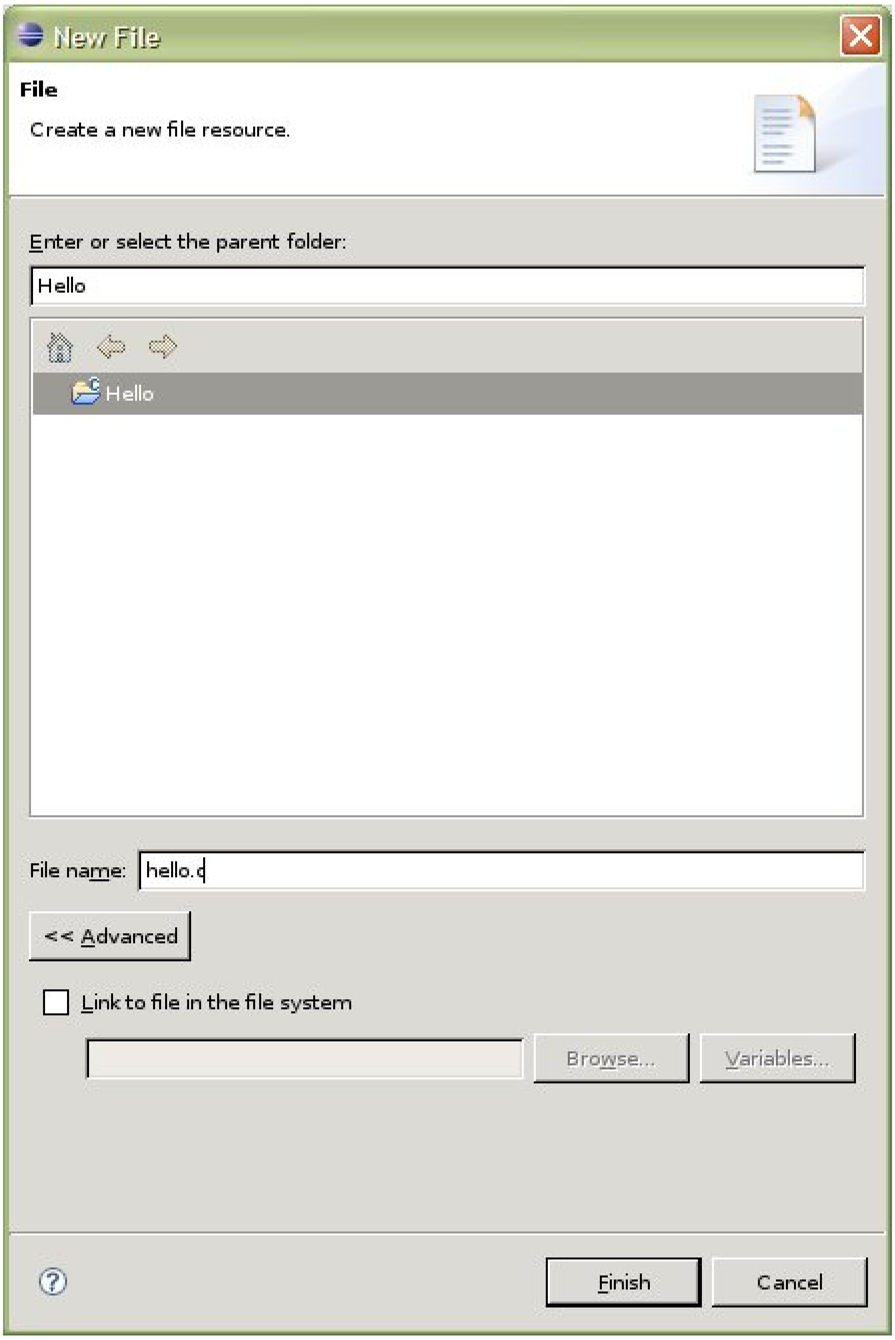
☐ Remember my decision

Select *“**Yes**”* to proceed.

A folder with the newly created project will now appear in the Project Explorer section; it is empty because currently there are no files. In order to add a new file, right click on the project name and select **New** → **File** in the context menu.



Now write the desired file name in the appropriate field (if necessary write also the subdirectory where the file should be located) and then click on the **Finish** button in order to end the file creation procedure.



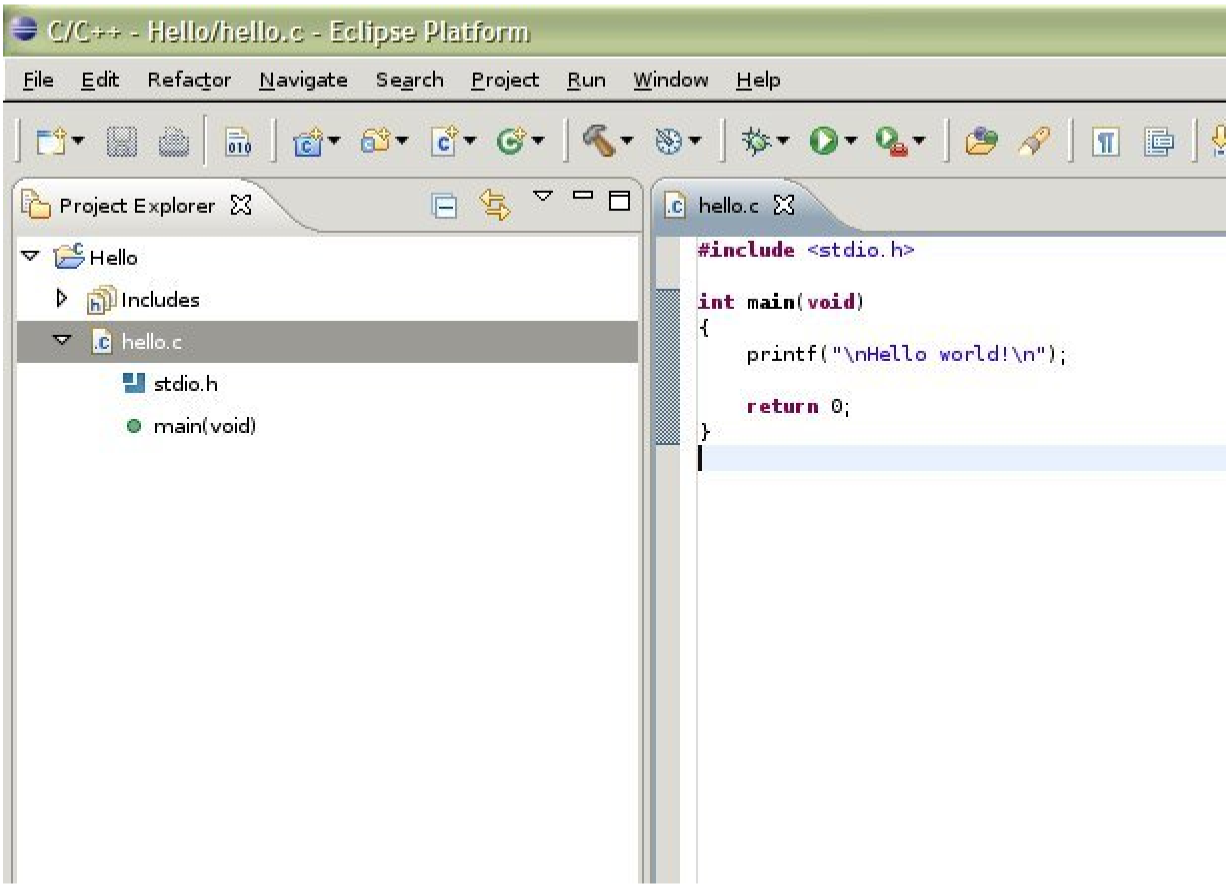
New File
File
Create a new file resource.
Enter or select the parent folder:

Hello

File name: hello.c

<< Advanced
□ Link to file in the file system
Browse...
Variables...
Finish
Cancel

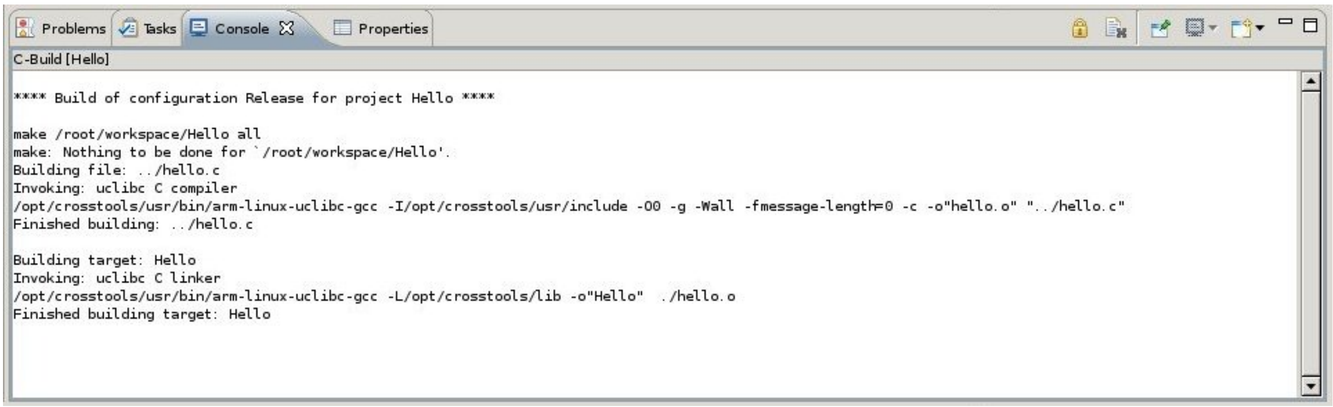
A blank text editor sheet will now appear in the main frame where you can write your application. In the following example, the classic “Hello world!” program is shown.



C/C++ - Hello/hello.c - Eclipse Platform
File Edit Refactor Navigate Search Project Run Window Help
Project Explorer
Hello
Includes
hello.c
stdio.h
main(void)
hello.c

```c
#include <stdio.h>

int main(void)
{
	printf("\nHello world!\n");

	return 0;
}
```

In order to build the application, select the menu option **Project** → **Build Project**. You should see in the console the messages shown below:



```
**** Build of configuration Release for project Hello ****

make /root/workspace/Hello all
make: Nothing to be done for `/root/workspace/Hello'.
Building file: ../hello.c
Invoking: uclibc C compiler
/opt/crosstools/usr/bin/arm-linux-uclibc-gcc -I/opt/crosstools/usr/include -O0 -g -Wall -fmessage-length=0 -c -o"hello.o" "../hello.c"
Finished building: ../hello.c

Building target: Hello
Invoking: uclibc C linker
/opt/crosstools/usr/bin/arm-linux-uclibc-gcc -L/opt/crosstools/lib -o"Hello"  ./hello.o
Finished building target: Hello
```

After a successful compilation, the binary file will be stored in the directory ***/<your workspace directory>/Project_Name/*****Release/**.

Please note that C++ programs need libraries which, by default, are not present in the target. Refer to Appendix D for further details.

6.2 Linking an application against a library

If your application makes use of an external library, the library itself and its header files must be into the Colinux Development Environment.
So start the Colinux console (**Windows Start Menu → All Programs → Telit Development Platform → Console**) and make sure that all the library components are correctly installed.
To find out the library type:

```
# ls /opt/crosstools/lib
```

or

```
# ls /opt/crosstools/telit/lib
```

then type:

```
# ls /opt/crosstools/usr/include
```

or

```
# ls /opt/crosstools/telit/include
```

to find out the library header files.

Whenever the library you want to use is not into the Colinux Development Environment, you can install it. Let's assume, for example, you want to install the libpthread.a library (which is actually already installed into the Colinux Development Environment).
Copy the library typing:

```
# cp /mnt/windows/<PATH>/libpthread.a /opt/crosstools/lib
```

Copy the header file typing:

```
# cp /mnt/windows/<PATH>/pthread.h /opt/crosstools/usr/include
```

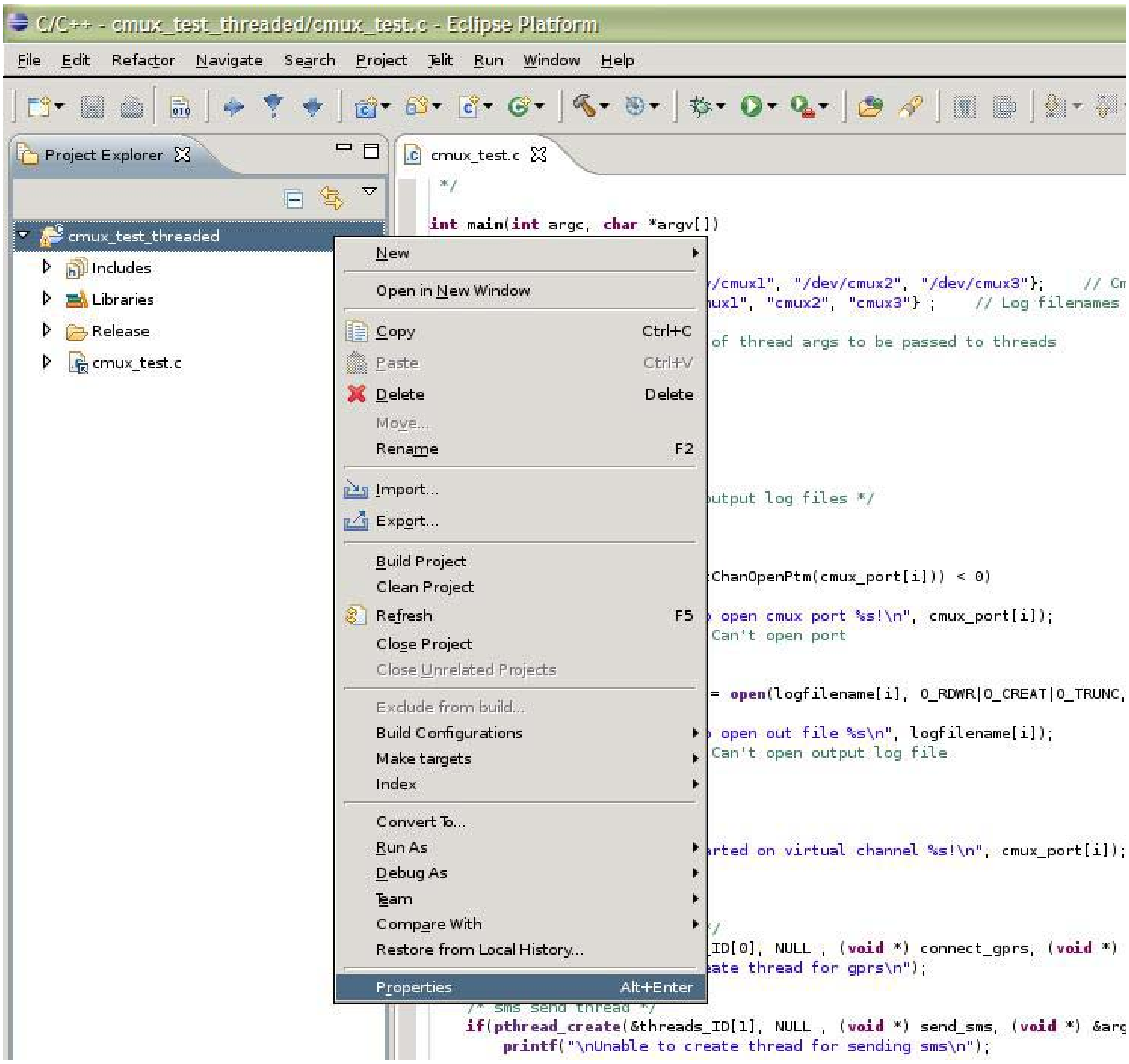
where **<PATH>** is the windows folder where you have stored the library files. For example, if you store them within C:\Temp you have to type:

```
# cp /mnt/windows/Temp/libpthread.a /opt/crosstools/lib
```

and

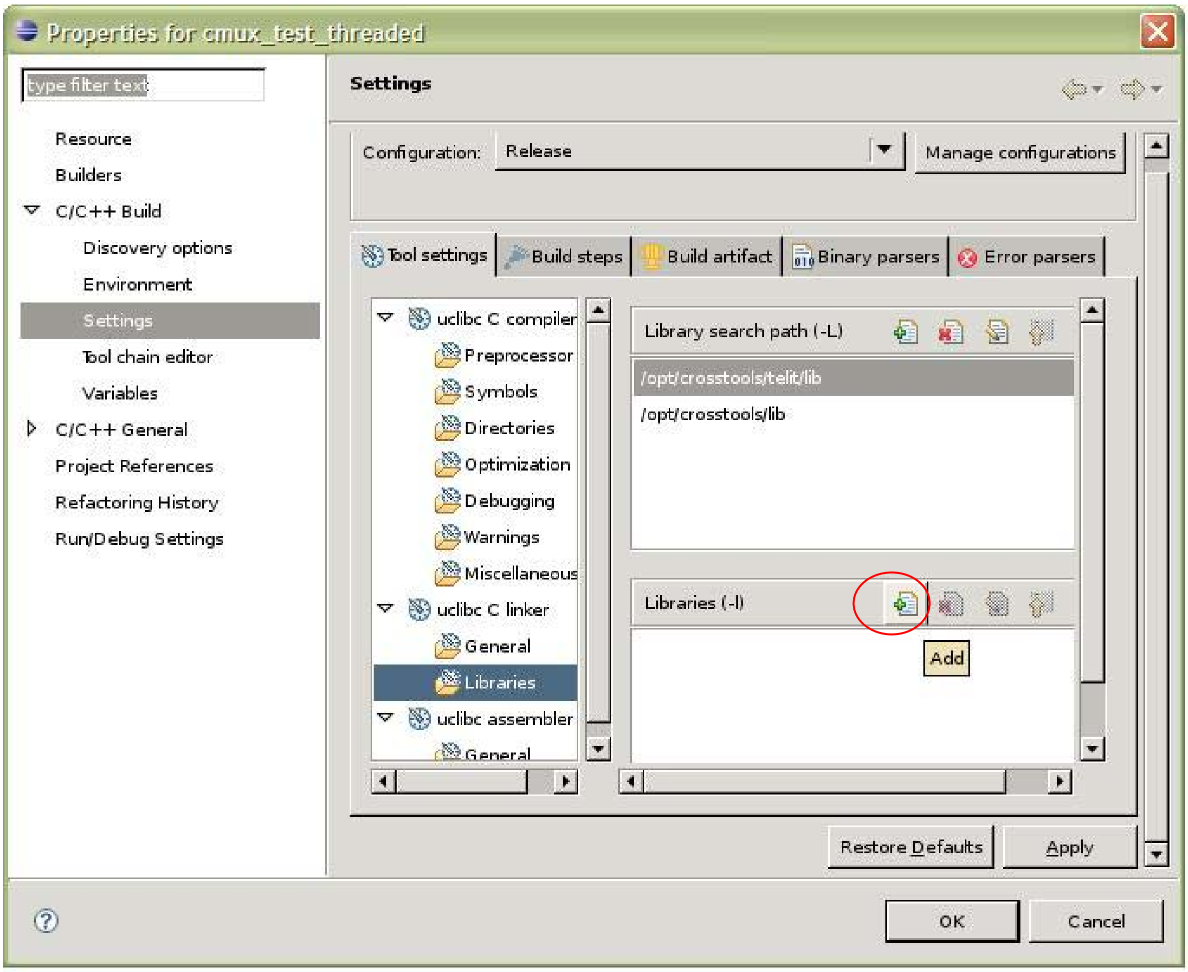
```
# cp /mnt/windows/Temp/pthread.h /opt/crosstools/usr/include
```

Once all the library components have been correctly set up, the application can be linked against the library itself. Under Eclipse right click on the project folder and select **Properties** as shown below:

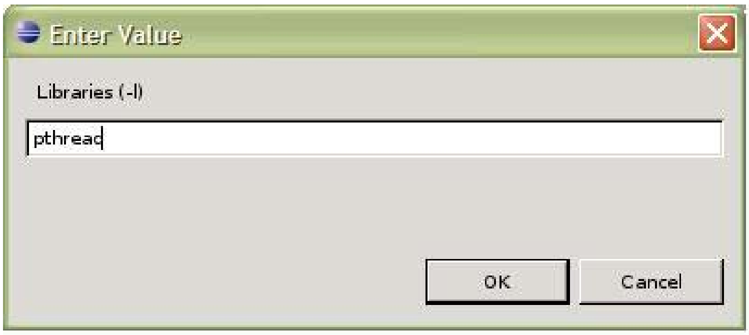


```
 */
int main(int argc, char *argv[])

/cmux1", "/dev/cmux2", "/dev/cmux3"};    // Cm
uxl", "cmux2", "cmux3"} ;    // Log filenames
of thread args to be passed to threads

utput log files */

ChanOpenPtm(cmux_port[i])) < 0)
 open cmux port %s!\n", cmux_port[i]);
Can't open port

= open(logfilename[i], O_RDWR|O_CREAT|O_TRUNC,
 open out file %s\n", logfilename[i]);
Can't open output log file

rted on virtual channel %s!\n", cmux_port[i]);

*/
ID[0], NULL , (void *) connect_gprs, (void *)
eate thread for gprs\n");
  /* sms send thread */
  if(pthread_create(&threads_ID[1], NULL , (void *) send_sms, (void *) &arg
      printf("\nUnable to create thread for sending sms\n");
```

The **Properties** tab for the application will appear. Now, from the **C/C++ Build** left menu, select **Settings** and then, from the **Tool settings** tab, select **Libraries** as shown below.

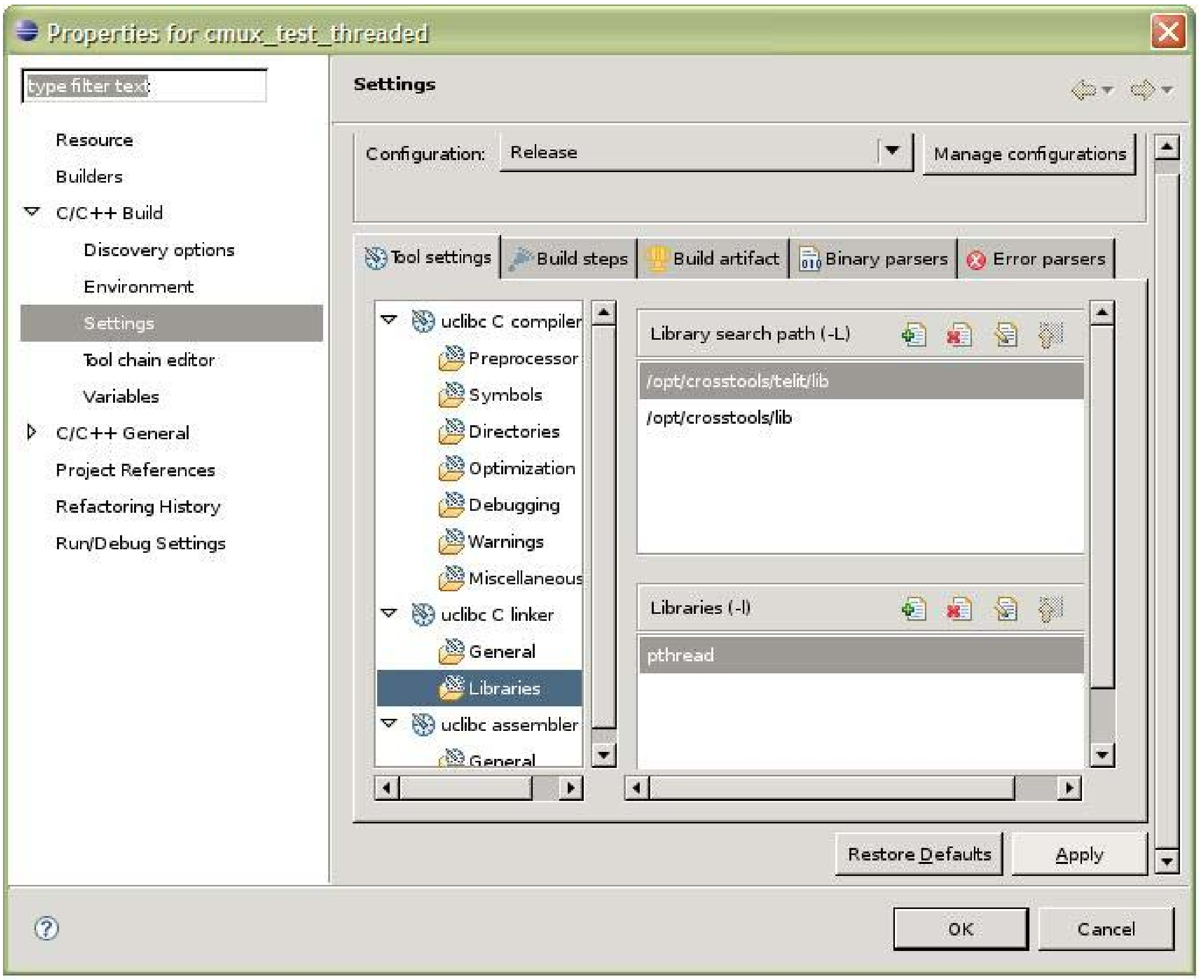


Now push the **Add** button (highlighted above) to link the library. Simply type the **pthread** value when the window below will appear and then push the **OK** button.



Please note that the ‘-l’ linker option automatically completes the library name with the ‘lib’ prefix: therefore there is no need to indicate the full library name (libpthread).

Once libpthread.a has been successfully added to the project, push the **Apply** and then the **OK** buttons of the main tab. Now the project using libpthread.a is ready to be compiled.



Configuration: Release

Library search path (-L)
/opt/crosstools/telit/lib
/opt/crosstools/lib

Libraries (-l)
pthread

6.3 Downloading a file to the target

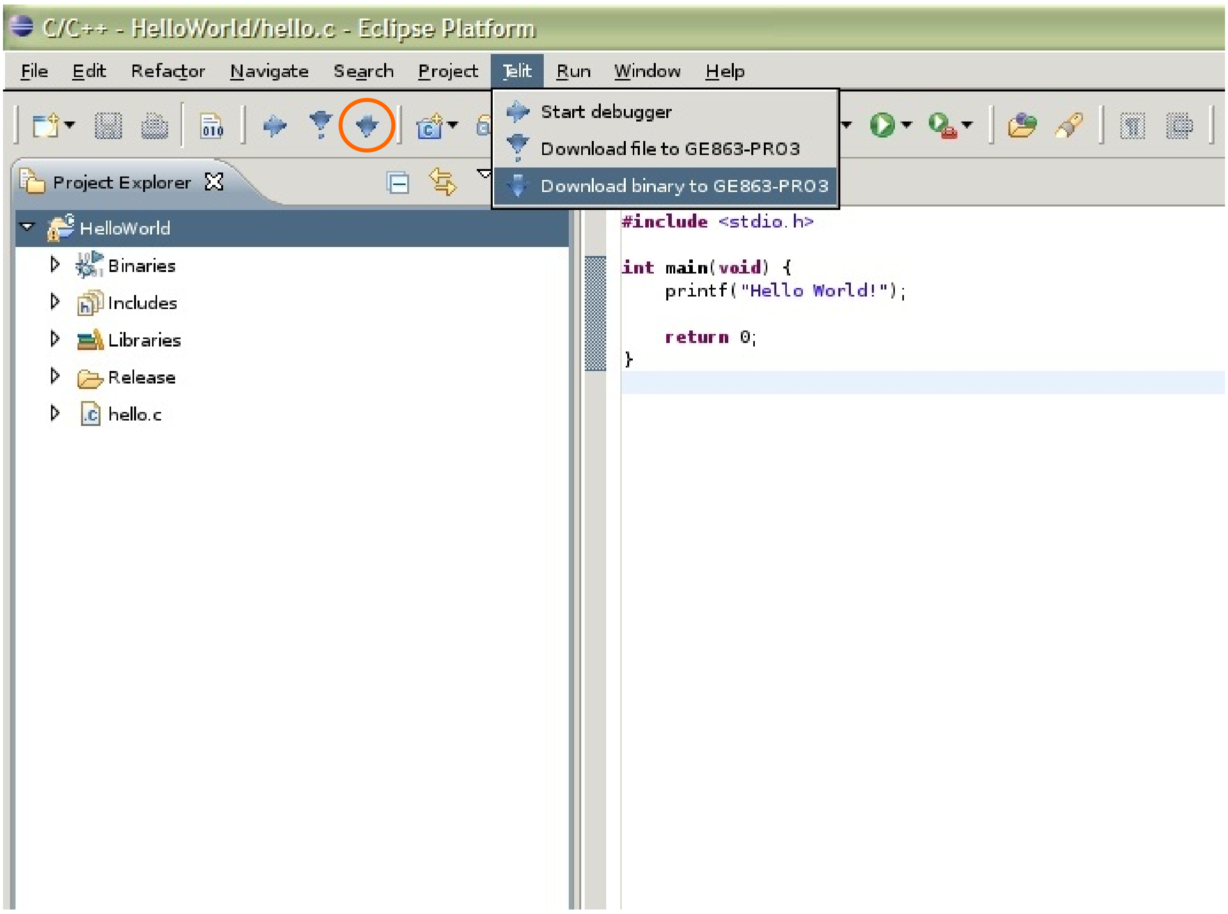
Once the project has been compiled, you can download it to the target. There are two possible options:

- Downloading the file using the Ethernet/USB port
- Downloading the file using the serial port.

6.3.1 Download through the Ethernet/USB port

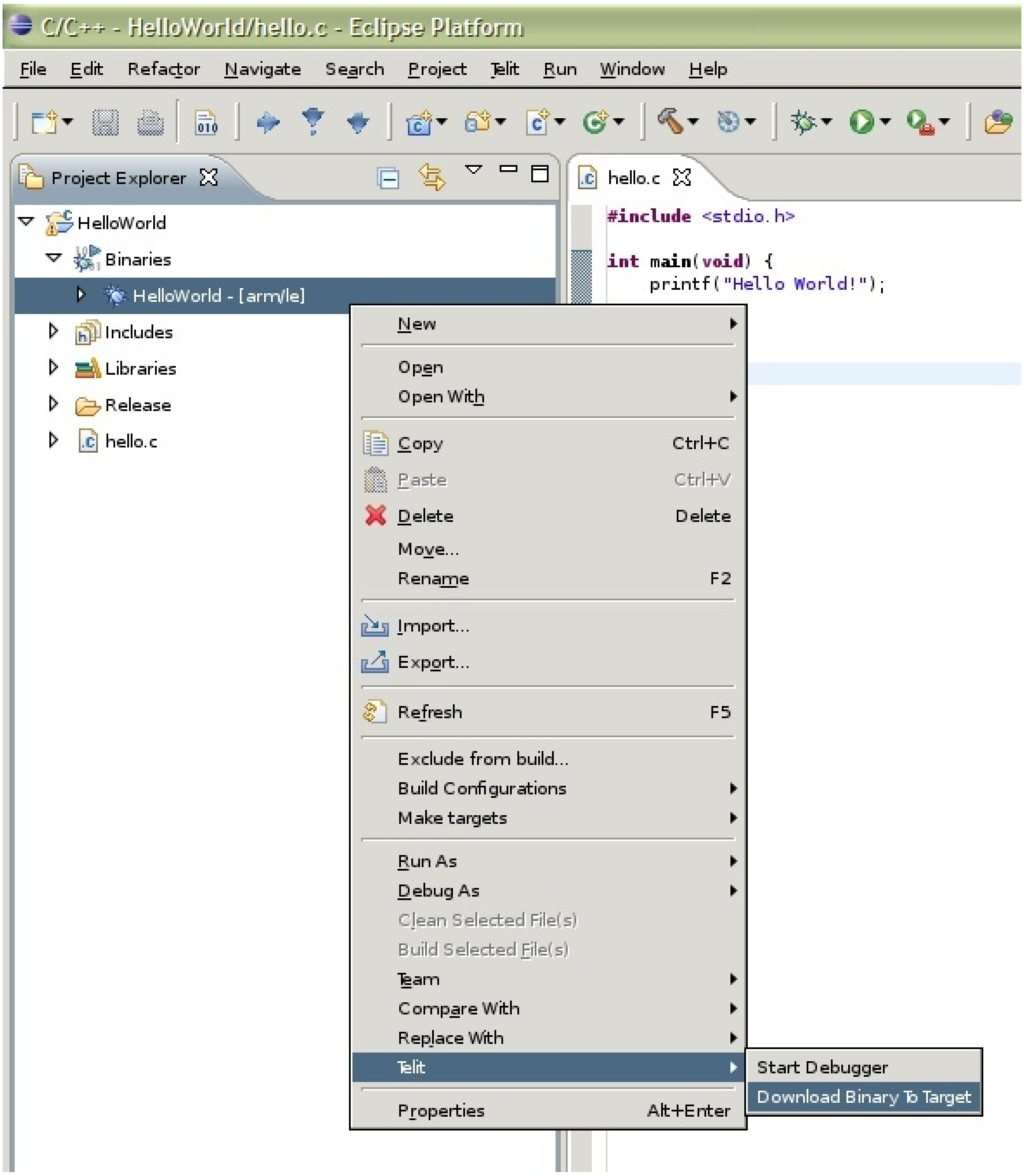
The simplest way to download the binary of a project is to use the IDE (note that the telnet daemon must be running in the target).

Highlight the project whose binary is going to be transferred and go to **Telit → Download binary to GE863-PRO3** or click on the button highlighted by the red circle in the picture.



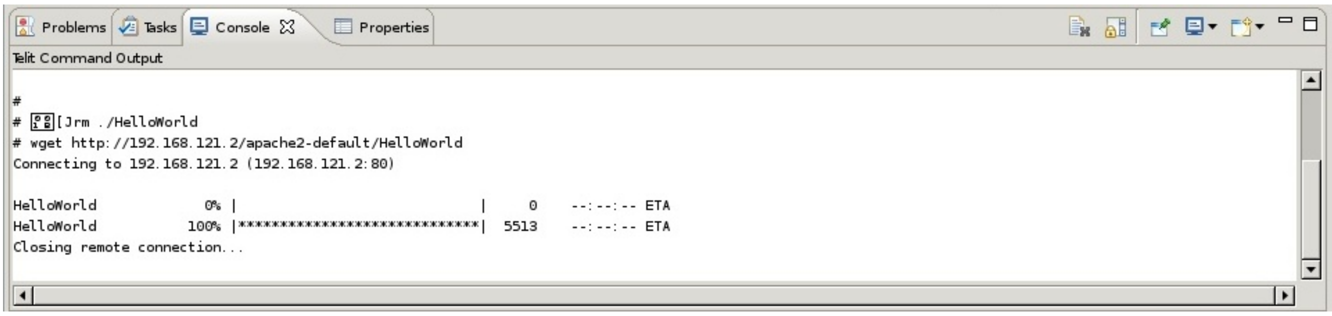
```
#include <stdio.h>

int main(void) {
    printf("Hello World!");

    return 0;
}
```

Alternatively, in the Project Explorer, you can right-click on the binary to be transferred, and choose the voice **Telit → Download Binary to Target** of the pop-up menu:



```
#include <stdio.h>

int main(void) {
    printf("Hello World!");
```

If the transfer succeeds you should see in the console a similar message:



Telit Command Output

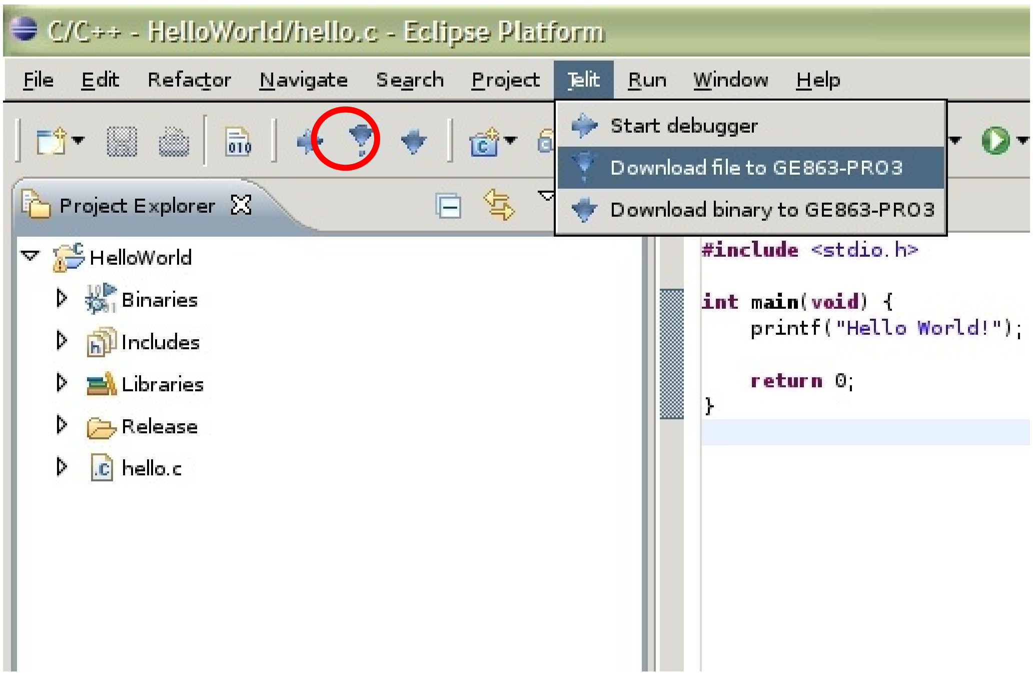
```
#
# □[Jrm ./HelloWorld
# wget http://192.168.121.2/apache2-default/HelloWorld
Connecting to 192.168.121.2 (192.168.121.2:80)

HelloWorld             0% |                               |     0    --:--:-- ETA
HelloWorld           100% |*******************************|  5513    --:--:-- ETA
Closing remote connection...
```

The binary can be found in the root directory of the target filesystem.

Pay attention that the first method will work only if the binary has the same name of the selected project and if the binary can be found inside the Release directory of the project.

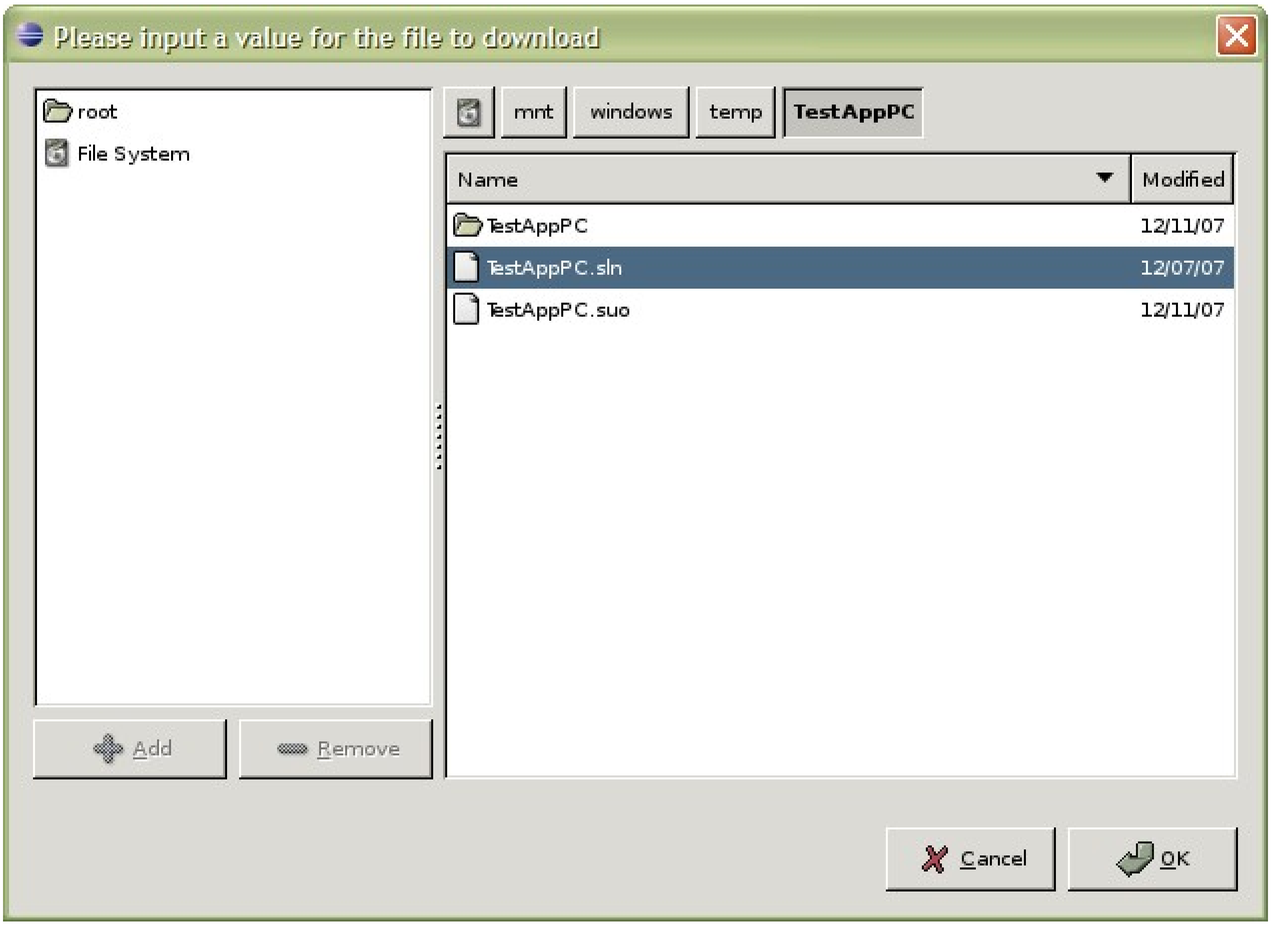
If you need to download to the target any other file than the compiled binary of a project follow these steps:

- Go to **Telit** → **Download file to GE863-PRO3** or click on the button highlighted by the red circle in the picture.



```
#include <stdio.h>

int main(void) {
    printf("Hello World!");

    return 0;
}
```

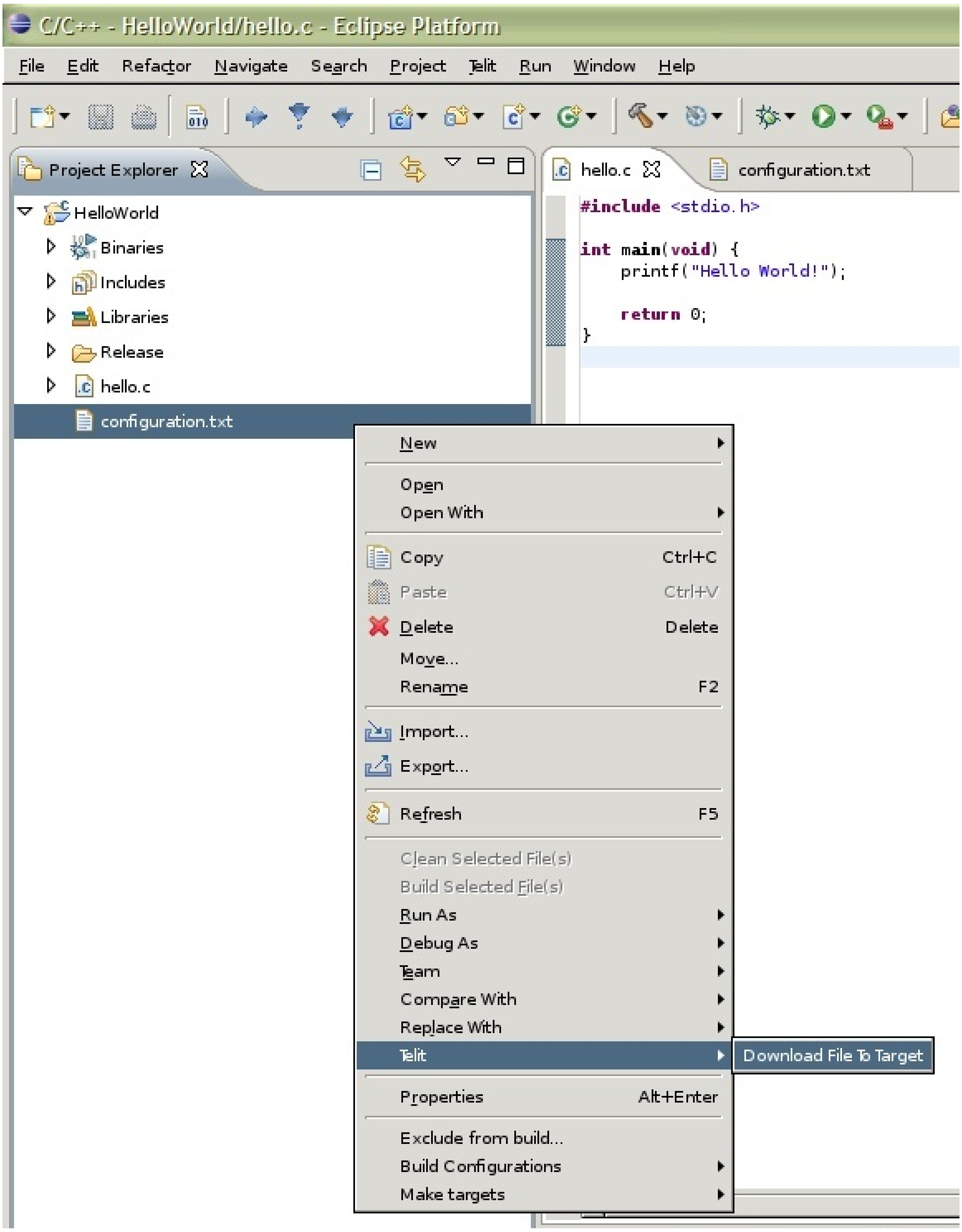
- Use the file selector to choose the file to be transferred and click **Ok**.



Please input a value for the file to download

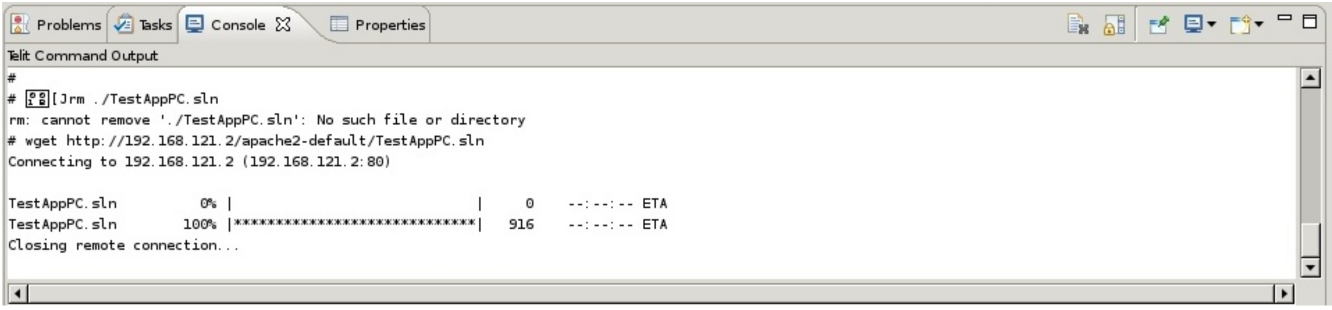
Name	Modified
TestAppPC	12/11/07
TestAppPC.sln	12/07/07
TestAppPC.suo	12/11/07

Alternatively, in the Project Explorer, if you have in a project the file to be transferred, you can right-click on it and choose the voice **Download File to Target** of the pop-up menu.



```
#include <stdio.h>

int main(void) {
    printf("Hello World!");

    return 0;
}
```

If the transfer succeeds you should see in the console named Telit Command Output a similar message:



Telit Command Output

```
#
# ␛[Jrm ./TestAppPC.sln
rm: cannot remove './TestAppPC.sln': No such file or directory
# wget http://192.168.121.2/apache2-default/TestAppPC.sln
Connecting to 192.168.121.2 (192.168.121.2:80)

TestAppPC.sln          0% |                             |     0    --:--:-- ETA
TestAppPC.sln        100% |*****************************|   916    --:--:-- ETA
Closing remote connection...
```

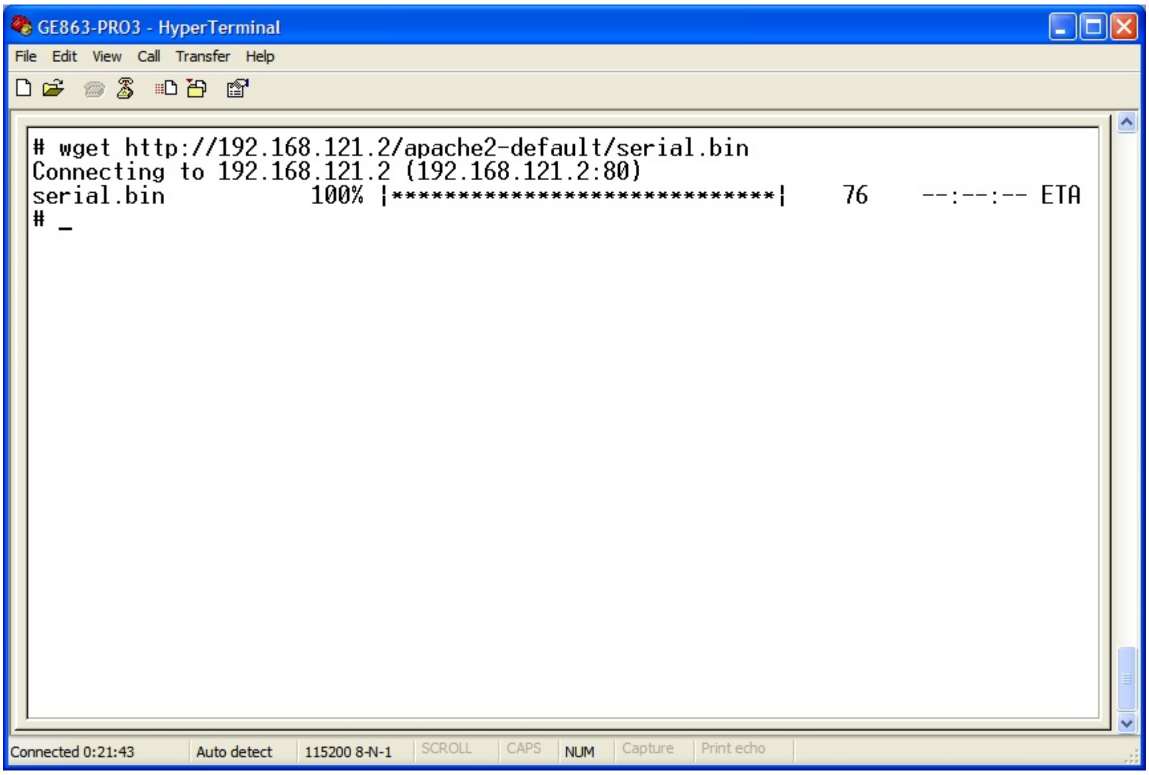
Alternatively you can choose not to use the IDE:

- Go to **Start → Telit Development Platform → Console**.
- Copy the files to be transferred in the directory `/var/www/apache2-default/`, for example opening a terminal and using the cp command:

  `cp <file_to_be_copied> /var/www/apache2-default/`

- From the target shell, within the directory where you want to place the downloaded file, type:

  ```
  wget http://192.168.121.2/apache2-default/file_to_be_copied
  ```



```
# wget http://192.168.121.2/apache2-default/serial.bin
Connecting to 192.168.121.2 (192.168.121.2:80)
serial.bin           100% |*****************************|    76    --:--:-- ETA
# _
```

If the file is an application you need to change its permission to execute it. Therefore type:

```
chmod 744 filename
```

6.3.2 Download through the serial port

Make sure your target “Upper=AP UART0” serial port is configured with the baudrate you wish to use for transferring the file. Open a console and type the following command to change the baudrate:

```
stty ispeed [ baudrate ] < /dev/ttyS1
```

Then type:

```
rz --zmodem </dev/ttyS1 >/dev/ttyS1
```

in order to let the target receive your binary file through the zmodem protocol using ttyS1.

Now use the “send file” option from your terminal (make sure that zmodem file transfer option has been set) to download your application to the target.

If the file is an application you need to change its permission to execute it. Therefore type:

```
chmod 744 filename
```

7 Debugging an application

7.1 Overview

Telit Development Environment allows remote debugging on the target through Ethernet/USB connection. Two applications are needed:

- gdbserver: server part of the debugger which runs in the target; it must be launched every time the debugging starts. The file can be found in coLinux filesystem directory `/opt/crosstools/arm-linux-uclibc/target_utils/gdbserver` and it needs to be copied in the target using one of the methods seen in chapter 6. Remember to set the executable flag for the file using the `chmod` command as explained in 6.3.2. You also need to copy in the target (inside the `/lib/` directory) *libthread_db.so.1* that can be found in coLinux filesystem directory `/opt/crosstools/lib.`
- arm-linux-gdb: client part of the debugger which runs in coLinux. The file can be found in coLinux filesystem directory `/opt/crosstools/usr/bin/arm-linux-gdb.` It commands the debugger behaviour and it is used through a graphical user interface in Telit Development Environment provided by Eclipse.

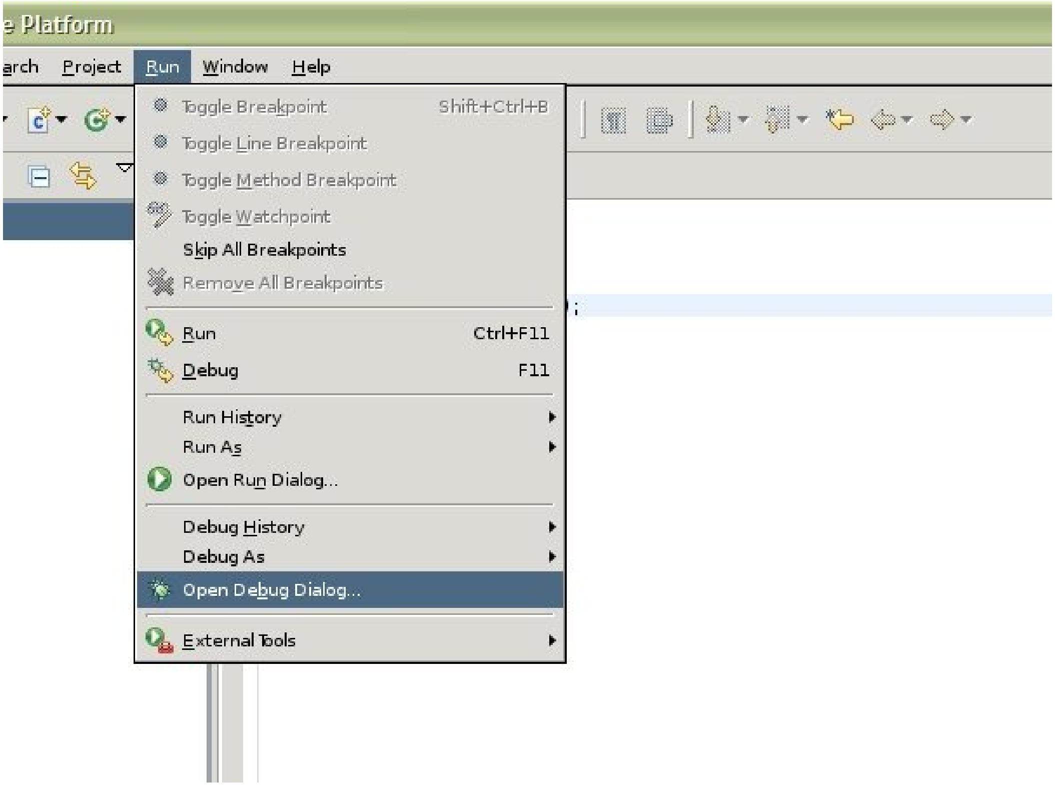
Both the server and the client need to be configured as explained in the following paragraphs.

7.2 Configuring the Debugger

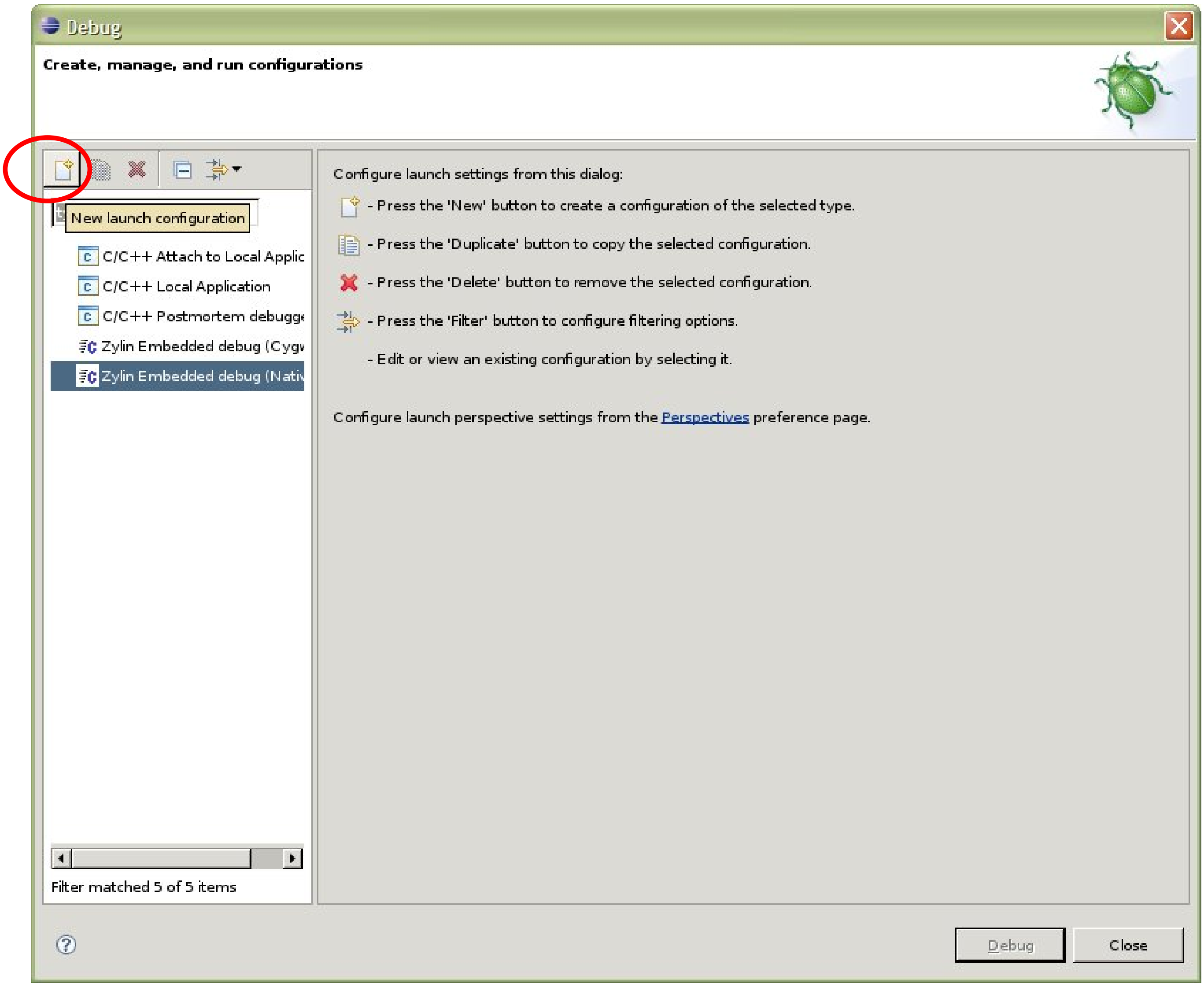
7.2.1 Creating a Debug Launch Configuration

Before start debugging your application, a **Debug Launch Configuration** for the project shall be created into Eclipse.
The Debug Launch Configuration locates the GDB debugger for Eclipse, locates the project's executable file (in this case it is used only to look up symbol information), and provides a startup script of GDB commands that are to be run as the debugger starts up.
A Debug Launch Configuration will be created for each project.

Select the menu option “**Run → Open Debug Dialog…**” to bring up the Debug Configuration Window.



In the “**Debug – Create, manage, and run configurations**” window below, click on “**Zylin Embedded debug (Native)**” followed by the “**New**” button.



Debug

Create, manage, and run configurations

Configure launch settings from this dialog:

- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

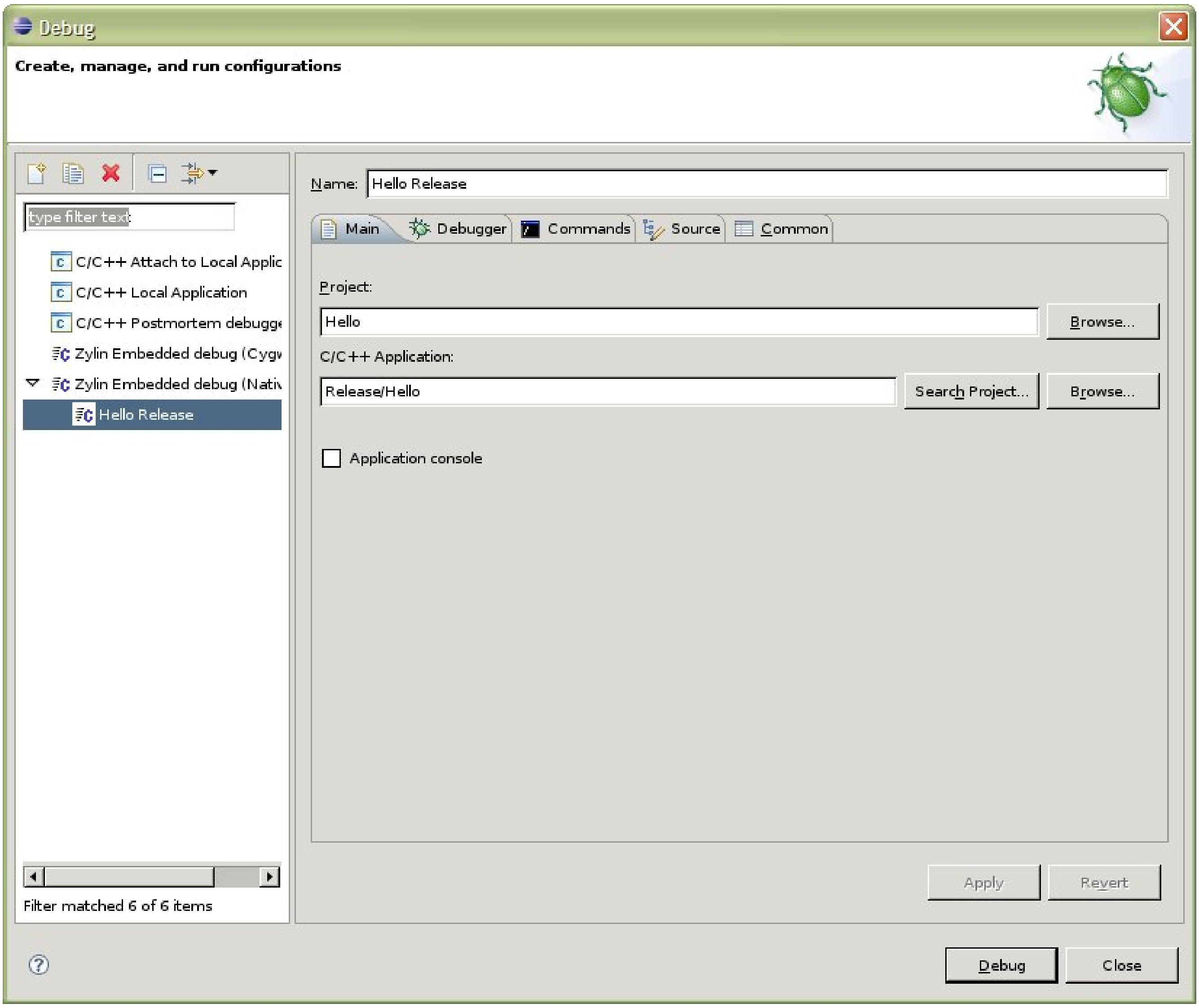
Configure launch perspective settings from the Perspectives preference page.

New launch configuration
C/C++ Attach to Local Applic
C/C++ Local Application
C/C++ Postmortem debugge
Zylin Embedded debug (Cygw
Zylin Embedded debug (Nativ
Filter matched 5 of 5 items
Debug
Close

The Debug “Create, manage and run configurations” window changes to the dialog shown below. Start by making sure that the “**Main**” tab is selected.

In the “Name:” text box, enter the name of this debug launch configuration (usually the project name). In this example use the “**Hello**”.
In the “Project” text box, use the “**Browse…**” button to find the project “**Hello**”.

In the “C/C++ Application” text box, use the “**Search Project…**” button to find the executable “**Hello”**.



Debug

Create, manage, and run configurations

type filter text

- C/C++ Attach to Local Applic
- C/C++ Local Application
- C/C++ Postmortem debugge
- Zylin Embedded debug (Cygw
- Zylin Embedded debug (Nativ
  - Hello Release

Filter matched 6 of 6 items

Name: Hello Release

Main | Debugger | Commands | Source | Common

Project:
Hello [Browse...]

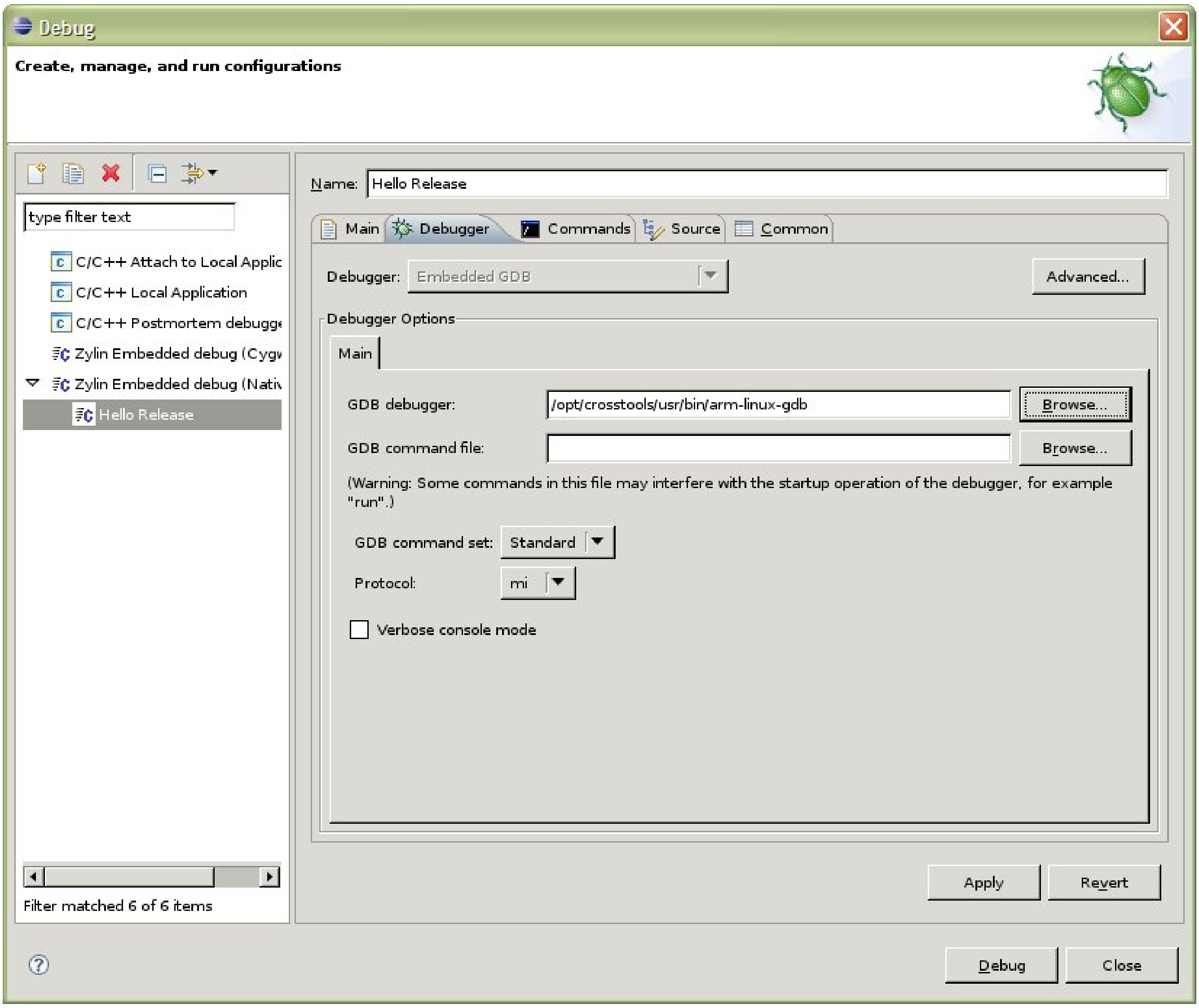
C/C++ Application:
Release/Hello [Search Project...] [Browse...]

☐ Application console

Now select the “**Debugger**” tab as shown below.

In the text box labeled “**GDB Debugger:**” use the “**Browse...**” button to locate the “**arm-linux-gdb**” file. It will be found in the “**/opt/crosstools/usr/bin**” folder.

Finally remove the “.gdbinit” GDB init script filename from the “**GDB command file:**” text box.



Name: Hello Release

Debugger: Embedded GDB

GDB debugger: /opt/crosstools/usr/bin/arm-linux-gdb

GDB command file:

(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

GDB command set: Standard

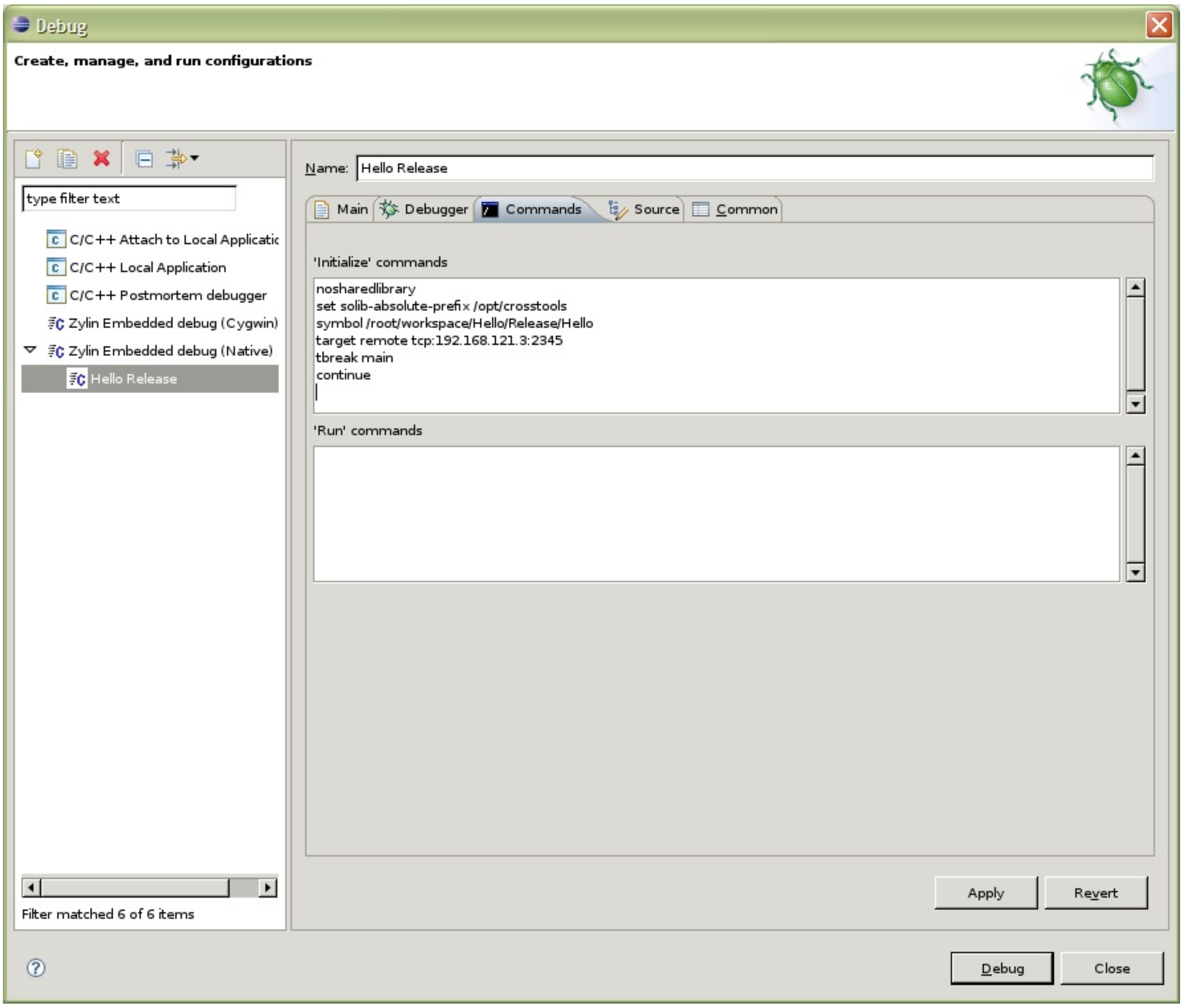
Protocol: mi

□ Verbose console mode

GDB can communicate with the target over an IP network/Ethernet-link-on-USB using TCP or UDP.GDB – IP network connection configuration (Ethernet or USB)

Select the “**Commands**” tab.
If your target is accessible through a network cable, in the “’Initialize’ commands” text window, enter the following GDB startup commands, as shown in the picture below:

- nosharedlibrary
- set solib-absolute-prefix /opt/crosstools
- symbol [filename]
- target remote tcp:192.168.121.3:2345
- tbreak main
- continue


Debug
Create, manage, and run configurations
type filter text
C/C++ Attach to Local Applicatic
C/C++ Local Application
C/C++ Postmortem debugger
Zylin Embedded debug (Cygwin)
Zylin Embedded debug (Native)
Hello Release
Filter matched 6 of 6 items
Main
Debugger
Commands
Source
Common
Apply
Revert
Debug
Close

Name: Hello Release

'Initialize' commands

```
nosharedlibrary
set solib-absolute-prefix /opt/crosstools
symbol /root/workspace/Hello/Release/Hello
target remote tcp:192.168.121.3:2345
tbreak main
continue
```

'Run' commands

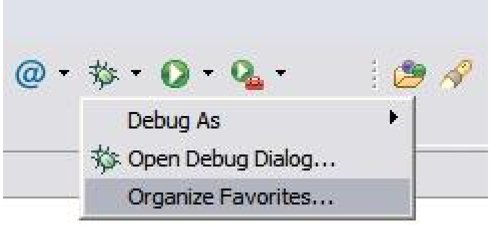
In the previous lines [filename] refers to the binary to be debugged, including the file path (for example the default workspace is `/root/workspace/`*`project_name`*`/Release/`*`file_name`*).
The “target remote tcp” command debugs using a TCP connection to port 2345 on 192.168.121.3 (the IP address you should use in the target if you have not changed default configuration).

Click on “**Apply**” followed by “**Close**” to finish specification of the debug launch configuration for **Hello** debugging. Eclipse may ask you if you want to save this configuration, answer “**Yes**”.

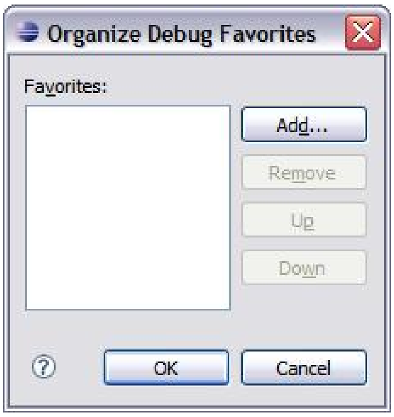
The “**Source**” and “**Common**” tabs can be left in their default state.

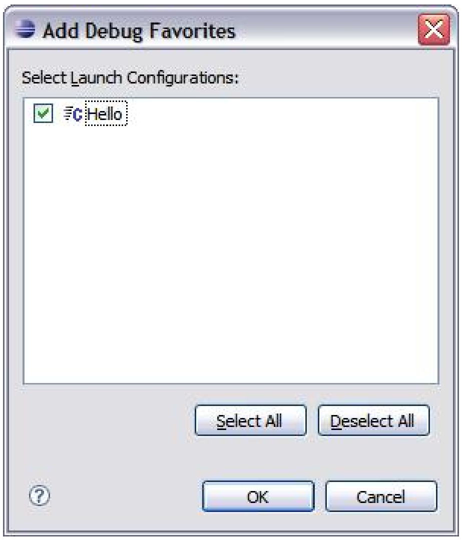
7.2.2 Organizing Favorites

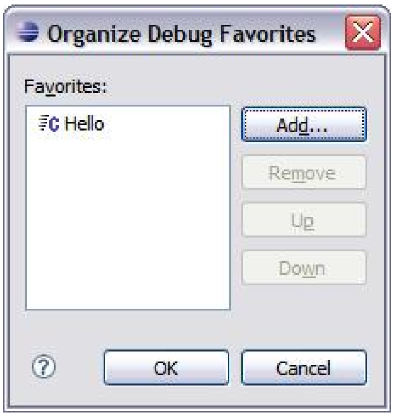
In order to simplify the debug process you can add the launch configuration that you wish to use in the **Favorites.** For example add the **Hello** embedded debug launch configuration into the Debug pull-down menu’s list of favorites.
In the toolbar, click on the **down arrowhead** next to the debug symbol and then click “**Organize Favorites…**”.



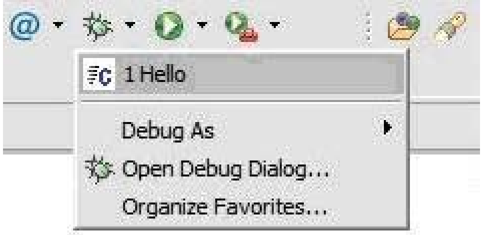
Click on “**Add…**” and either checkmark the “**Hello**” or click the “**Select All**” button.
Finally, click “**OK**” to enter this debug launch configuration into the debugger list of favorites, as shown below.





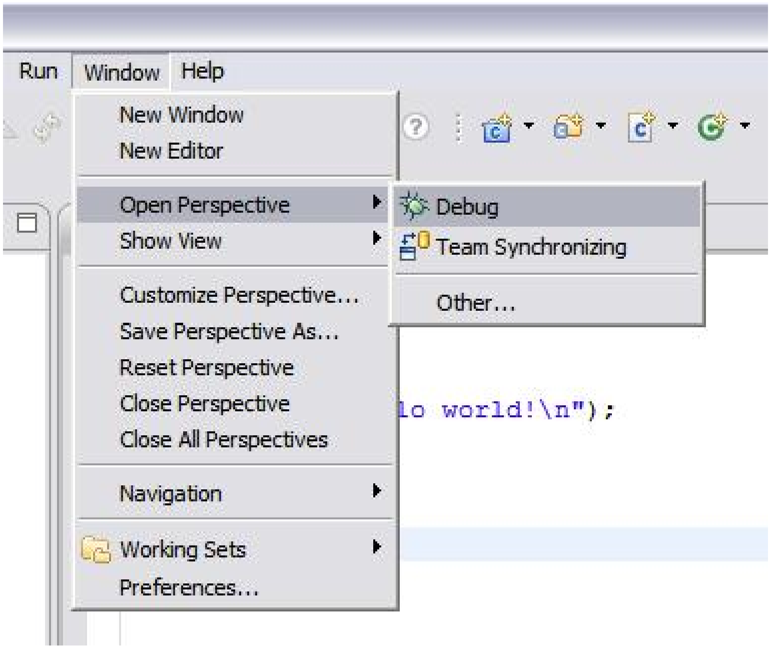


Now when you click on the Debug Toolbar button’s down arrowhead, you will see the “**Hello**” debug launch configuration installed as a favorite, as shown below.

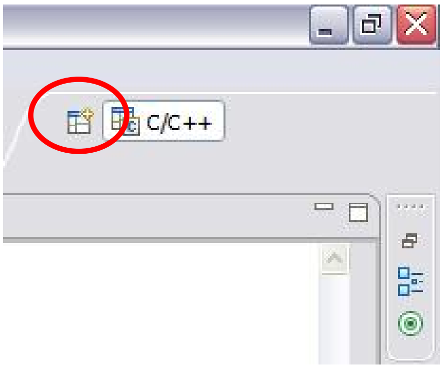


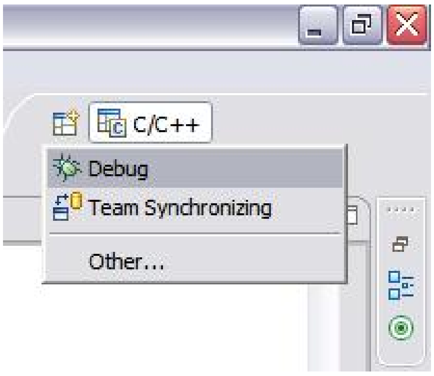
Now everything is in place to debug the project.

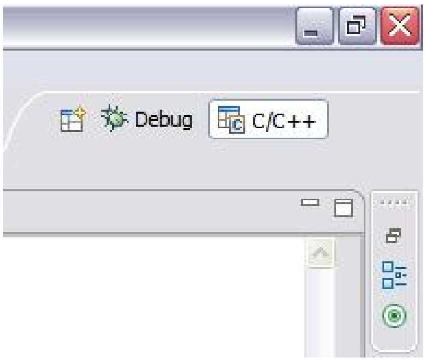
To debug, it is needed to switch from the C/C++ perspective to the Debug perspective. The standard way is to click on “**Window** → **Open Perspective** → **Debug**” as shown below.



A more convenient way to switch perspectives is to click on the “perspective” buttons at the Eclipse upper-right window location. Click on the “**OpenPerspective**” toolbar button below on the left and then choose “**Debug**” when the other perspectives are displayed.







Now a “Debug” button is shown below. You may have to drag on the edge to expose all the perspective buttons. You can also right-click on any of the buttons and “Close” them in order to reduce the display to only the perspectives you are interested in.

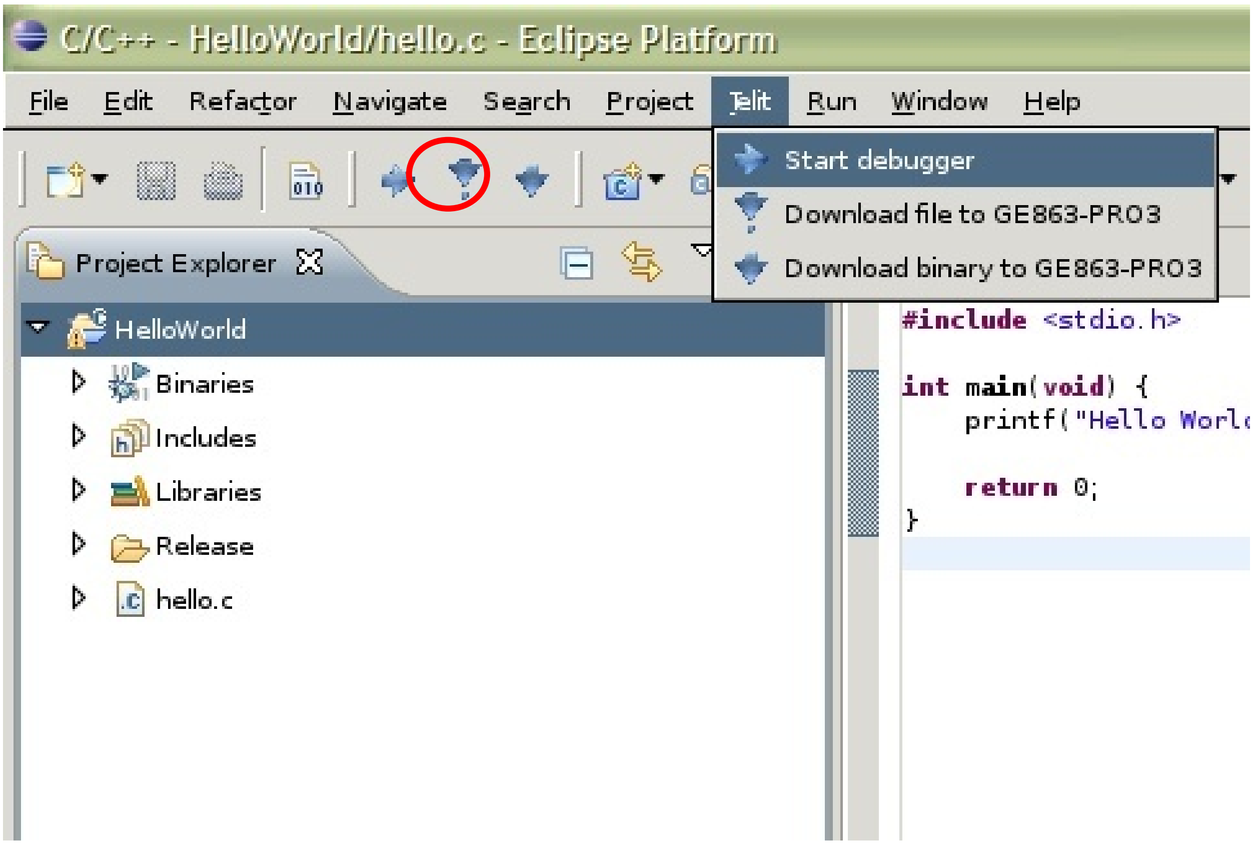
7.3 Running gdbserver

7.3.1 Remote IP network connection

As previously explained the **gdbserver** should be present in the target in the / directory and the **libthread_db.so.1** in the */lib/* directory.

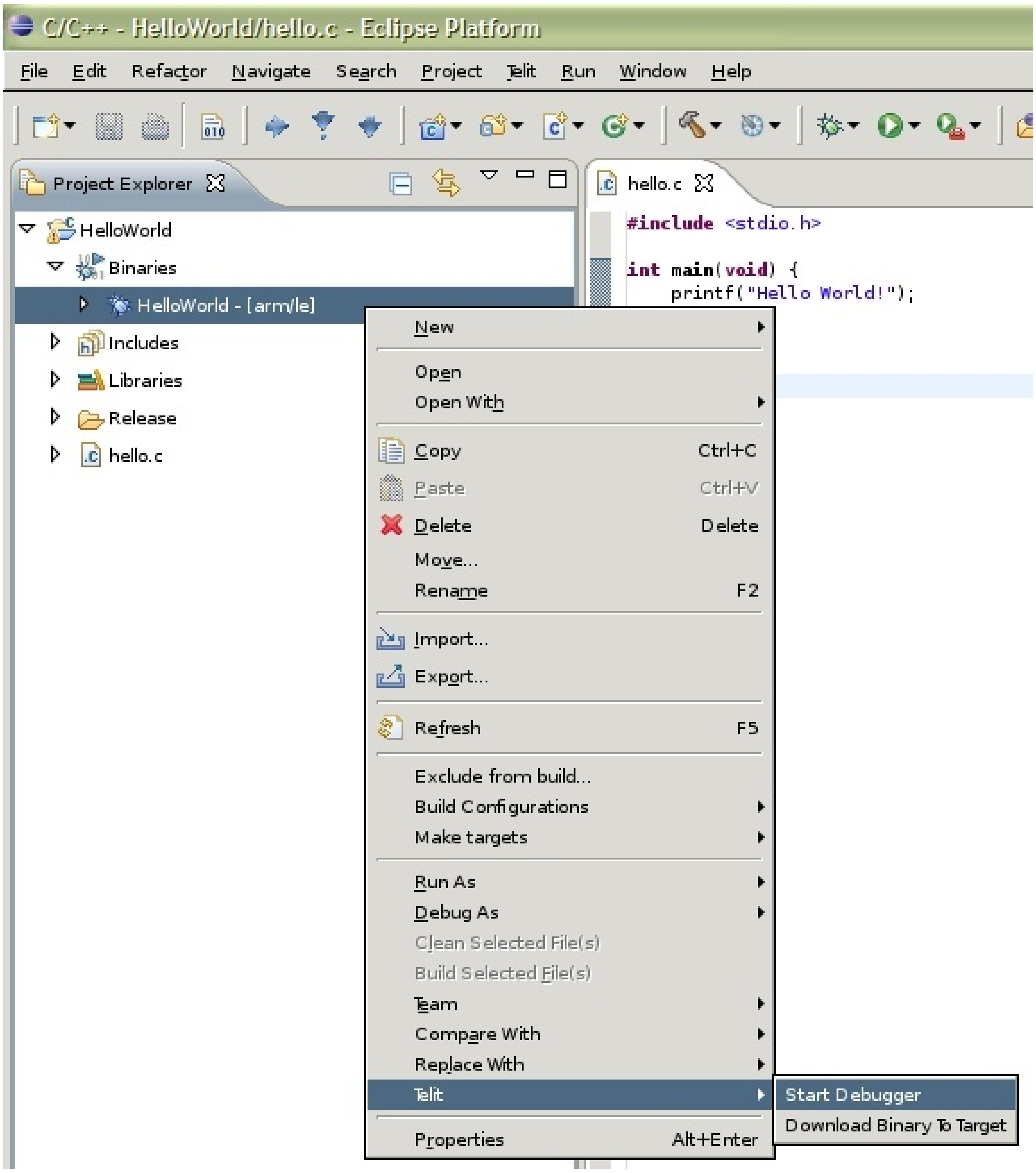
Be sure your application (also a stripped version of this latter can be used) has been already copied into the target (see paragraph 6.3.1).

If you are using the IDE, the simplest way to launch the debugger is highlighting the project to be debugged and going to **Telit → Start Debugger** or click on the button pointed out by the red circle in the following picture:


C/C++ - HelloWorld/hello.c - Eclipse Platform
File Edit Refactor Navigate Search Project Telit Run Window Help
Start debugger
Download file to GE863-PRO3
Download binary to GE863-PRO3
Project Explorer
HelloWorld
Binaries
Includes
Libraries
Release
hello.c

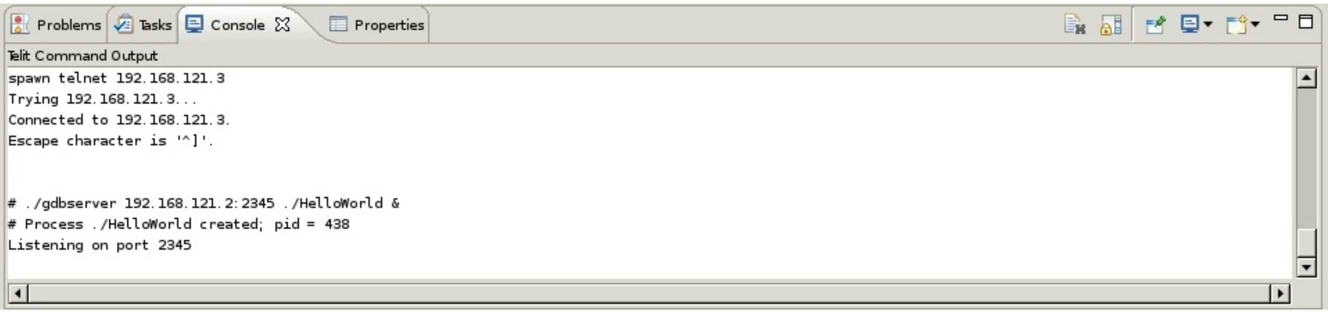
```
#include <stdio.h>

int main(void) {
    printf("Hello Worl

    return 0;
}
```

Alternatively, in the Project Explorer, you can right-click on the binary to be debugged, and choose the voice **Telit** → **Start Debugger** of the pop-up menu:



```
#include <stdio.h>

int main(void) {
    printf("Hello World!");
```

Note that the telnet daemon must be running in the target. If the operation succeeds you should see in the console named Telit Command Output a similar message:



```
spawn telnet 192.168.121.3
Trying 192.168.121.3...
Connected to 192.168.121.3.
Escape character is '^]'.


# ./gdbserver 192.168.121.2:2345 ./HelloWorld &
# Process ./HelloWorld created; pid = 438
Listening on port 2345
```

If you want to manually start the gdbserver type in a target shell:

```
    ./gdbserver 192.168.121.2:2345 [ filename ] &
```

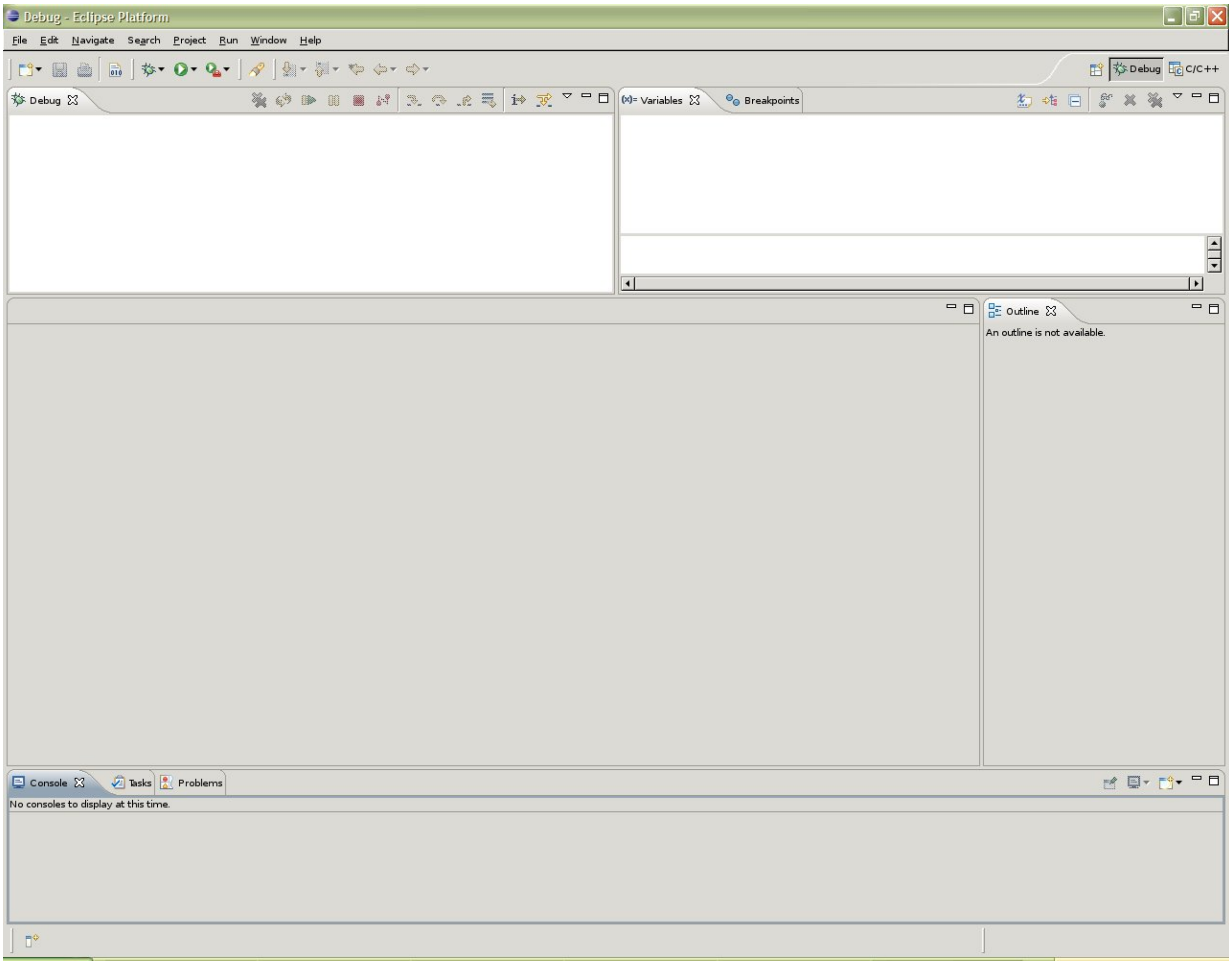
If The ‘host:2345’ argument means that gdbserver is to expect a TCP connection from machine ‘host’ (your workstation running Eclipse) to local TCP port 2345.
[filename] is your application’s filename with its own path.

It is recommended to let gdbserver run in background adding the “&” symbol to the command line as shown above.

Now gdbserver listens on port 2345 for a remote connection.

7.3.2 Running GDB

To start the Eclipse debugger, click on the “**Debug**” perspective button at the upper-right to open the Debug Perspective display, shown below.



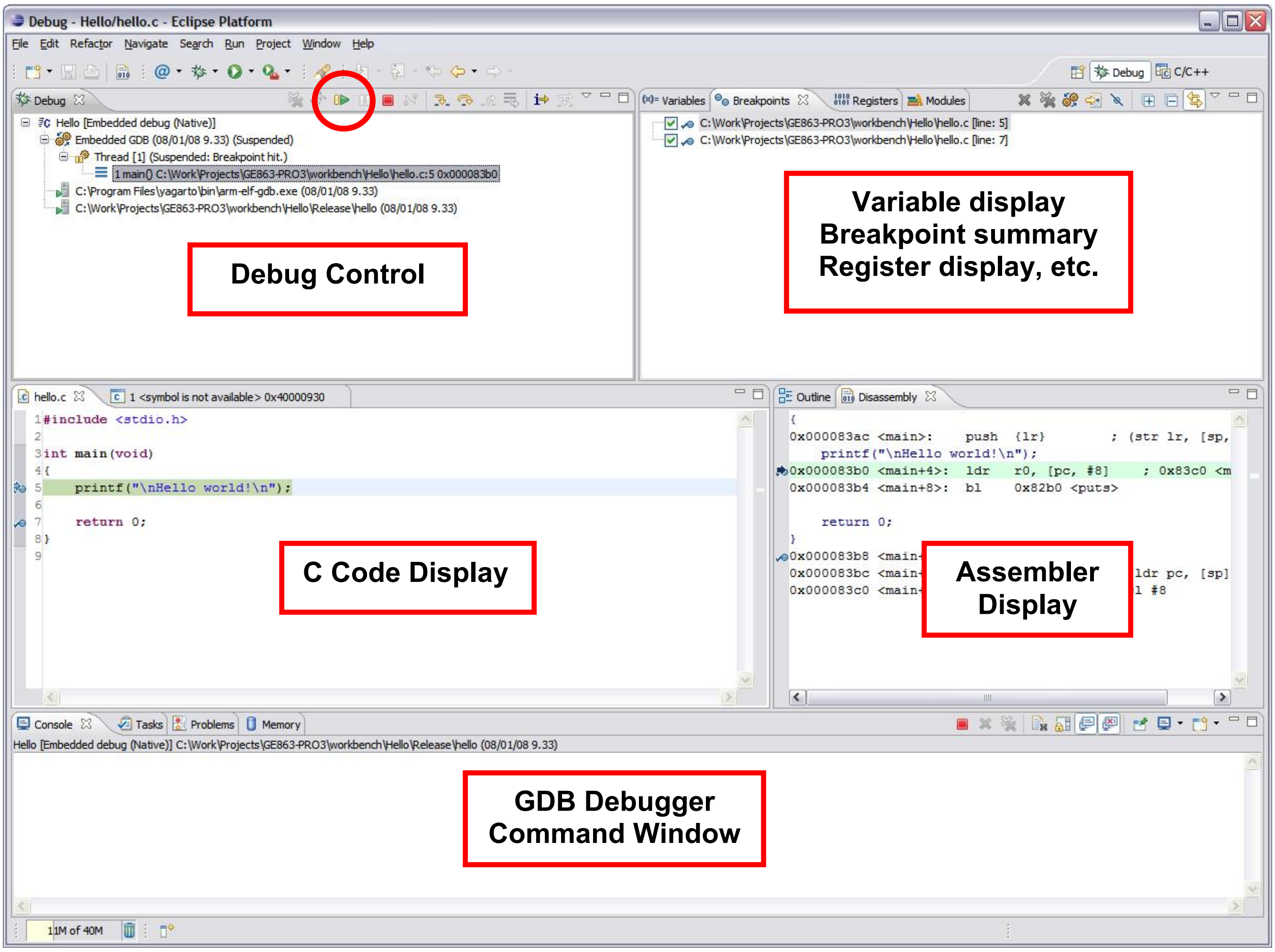
An outline is not available.

No consoles to display at this time.

If your display doesn’t look exactly like the debug display above, click on “**Window** → **Show View**” and select any of the missing elements.

Once the server in the target has been started go to **Run** → **Open Debug Dialog…,** choose the debug configuration for the project you wish to debug and click the **Debug** button.

In the screenshot below there are shown the components of the Debug perspective:



Debug - Hello/hello.c - Eclipse Platform

```
Hello [Embedded debug (Native)]
  Embedded GDB (08/01/08 9.33) (Suspended)
    Thread [1] (Suspended: Breakpoint hit.)
      1 main() C:\Work\Projects\GE863-PRO3\workbench\Hello\hello.c:5 0x000083b0
  C:\Program Files\yagarto\bin\arm-elf-gdb.exe (08/01/08 9.33)
  C:\Work\Projects\GE863-PRO3\workbench\Hello\Release\hello (08/01/08 9.33)
```

```
☑ C:\Work\Projects\GE863-PRO3\workbench\Hello\hello.c [line: 5]
☑ C:\Work\Projects\GE863-PRO3\workbench\Hello\hello.c [line: 7]
```

hello.c

1 <symbol is not available> 0x40000930

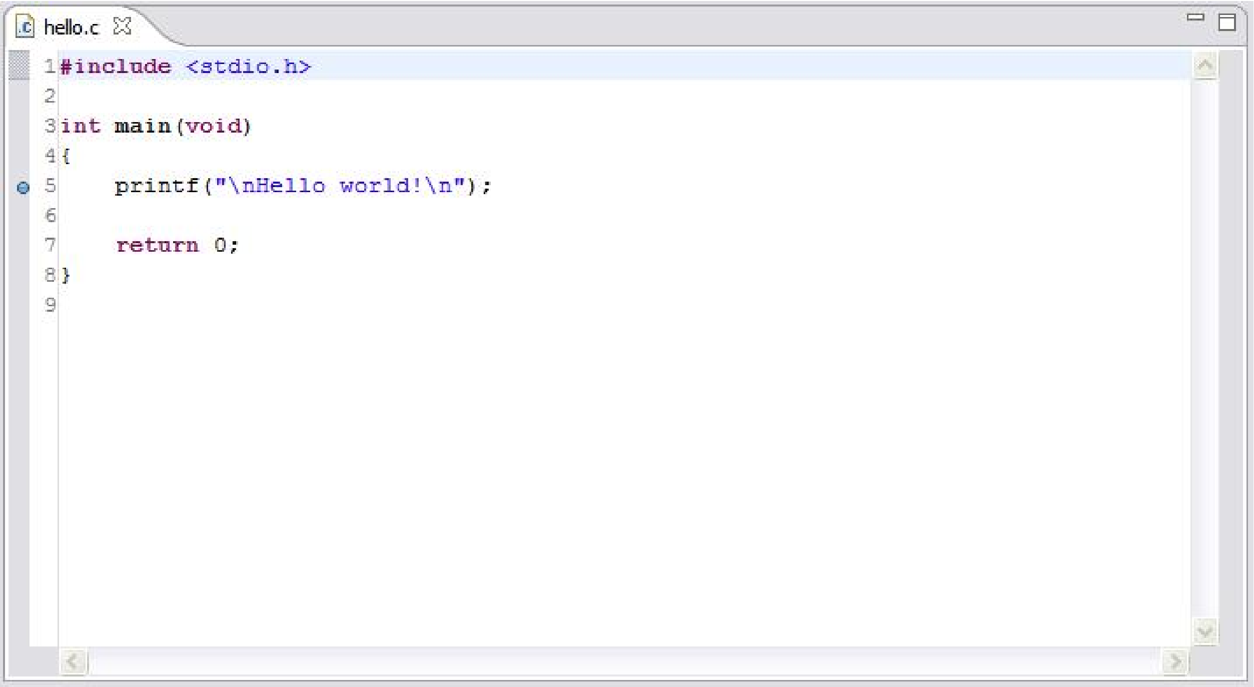
```
#include <stdio.h>

int main(void)
{
    printf("\nHello world!\n");

    return 0;
}

```

```
{
0x000083ac <main>:    push  {lr}        ; (str lr, [sp,
    printf("\nHello world!\n");
0x000083b0 <main+4>:  ldr   r0, [pc, #8]     ; 0x83c0 <m
0x000083b4 <main+8>:  bl    0x82b0 <puts>

    return 0;
}
0x000083b8 <main+
0x000083bc <main+                          ldr pc, [sp]
0x000083c0 <main+                          l #8
```

Hello [Embedded debug (Native)] C:\Work\Projects\GE863-PRO3\workbench\Hello\Release\hello (08/01/08 9.33)

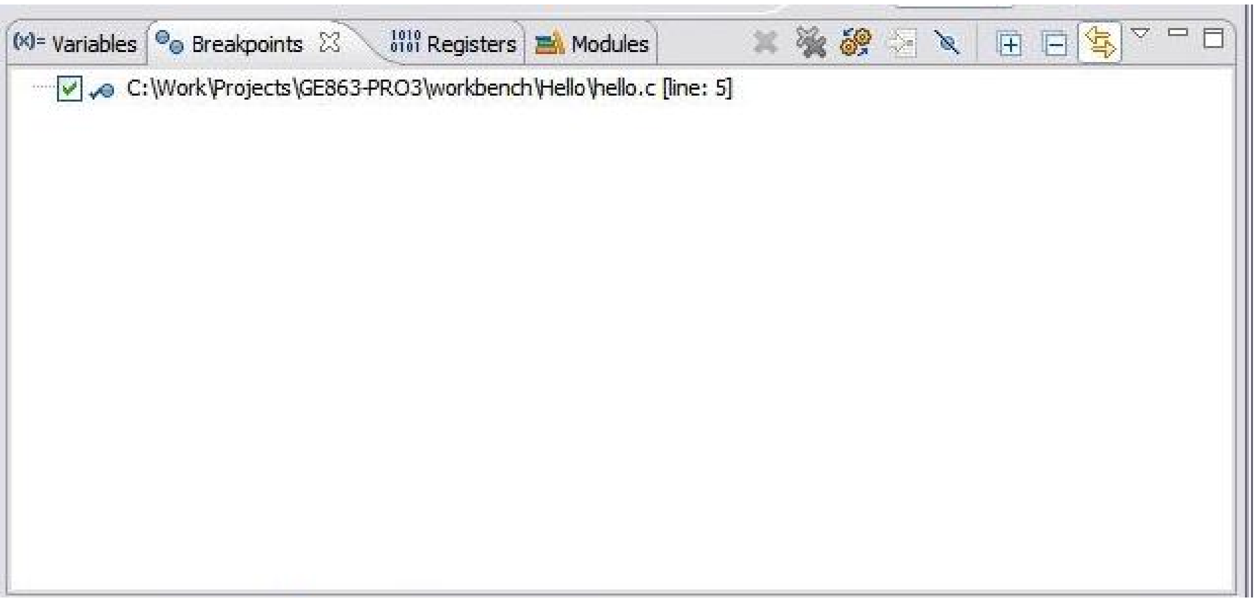
To start the debugging select the line Embedded GDB under the project name debug configuration row and click the green arrow highlighted in the screenshot.

Now you are ready to debug your application. To set breakpoints double-click on the far left edge of the line. Double-clicking on the same spot will remove it.



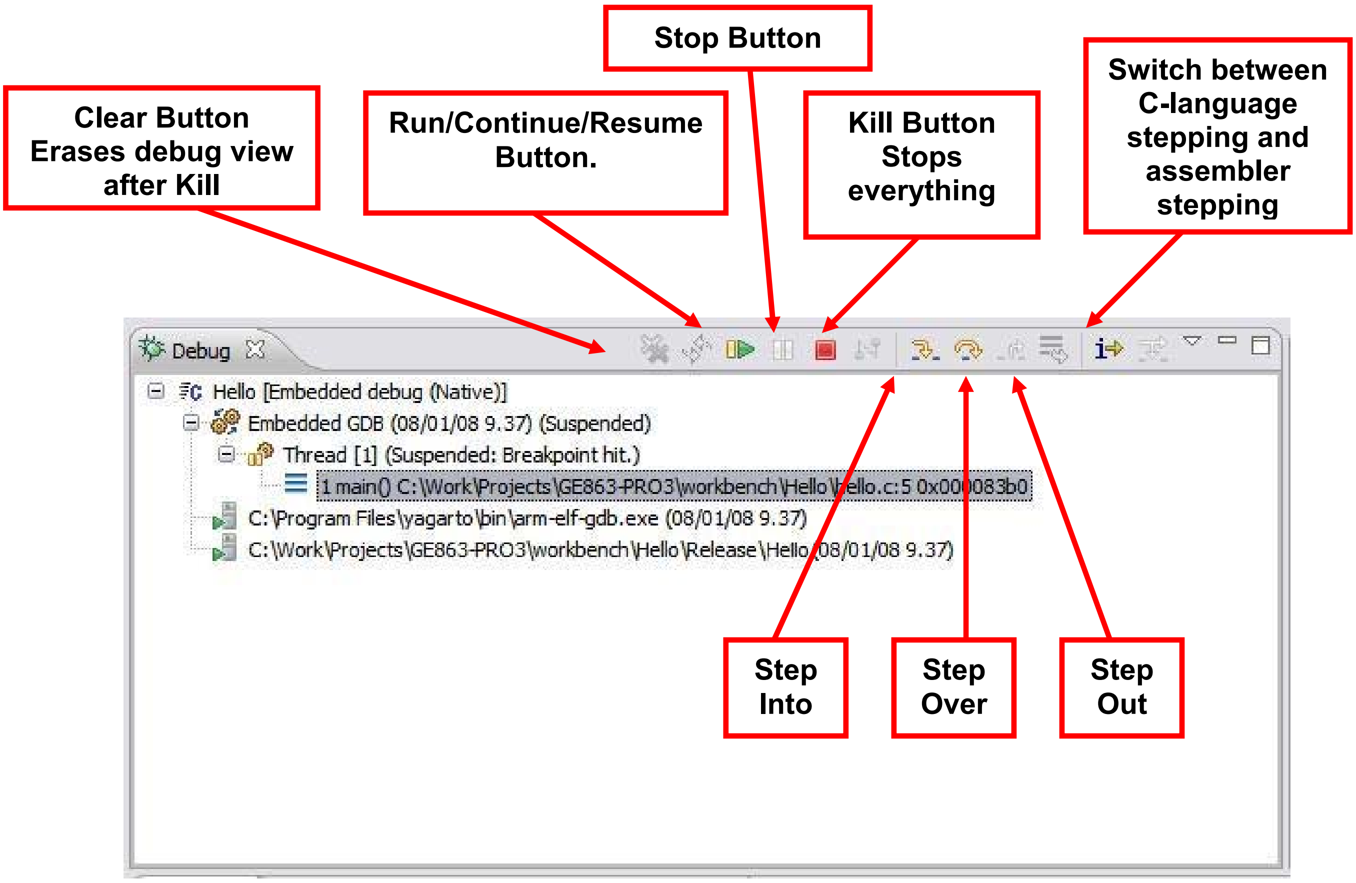
```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("\nHello world!\n");
6
7     return 0;
8 }
9
```

Note in the upper right “Breakpoint Summary” pane, the new breakpoint at line 5 has been indicated, as shown below.



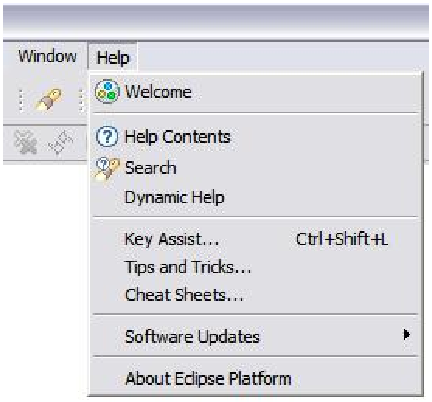
☑ C:\Work\Projects\GE863-PRO3\workbench\Hello\hello.c [line: 5]

Use the debug control buttons shown below to control debugging.


Stop Button
Clear Button Erases debug view after Kill
Run/Continue/Resume Button.
Kill Button Stops everything
Switch between C-language stepping and assembler stepping
Debug
Step Into
Step Over
Step Out

```
Hello [Embedded debug (Native)]
  Embedded GDB (08/01/08 9.37) (Suspended)
    Thread [1] (Suspended: Breakpoint hit.)
      1 main() C:\Work\Projects\GE863-PRO3\workbench\Hello\hello.c:5 0x000083b0
  C:\Program Files\yagarto\bin\arm-elf-gdb.exe (08/01/08 9.37)
  C:\Work\Projects\GE863-PRO3\workbench\Hello\Release\Hello (08/01/08 9.37)
```

See Eclipse’s Help Contents for further information.



8 General suggestions

- Using a workspace in Windows filesystem it is not possible to rename an existing project: if you try the project name changes but connection to the files included is lost.
- Using a workspace in Windows filesystem it is not possible to delete a project: if you try the project seems to be deleted in the workspace, but its files are still present in the filesystem.
- Using a workspace in Windows filesystem sometimes Eclipse reports error in preferences saving, however this is only a warning.
- If Eclipse reports error in saving files containing special characters you have to change text enconding preferences. Go to Windows → Preferences → General → Workspace → Text File Encoding and choose, for example, ISO-8859-1.
- Occasional Xming crashes happen: close coLinux and restart it to solve the problem.
- Using Ethernet on USB with a Windows bridge (ad described in chapter 4) causes ARP problems, i.e. the target doesn't succeed in communicating with other systems if those haven't previously sent data to it. This problem seems to be related in the RNDIS driver: a workaround is to manually insert entries in ARP table (for example in a startup script) for linking a NIC mac address to its IP address. The shell command to be used is:

  ```
  arp –s hostname hw_address
  ```

- If the software "Putty" is present in your system be sure that its default configuration is ssh and not serial, otherwise Telit Development Environment applications won't start.

Appendix A: How to change default IP addresses

Telit Development Environment default installation will create a C class network 192.168.121.0; however it is possible that you already have in your system such a network. In this case, you have to manually change all the 192.168.121.x addresses to a free address space in your network. Follow these steps:

- Start coLinux.
- Go to Telit Development Environment installation directory and run *colinux-console-fltk.exe*.
- After login (user: root, password: root) type in the console:

  ```
  nano /etc/network/interfaces
  ```

  and modify the rows

  ```
  address 192.168.121.2
  netmask 255.255.255.0
  ```

  with the proper addresses, then save the file.

  Pay attention that you always need to keep the correct correspondence between addresses.

- Type in the console:

  ```
  nano /root/.profile
  ```

  and modify the row

  ```
  DISPLAY=192.168.121.1:0.0
  ```

  with the proper address, then save the file.

- Type in the console:

  ```
  halt
  ```

  then close the coLinux window.

- Go to Telit Development Environment installation directory, enter in the Xming directory and open the file X0.hosts with a text editor. Modify the rows

  ```
  192.168.121.2
  192.168.121.1
  ```

  with the proper addresses, then save the file.

- Modify the IP address of the network bridge according to the proper value (see paragraph 4.1.3 for details).

- Go to **Start menu** → **All Programs** → **Telit Development Platform** and right click on the item Console: modify the addresses `192.168.121.2` and `192.168.121.1:0.0` with the proper values, then click Ok. Repeat the same procedure for the items Telit Customized Eclipse, mc and Synaptic.

- Start coLinux.

- Launch Telit Customized Eclipse.

- Go to **Window** → **Preferences** and click on the item Telit GE863 Preferences. Modify the addresses in coLinux IP address and GE863 IP address text boxes with the proper values.

- When configuring the target IP address remember to use the new value; see paragraph 4.1.2 for details.

Appendix B: How to make a back-up

If you wish to make a back-up of the entire development environment go to its installation directory and copy the following files:

- root_fs.img
- swap.img

If you want to restore the system to this back-up, copy back the two files to the development environment installation directory.

⚠ Please note that if your Eclipse workspace is in Windows filesystem this won't be backed-up.

Appendix C: Recovering from a serious error

If the development environment doesn't work anymore you can follow these steps to restore the system.

⚠ **THIS METHOD COMPLETELY DESTROYS ALL THE DATA YOU HAVE STORED INSIDE COLINUX FILESYSTEM. USE THIS ONLY IF YOU HAVE NOT ANY OTHER SOLUTION TO RESTORE THE SYSTEM!**

- If running, close coLinux.
- Go to Telit Development Environment installation directory.
- Double click on the file dskimageC_Cpp_dd_mm_yyyy.exe where dd_mm_yyyy is a date.
- Click **Extract**.

After several minutes the development environment will be restored. If this method doesn't work uninstall the entire development environment (including coLinux and Xming), delete the installation directory and try to reinstall it following the steps reported in 3.3.

Appendix D: Copying C++ libraries in the target

By default the target does not have the libraries necessary for C++ applications. In order to copy them follow these steps:

- Start coLinux (Start menu → All Programs → Telit Development Platform → Start Colinux).
- Open a console (Start menu → All Programs → Telit Development Platform → Console).
- In the console type:

```
mkdir /root/cpp_libraries
cd /opt/crosstools/usr/arm-linux-uclibc/lib/
arm-linux-objcopy --strip-debug libstdc++.so.6
/root/cpp_libraries/libstdc++.so.6
arm-linux-objcopy --strip-debug libgcc_s.so.1 /root/
cpp_libraries/libgcc_s.so.1
```

- Copy the two files (`/root/cpp_libraries/libstdc++.so.6` and `/root/cpp_libraries/libgcc_s.so.1`) in the target using one of the methods described in previous chapters.
- Move the two files in the target to the directory `/lib`.

Appendix E: Upgrading the firmware kernel sources with Kernel Upgrade Tool

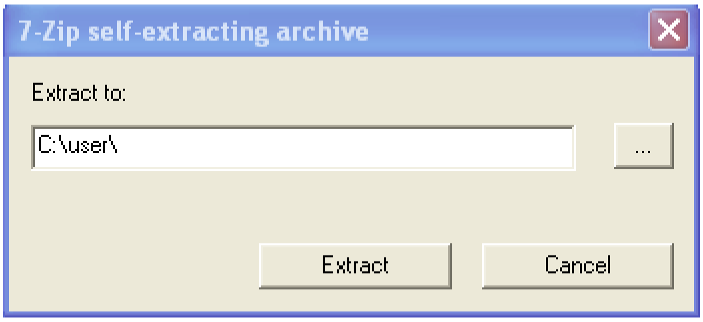
Overview

The Kernel Upgrade Tool is a Windows application used for upgrading kernel Linux in coLinux development environment. The updated kernel version depends on the tool version.

Usage

For using the Kernel Upgrade Tool follow these steps:

- Double click on the file Telit-UpgradeKernel-<version_number>.exe and choose the directory where the tool can be unpacked.



7-Zip self-extracting archive

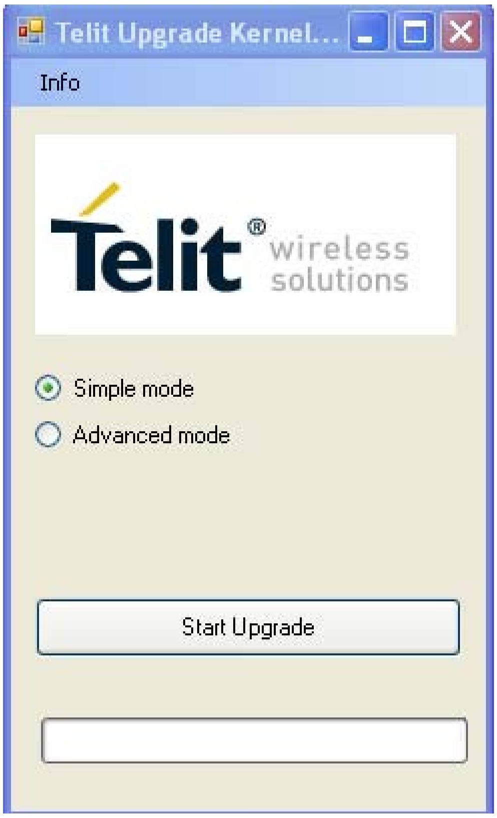
Extract to: C:\user\

- Go to the directory where the tool has been unpacked and double click on the file

  UpgradeKernel.exe

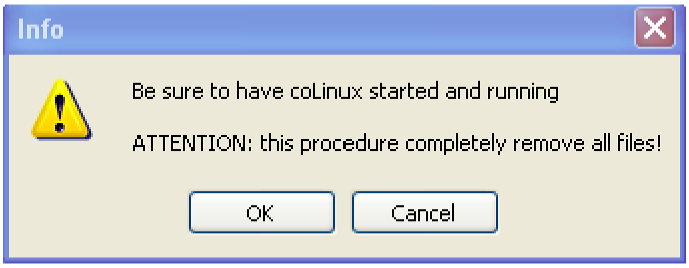
  for launching the tool.

- If you have kept the default ip addresses for coLinux and the target then simply click the **Start Upgrade** button:



● Simple mode
○ Advanced mode

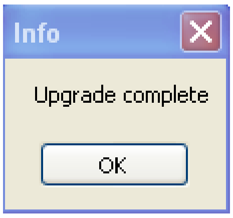
- When you are ready confirm the action clicking Ok, otherwise Cancel:



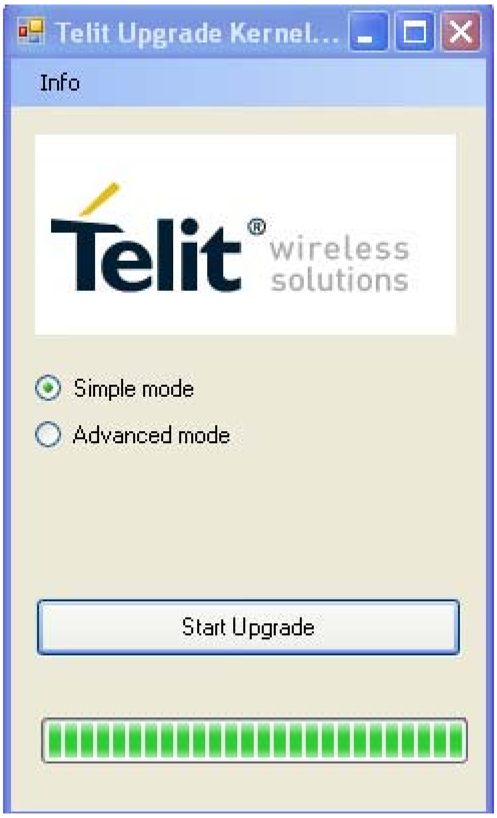
Be sure to have coLinux started and running

ATTENTION: this procedure completely remove all files!

- The upgrade takes some seconds. If the update succeeds it is shown the following pop-up:

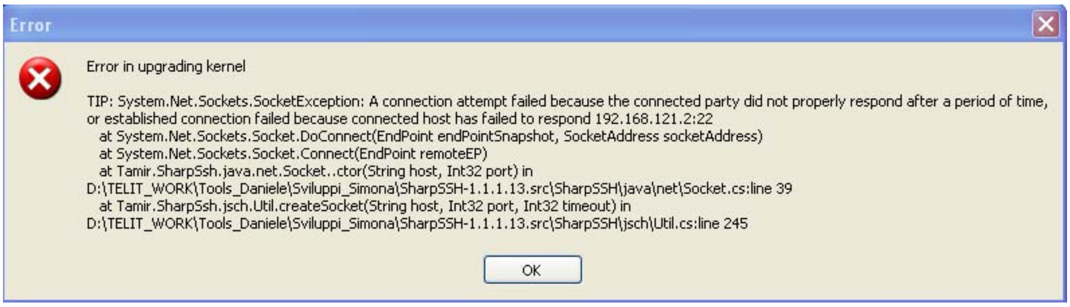


Upgrade complete



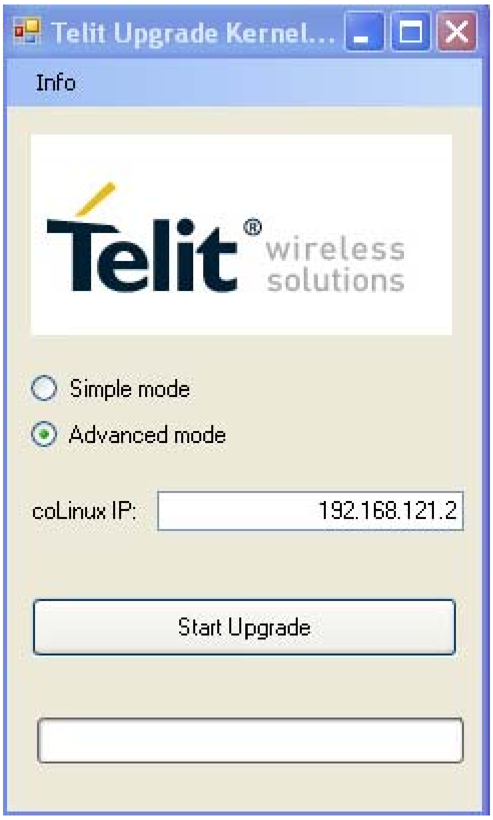
● Simple mode
○ Advanced mode

- If the update doesn't succeed a pop-up like this will appear:



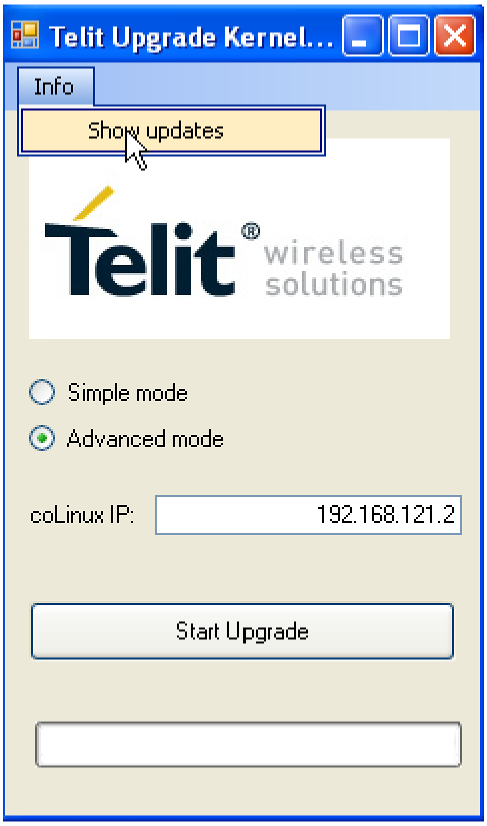
```
Error in upgrading kernel

TIP: System.Net.Sockets.SocketException: A connection attempt failed because the connected party did not properly respond after a period of time,
or established connection failed because connected host has failed to respond 192.168.121.2:22
   at System.Net.Sockets.Socket.DoConnect(EndPoint endPointSnapshot, SocketAddress socketAddress)
   at System.Net.Sockets.Socket.Connect(EndPoint remoteEP)
   at Tamir.SharpSsh.java.net.Socket..ctor(String host, Int32 port) in
D:\TELIT_WORK\Tools_Daniele\Sviluppi_Simona\SharpSSH-1.1.1.13.src\SharpSSH\java\net\Socket.cs:line 39
   at Tamir.SharpSsh.jsch.Util.createSocket(String host, Int32 port, Int32 timeout) in
D:\TELIT_WORK\Tools_Daniele\Sviluppi_Simona\SharpSSH-1.1.1.13.src\SharpSSH\jsch\Util.cs:line 245
```

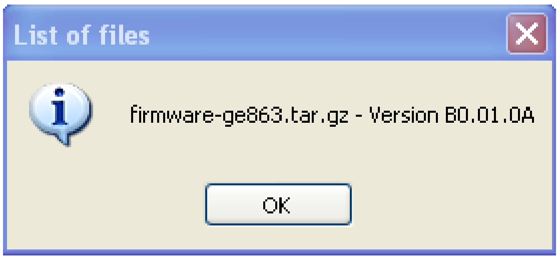
- If you have not kept the default ip addresses for coLinux and the target then choose the **Advanced Mode** of the Radio Button and insert the proper ip address in the coLinux ip text field, then click the **Start Upgrade** button:



coLinux IP: 192.168.121.2

- If you want to see which updates the tool carry out click on **Info** → **Show Updates**:





firmware-ge863.tar.gz - Version B0.01.0A

•

Appendix F: Upgrading U-boot with U-boot upgrade Tool

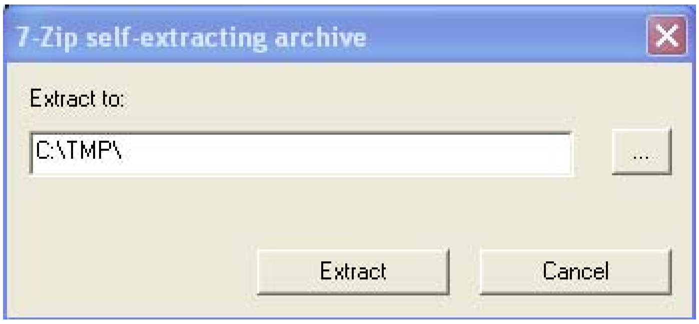
Overview

The u-boot Upgrade Tool is a Windows application used for upgrading u-boot in coLinux development environment. The updated u-boot version depends on the tool version.

Usage

For using the u-boot Upgrade Tool follow these steps:

- Double click on the file Telit-UpgradeU-boot_<version_number>.exe and choose the directory where the tool can be unpacked.

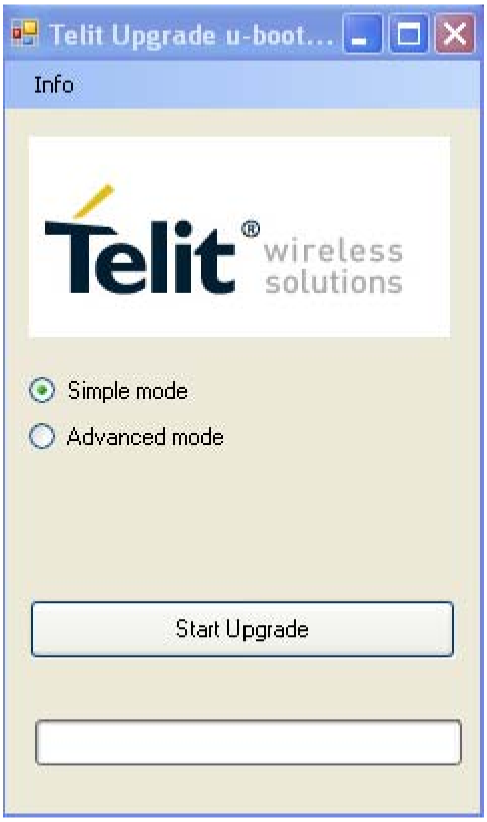


- Go to the directory where the tool has been unpacked and double click on the file

  UpgradeU-boot.exe

  for launching the tool.

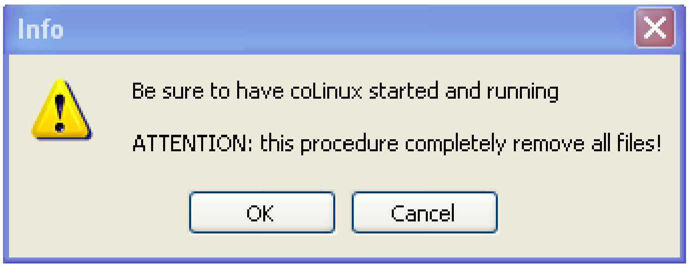
- If you have kept the default ip addresses for coLinux and the target then simply click the **Start Upgrade** button:



● Simple mode

○ Advanced mode

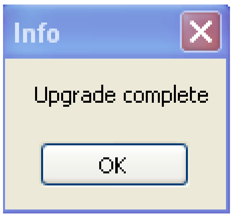
- When you are ready confirm the action clicking Ok, otherwise Cancel:



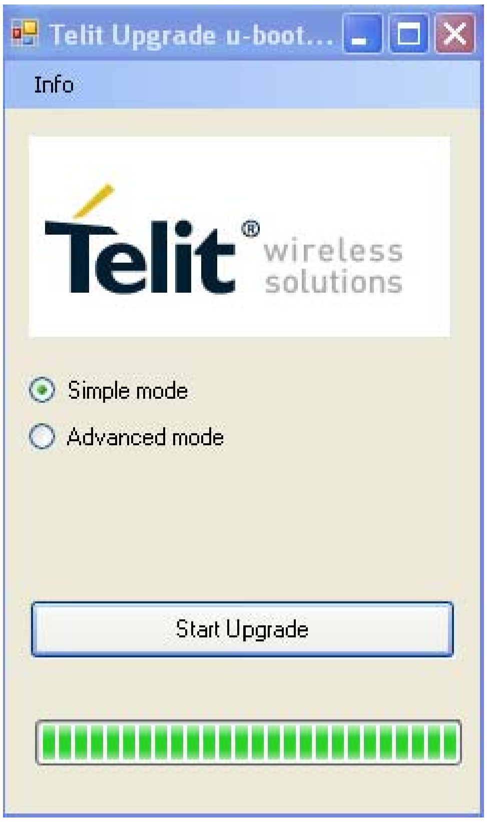
Be sure to have coLinux started and running

ATTENTION: this procedure completely remove all files!

- The upgrade takes some seconds. If the update succeeds it is shown the following pop-up:

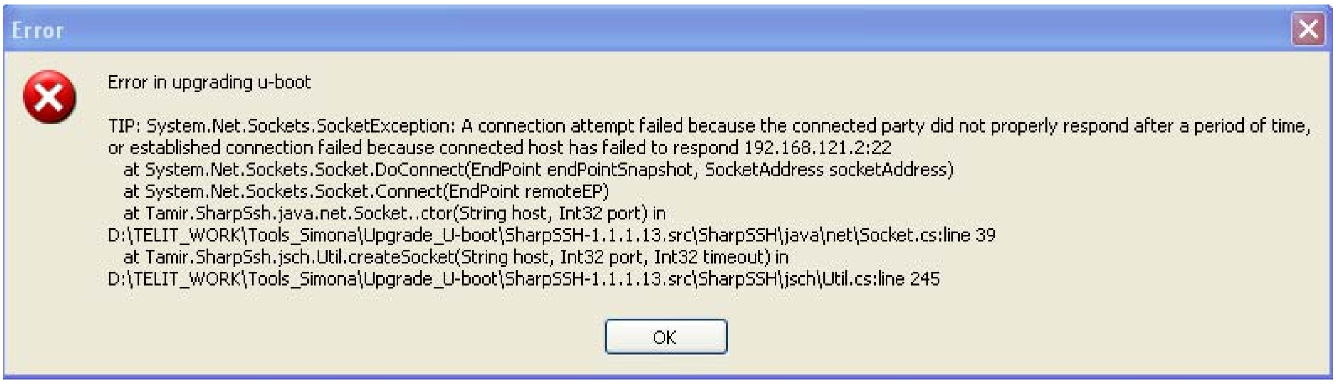


Upgrade complete



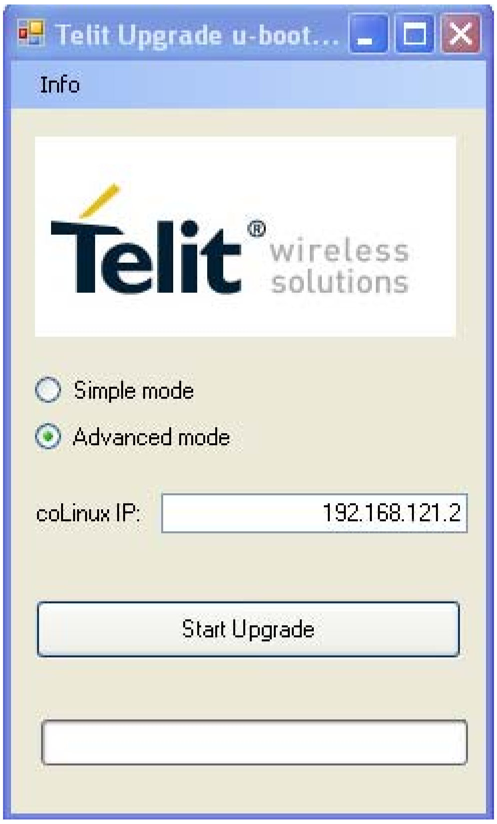
Telit Upgrade u-boot...
Info
Telit wireless solutions
● Simple mode
○ Advanced mode
Start Upgrade

- If the update doesn’t succeed a pop-up like this will appear:



```
Error in upgrading u-boot

TIP: System.Net.Sockets.SocketException: A connection attempt failed because the connected party did not properly respond after a period of time,
or established connection failed because connected host has failed to respond 192.168.121.2:22
   at System.Net.Sockets.Socket.DoConnect(EndPoint endPointSnapshot, SocketAddress socketAddress)
   at System.Net.Sockets.Socket.Connect(EndPoint remoteEP)
   at Tamir.SharpSsh.java.net.Socket..ctor(String host, Int32 port) in
D:\TELIT_WORK\Tools_Simona\Upgrade_U-boot\SharpSSH-1.1.1.13.src\SharpSSH\java\net\Socket.cs:line 39
   at Tamir.SharpSsh.jsch.Util.createSocket(String host, Int32 port, Int32 timeout) in
D:\TELIT_WORK\Tools_Simona\Upgrade_U-boot\SharpSSH-1.1.1.13.src\SharpSSH\jsch\Util.cs:line 245
```

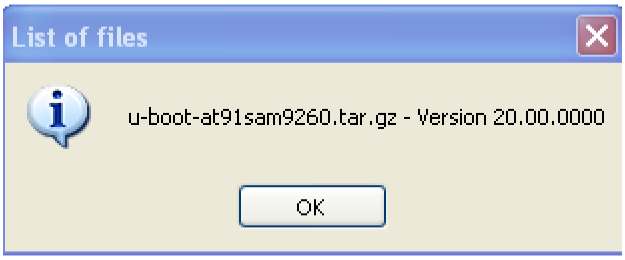
- If you have not kept the default ip addresses for coLinux and the target then choose the **Advanced Mode** of the Radio Button and insert the proper ip address in the coLinux ip text field, then click the **Start Upgrade** button:



○ Simple mode
● Advanced mode

coLinux IP: 192.168.121.2

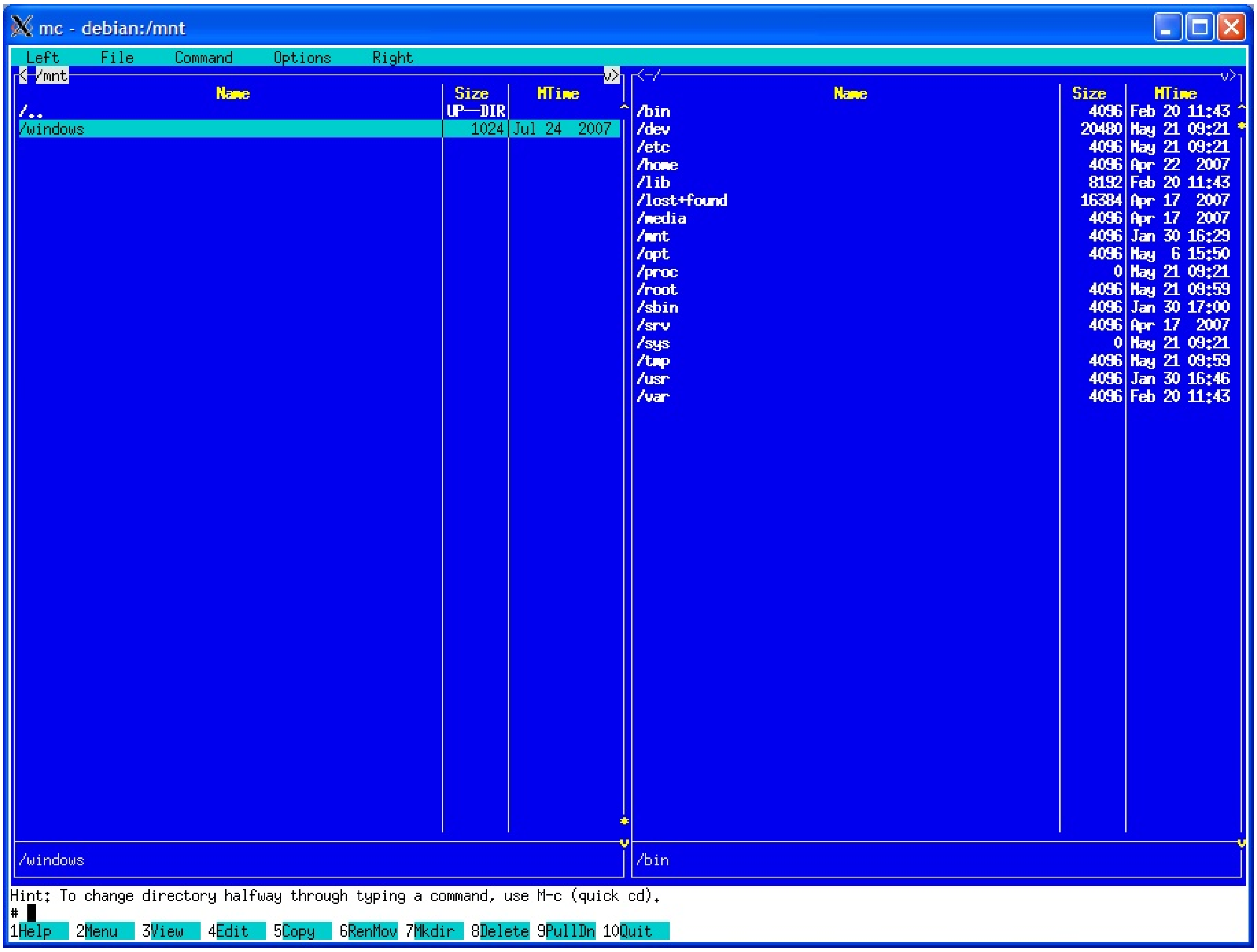
- If you want to see which updates the tool carried out click on **Info** → **Show Updates**:



u-boot-at91sam9260.tar.gz - Version 20.00.0000

Appendix G: Transferring files between coLinux and Windows

For transferring files between coLinux and Windows filesystems it is possible to use a program called Midnight Commander (Start menu → All Programs → Telit Development Platform → Midnight Commander).



```
/mnt                                                    /
Name                Size    MTime          Name              Size   MTime
/..                 UP--DIR                /bin              4096   Feb 20 11:43
/windows            1024    Jul 24  2007   /dev              20480  May 21 09:21
                                           /etc              4096   May 21 09:21
                                           /home             4096   Apr 22  2007
                                           /lib              8192   Feb 20 11:43
                                           /lost+found       16384  Apr 17  2007
                                           /media            4096   Apr 17  2007
                                           /mnt              4096   Jan 30 16:29
                                           /opt              4096   May  6 15:50
                                           /proc             0      May 21 09:21
                                           /root             4096   May 21 09:59
                                           /sbin             4096   Jan 30 17:00
                                           /srv              4096   Apr 17  2007
                                           /sys              0      May 21 09:21
                                           /tmp              4096   May 21 09:59
                                           /usr              4096   Jan 30 16:46
                                           /var              4096   Feb 20 11:43

/windows                                   /bin
Hint: To change directory halfway through typing a command, use M-c (quick cd).
#
```

It works as a normal file explorer: the window is divided into two columns, each one showing the content of a directory in the filesystem. Midnight Commander allows executing all normal operations on files (copying, renaming, deleting…). You can find the content of your Windows filesystem inside /mnt/windows. For help press F1.