TLL6219 ARM9 Module

The Learning Labs, Inc.

Getting Started Manual, v1.2

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of TLL.

TLL provides this document as is, without warranty of any kind, expressed or implied, including, but not limited to, the particular purpose. TLL may make improvements and/or changes in this manual or in the products(s) and/or the program(s) described in this manual at any time.

Information in this manual is intended to be accurate and reliable. However, TLL assumes no responsibility for its use, or for any infringements of rights of other parties, which may result from its use.

This document could include technical or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication. This manual is provided solely and exclusively for educational use and this information or related products should not be used nor relied upon for any purpose except for education and training.

Technical Support

Please contact your local TLL authorized product representative for questions regarding hardware, software or applications issues. Any updates or patches will be sent to you automatically as long as your registration is current. The TLL products are designed to be supported remotely by allowing viewing of the user’s desktop. It is highly recommended that the PC from which you are using TLL products is connected to an Internet link that allows Web browser access. In this way our technical support staff can view your desktop and work with you to understand and solve technical issues.

Table of Contents

Chapter 1

Prerequisites

System Requirements:

The following are the prerequisites for the Computers to run required software tools for ARM Mezzanine applications. Make sure that these components or minimum hardware is present.

- Pentium 1 GHz or higher -- Although lower processors will run the software tools, the desired results are not guaranteed.
- 256MB RAM – for average applications.
- Minimum 5-GB on Hard Disk – The Webpack ISE requires around 3.5 GB for all its components.
- SVGA Monitor with the resolution of 1024x768 -- Monitor resolutions has been kept at 1024x768, which is ideal.
- **Windows 2000 or XP operating system with ServicePack-4 (for Windows 2000) & ServicePack-2 (for WinXP)**
- Three USB2.0 ports
- RS232 serial port – 1 No
- Network Interface Card (NIC) The network interface card should have been configured on your system. Make sure that you have configured the network card with a static IP address.

Software requirements:

The following software need to be installed in the PC used for TLL development platform.

- Open Source tools for ARM-ELF compatible file for uMON
- HyperTerminal for Serial Communication

Hardware requirements:

The following hardware and accessories are required to start the experimentation

- TLL6219 ARM Mezzanine module
- TLL5000 Platform.
- Power supply for TLL5000 Platform – 18.5 volts / 3.5 Amps
- USB cables – 2 Nos.
- Platform Cable – USB (This converts USB to JTAG).
- CD provided by TLL.

The TLL-5000 development platform CD contains

Sl. no	Directory	Description	Remarks
1	Tools	ARM tools for Compilation	This installation provides the tools for ARM9 architecture
2	EXERCISES	Exercises for C coding using MicroMonitor Instructions	The Micromonitor is bootup Environment for Embedded Platforms
4	Manuals	Contains the ARM Mezzanine manual, Getting Started Micro Monitor manual & Detailed Micromonitor user manual	The documents are in .pdf format and the same can be viewed with “Adobe Reader” application.

Table 1-1
TLL-5000 development platform CD contents

Chapter 2

Setting up of TLL5000 & TLL6219 Platforms

Figure 1-1 shows the connections to be made between the PC and TLL5000 Platform. It is strictly recommended to follow the steps as per the manual to avoid any problems to the hardware.

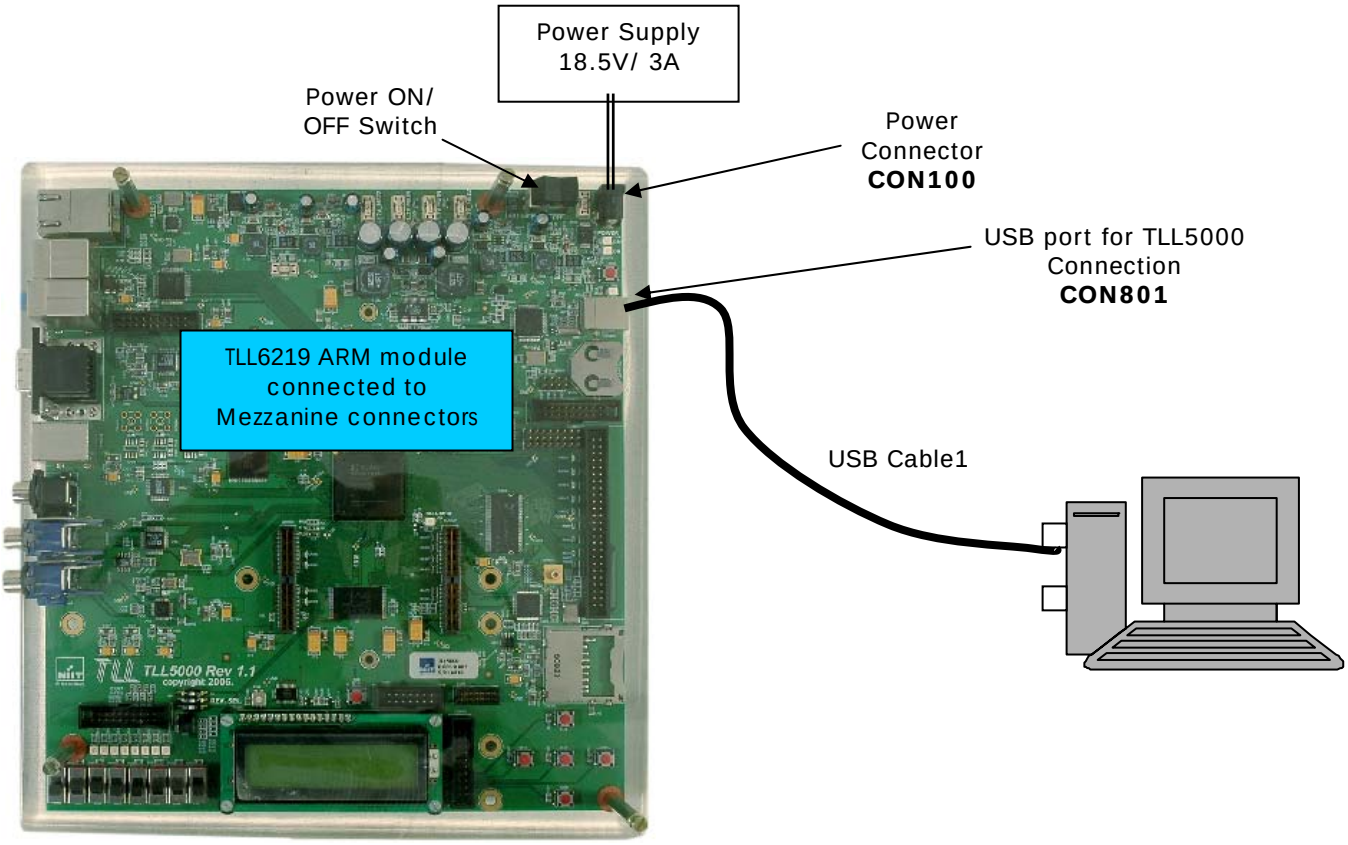

Figure 2-1
TLL6219 setup on TLL5000.

Note: For details on the TLL5000 setup and programming, refer to the getting started manual of TLL5000.

Power ON and TLL5000-Monitoring using TLL5000_PC_App.exe:

1. Connect the power supply to the power supply connector – **CON100**.
2. Connect the USB cable1 between PC-USB port & CON801 of TLL5000. This connection is used for the TLL5000 application.
3. Switch ON the power supply on the TLL5000 Platform with POWER ON/OFF Switch.
4. As the power is switched ON, the LEDs around the power connector will be ON.
5. Now click on the TLL5000_PC_App.exe application in the PC. This will bring up the following screen on the monitor.

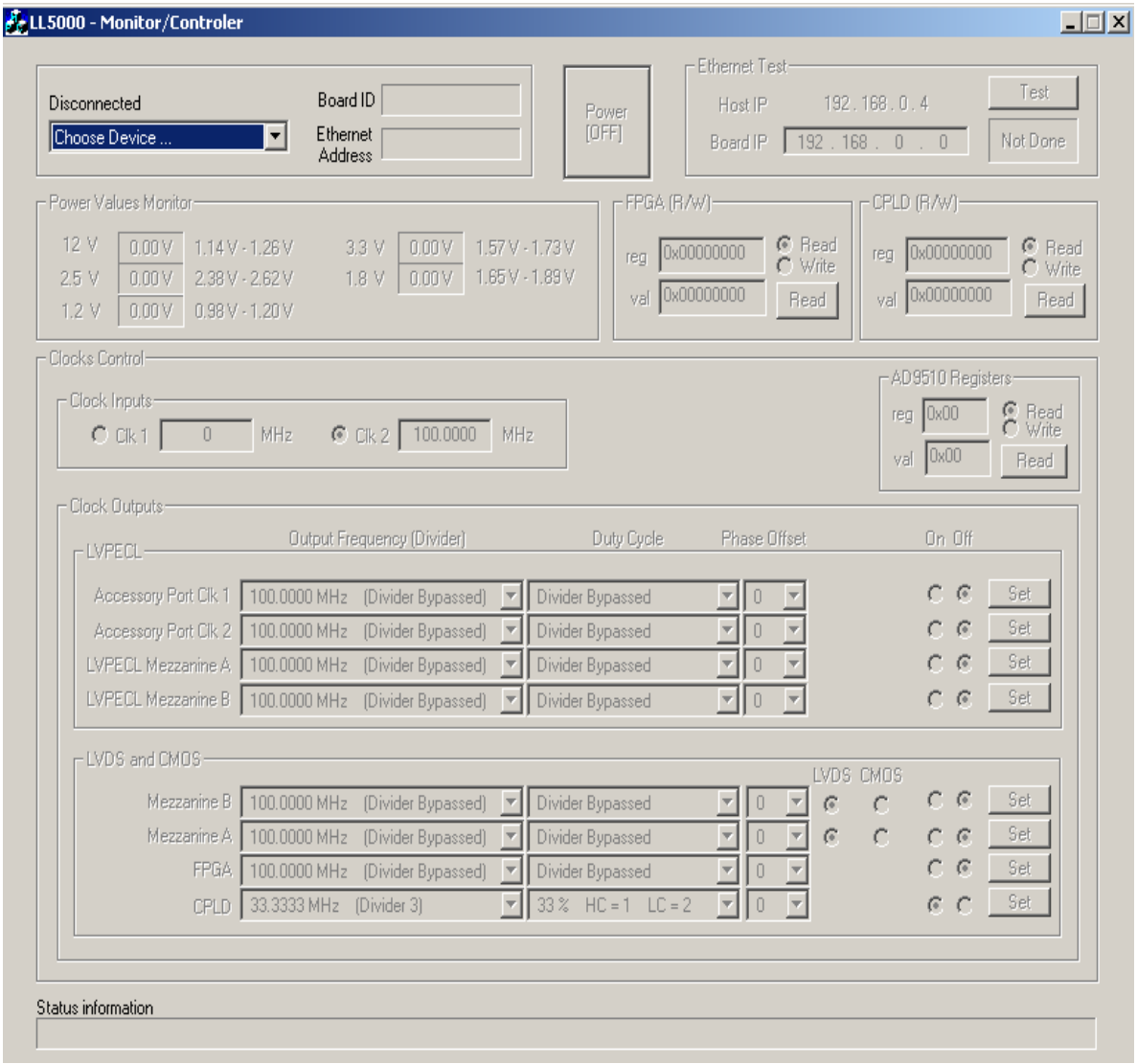

Figure 2-2
Invoking TLL5000 Application.

6. Click on the Choose device option in the step-5 screen, options will be displayed as shown below:

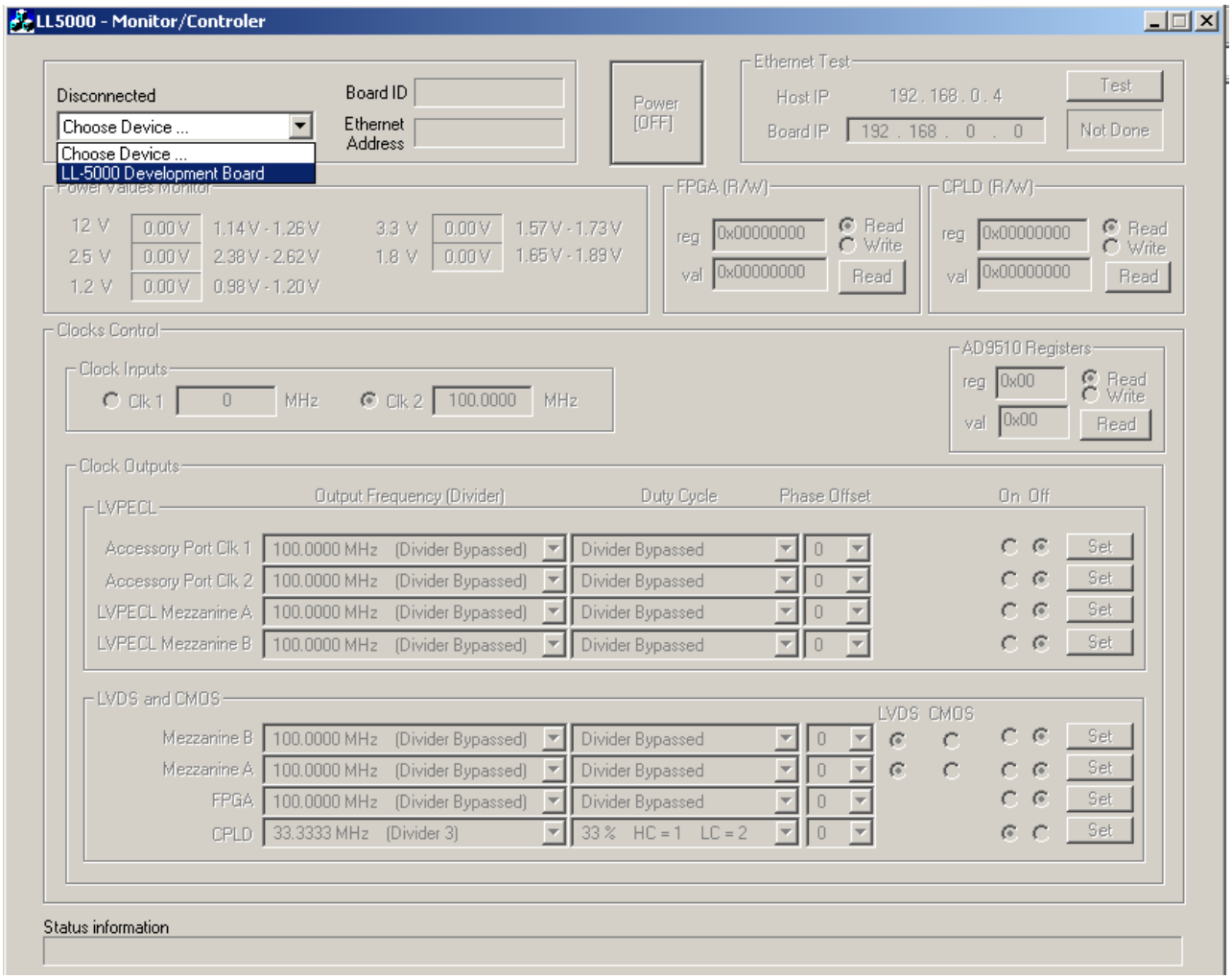

Figure 2-3
Selecting the target platform.

In the options above we shall see “TLL5000 Development Board”. This option will be enabled only with the proper USB device identification of TLL5000 platform.

7. Now click on the "**TLL5000 Development Board**" option in Step-6. The following screen appears on PC.

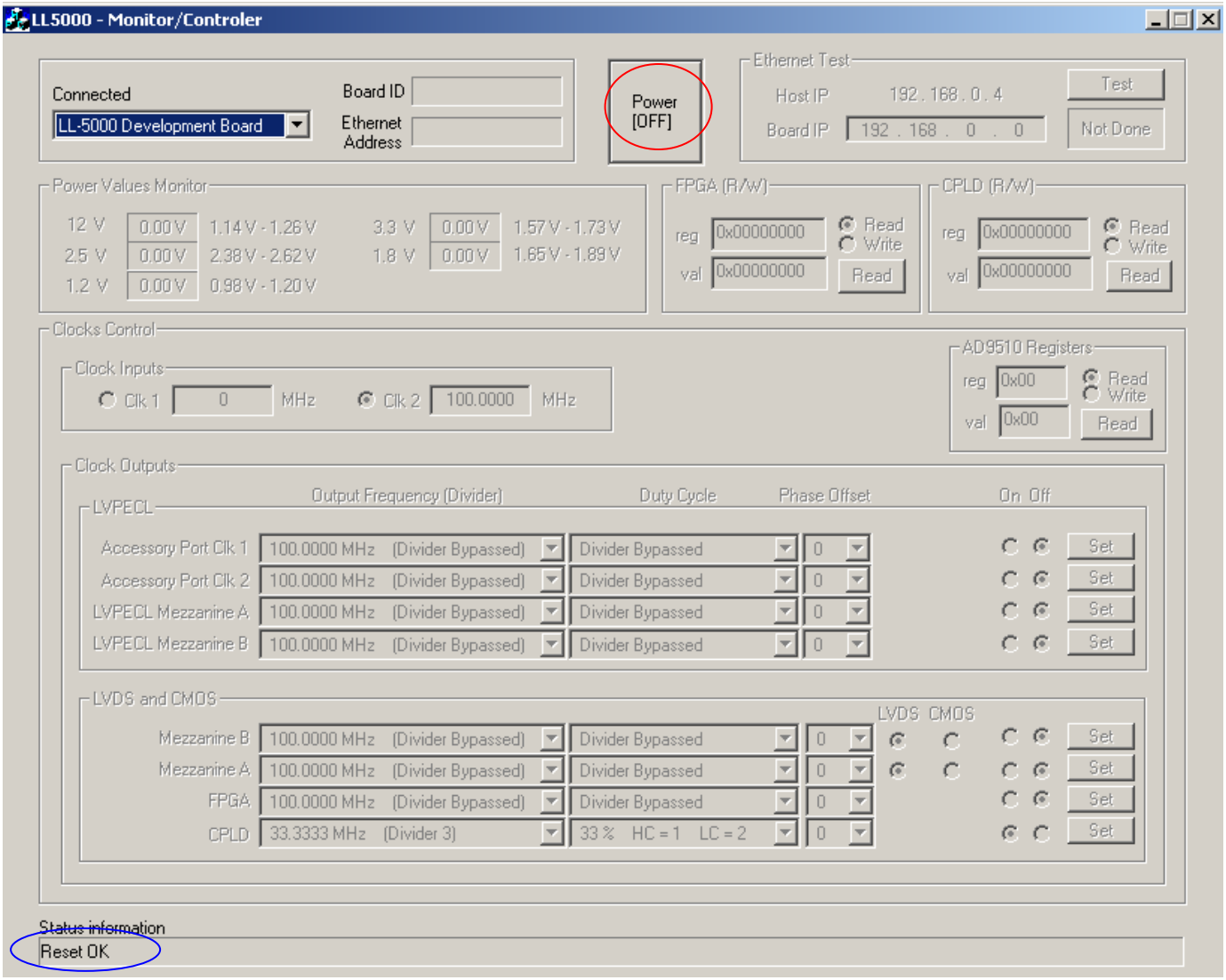

Figure 2-4
Power Up the target platform.

8. At this stage the screen displays the status of OK (see the Status information marked in blue in the above screen).

9. Now the power supply for target board components shall be started by clicking on POWER (marked in the red color) in the above screen. Now the power supply is enabled for all the parts of the board and the screen is shown as below.

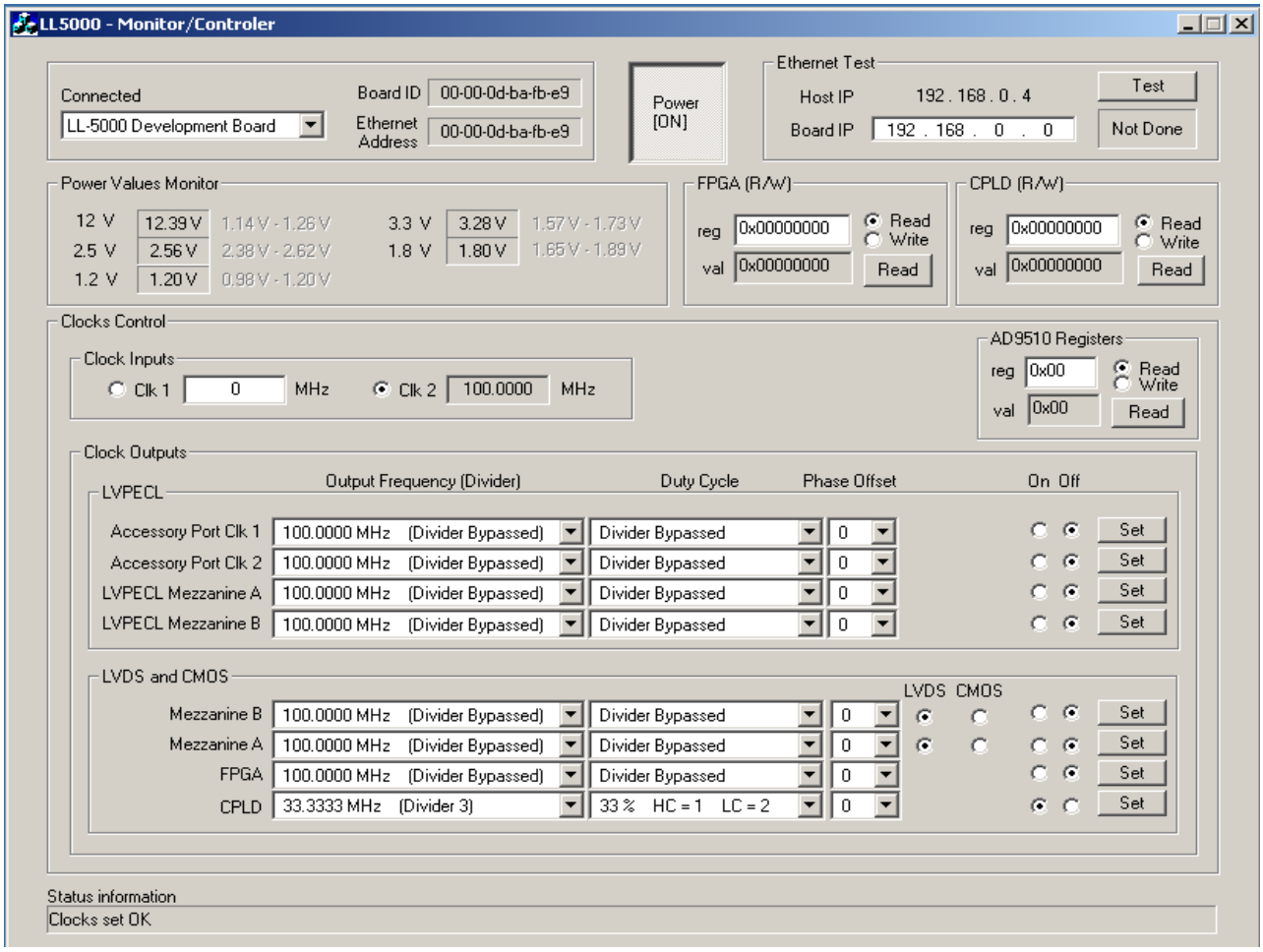

Figure 2-5
Checking the status of the target platform.

The TLL5000 application provides 3.3 volts power supply to the ARM9 Mezzanine module through Mezzanine connectors

Setting up of TLL6219 platform:

1. Connect the RS232 cable between the ARM Mezzanine module and the PC

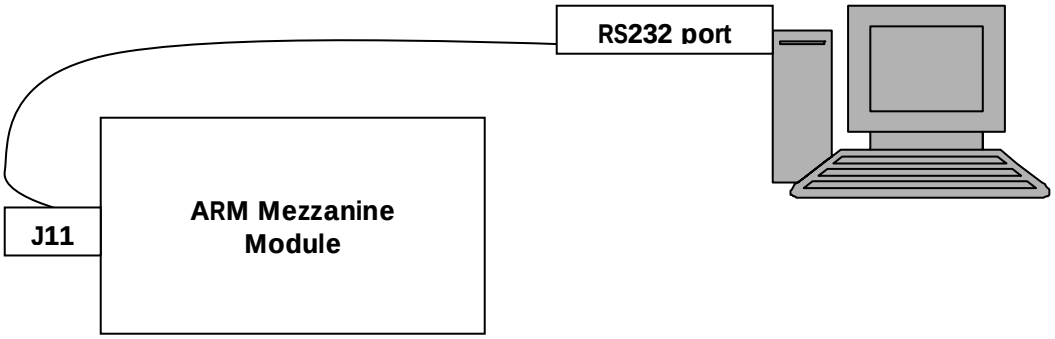

Figure 2-6
TLL6219 Setup.

2. Power on the power supply through the TLL5000 platform. The 3.3volts supplied by the base board will switch on LEDs MZ_3V3 and BB_3V3
3. Start the HyperTerminal in the PC with Baud rate 38400 bps, Data Bits – 8, Parity – None, Stop bits – 1, Flow control – None
4. Press the reset button on ARM Mezzanine board. The following screen appears on your desktop, conforming the proper communication between the PC and ARM module

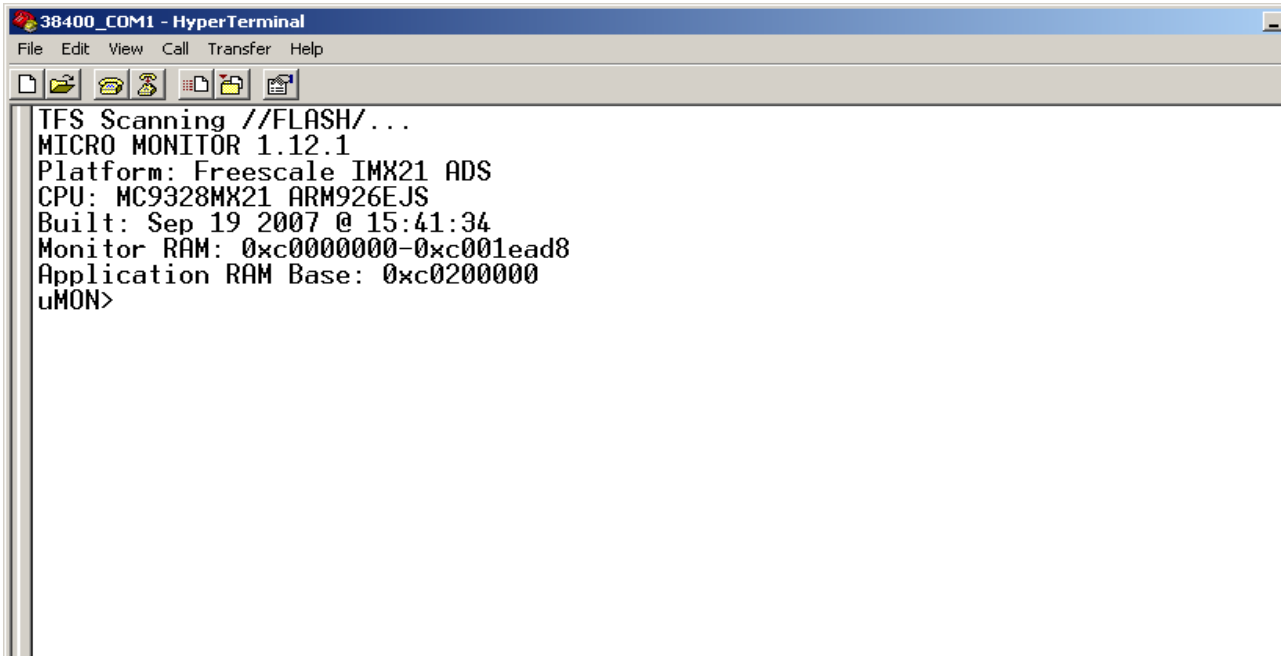

Figure 2-7
Booting up of TLL6219.

5. The above screen confirms the proper boot up of TLL6219 ARM module

Chapter 3

Downloading & Testing for TLL6219

Steps for Writing the C-Code for TLL6219 using Micromonitor Commands

Example code:

```
/* Library for utilizing the in-built Micromonitor fuctions */
#include "monlib.h"

/* Main Function */
int  main(int argc, char *argv[])
{
        mon_printf("Hello embedded world !\n");
        return(0);
}

/* Sub function for mon_connect command to transmit the data on to console */
int  start(void)
{
        int     argc;
        char    **argv;

        monConnect((int(*)())(*(unsigned long *)MONCOMPTR),(void *)0,(void *)0);

        /* Extract argc/argv from structure and call main(): */
        mon_getargv(&argc, &argv);

        /* Call main, then return to monitor. */
        return(main(argc, argv));
}
```

The above example code is for printing the data on the TLL6219 HyperTerminal console. Since we are printing the data on the HyperTerminal console, the Pre-stored data in the program is communicated to the PC through serial port. The received data in the serial port is printed on the HyperTerminal

Any Notepad or WordPad shall be used as the editor for the above code, Let us take the above the code is stored in the directory “D:\TLL6219\ARM_uMon-Examples\workspace\test” as **main1.c**

Note: The same directory contains the C codes main2.c, main3.c, main4.c, main5.c. These c codes will provide example commands for CLI, passing arguments & the same parameters shall be used for computing

Commands supported Micromonitor:

call	cm	dm	echo
edit	exit	flash	fm
gosub	goto	heap	help
?	history	if	item
mt	pm	read	reset
return	set	sleep	sm
tfs	xmodem	version	ldatags

Compilation Procedure:

- Click the I.MX tool s icon on your desktop

Figure 3-1
Invoking Eclipse Platform.

❖ The Eclipse platform brings up the IDE with linked tools chain for generating arm-elf files which shall be downloaded on to target TLL6219 platform, The Eclipse platform IDE looks as shown below

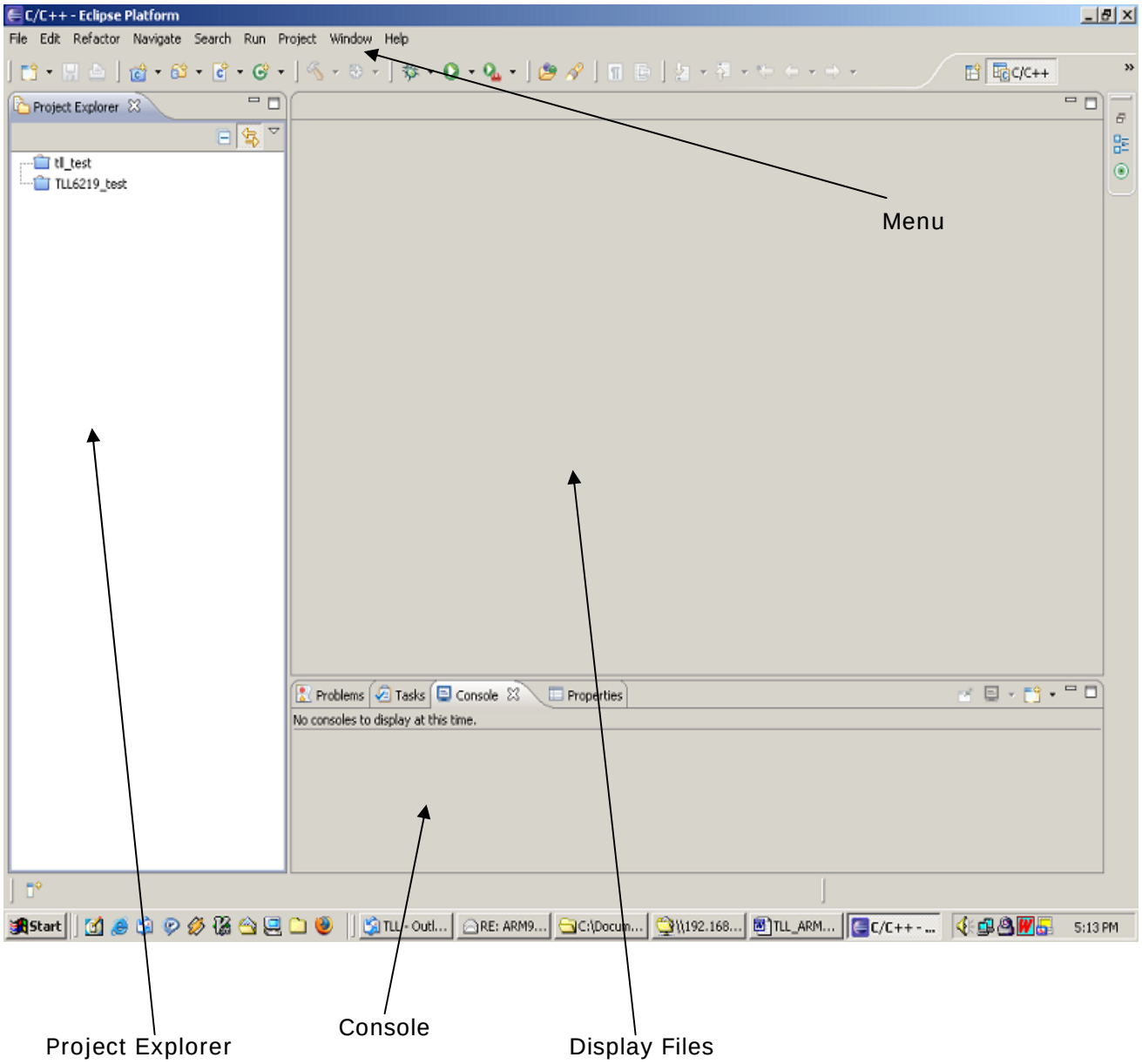

Figure 3-2
Eclipse Platform.

- The above screen shows the startup screen of Eclipse IDE.
  - The Project Explorer displays all the file names included in the project
  - These files shall be separately viewed in display window by double clicking on their names in Project explorer
  - The console displays the status and progress in each process performed with project.

Note: The default workspace for the example designs is assumed to be in "C:\TLL6219" directory

- Now click on the File → New → C Project, for stating the new project

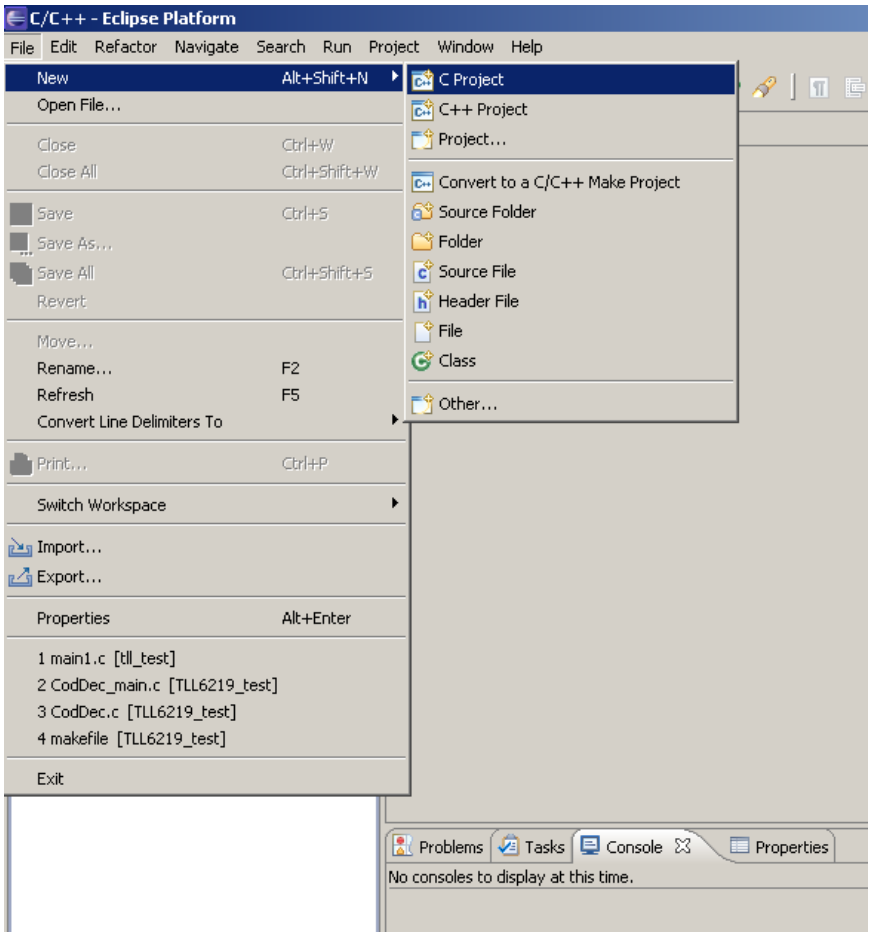

Figure 3-3
Eclipse File Menu.

❖ After clicking on New C project in above screen, the following screen appears on desktop

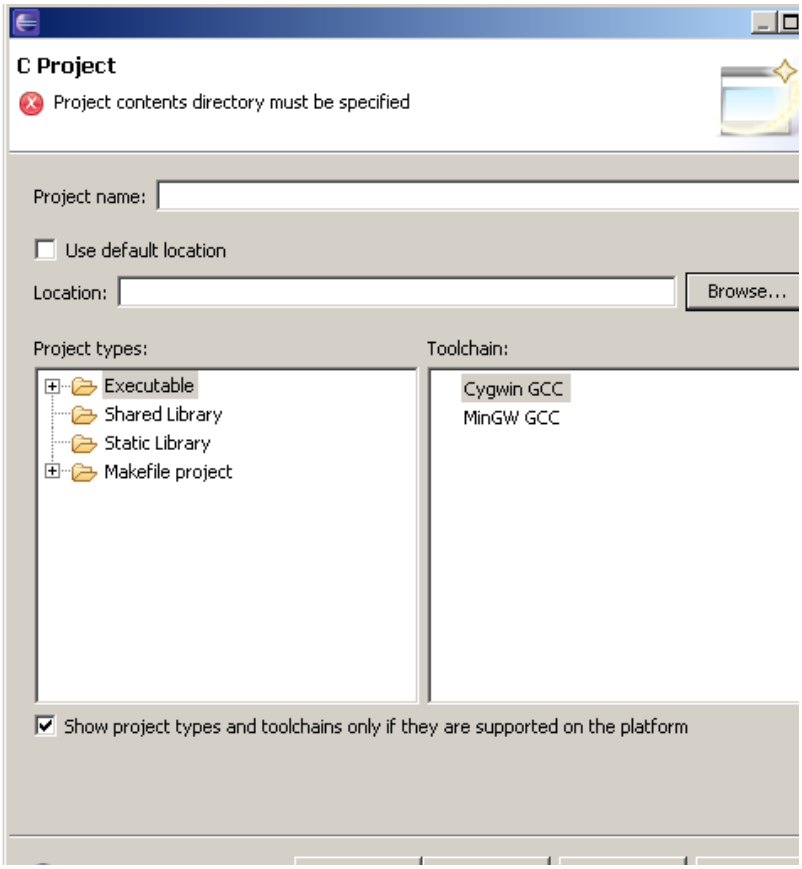

Figure 3-4
Choosing a Working Directory.

❖ In the screen shown above un check the "Use default location button" and browse to C:\TLL6219 to choose the Workspace directory

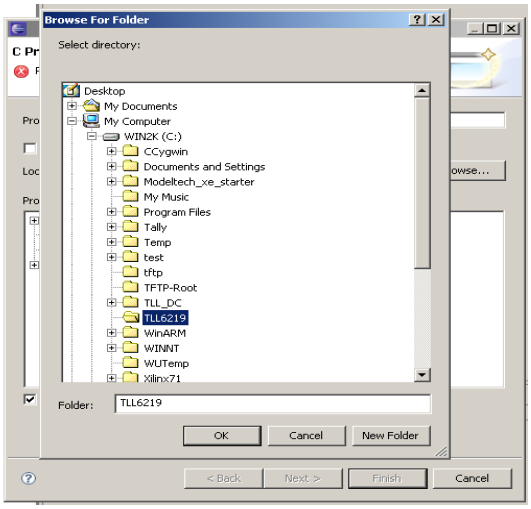

Figure 3-5
Choosing a Working Directory.

❖ After choosing the workspace directory, Enter the project name (it is considered as Test in this example)

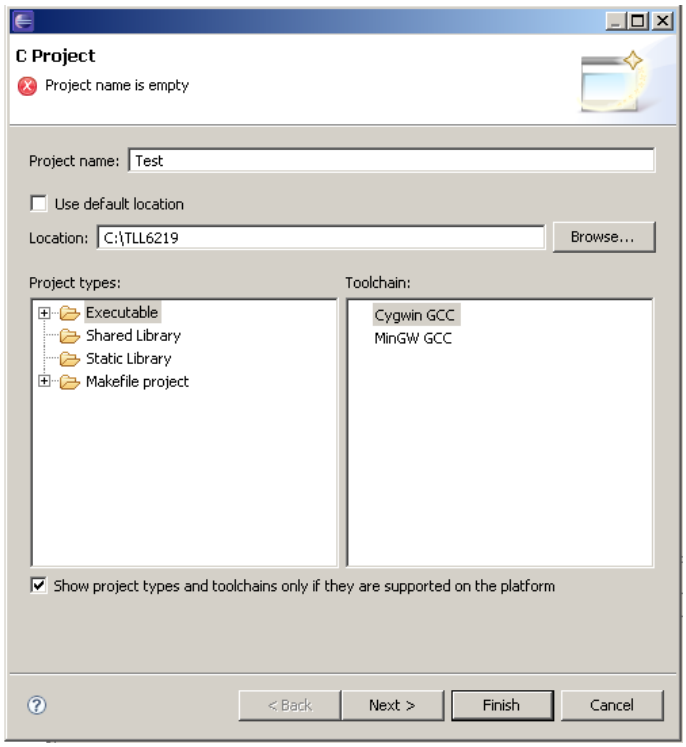

Figure 3-6
Naming a Project.

❖ Now choose the options in Project Types as Make file Project and Tool Chain as others as shown below

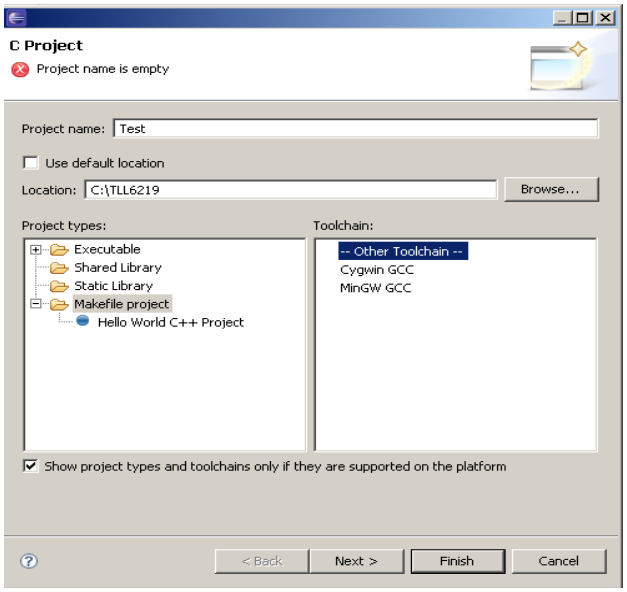

Figure 3-7
Choosing a Working Directory.

- Click on “Finish” to complete the settings. Now the control will be returned main screen.
- Add the source files to the project. In this example the source files are stored in the directory “D:\TLL6219\ARM_uMon-Examples\workspace\test”
- Now the screen looks like the following for importing the source files (Right mouse click on Project name)

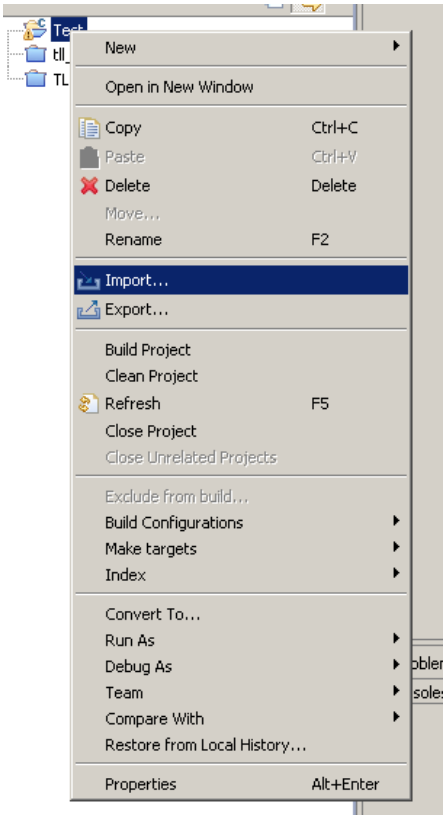

Figure 3-8
Importing Source Files.

❖ The right click on project name and import option brings up the following screen

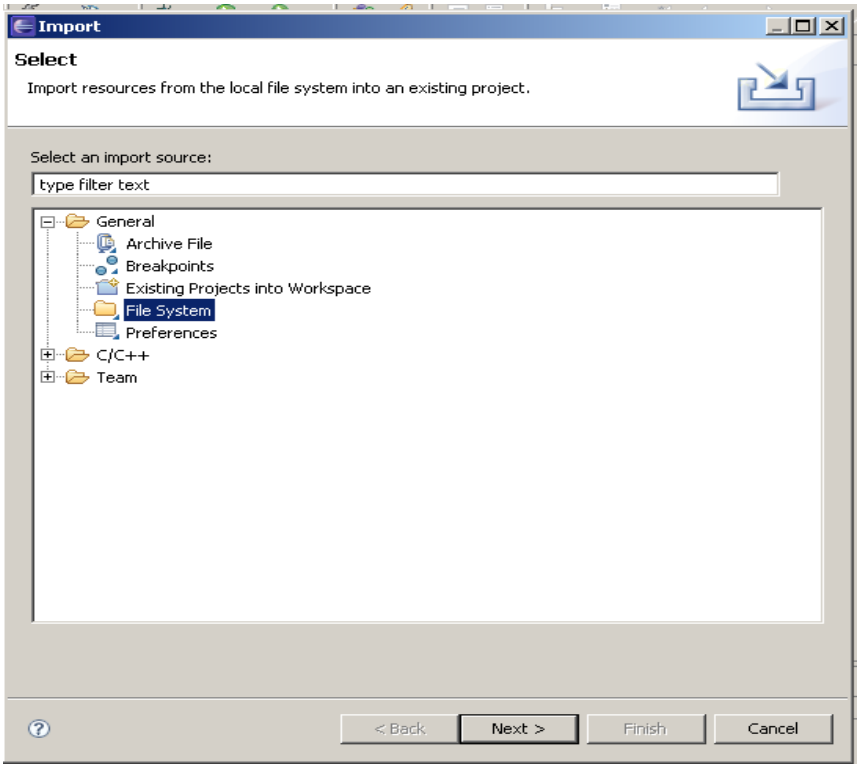

Figure 3-9
Importing Options.

❖ Choose the general and File Systems and click on the Next button to get the following screen

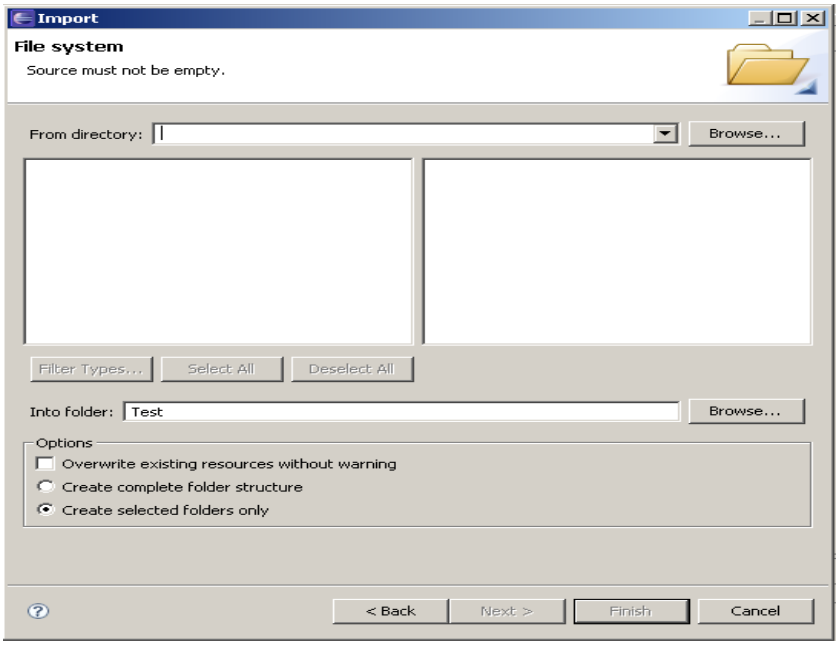

Figure 3-10
Choosing a Working Directory.

❖ Click on the browse button and browse to the path for choosing the source files "D:\TLL6219\ARM_uMon-Examples\workspace\test". The following screen appears

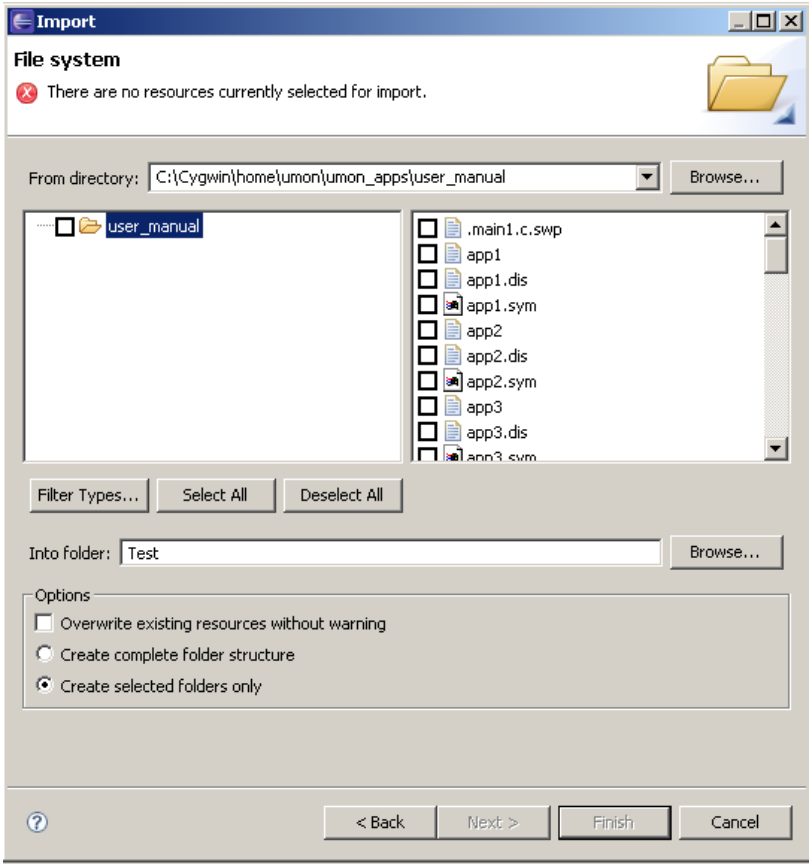

Figure 3-11
Choosing Source Files.

❖ The above screen displays all the files available in the folder, Select ALL button and then Click on Finish to add these selected files to the project.

- After the selection of source files, the control returns to the main window and all the files will be listed along with the project as shown in below screen

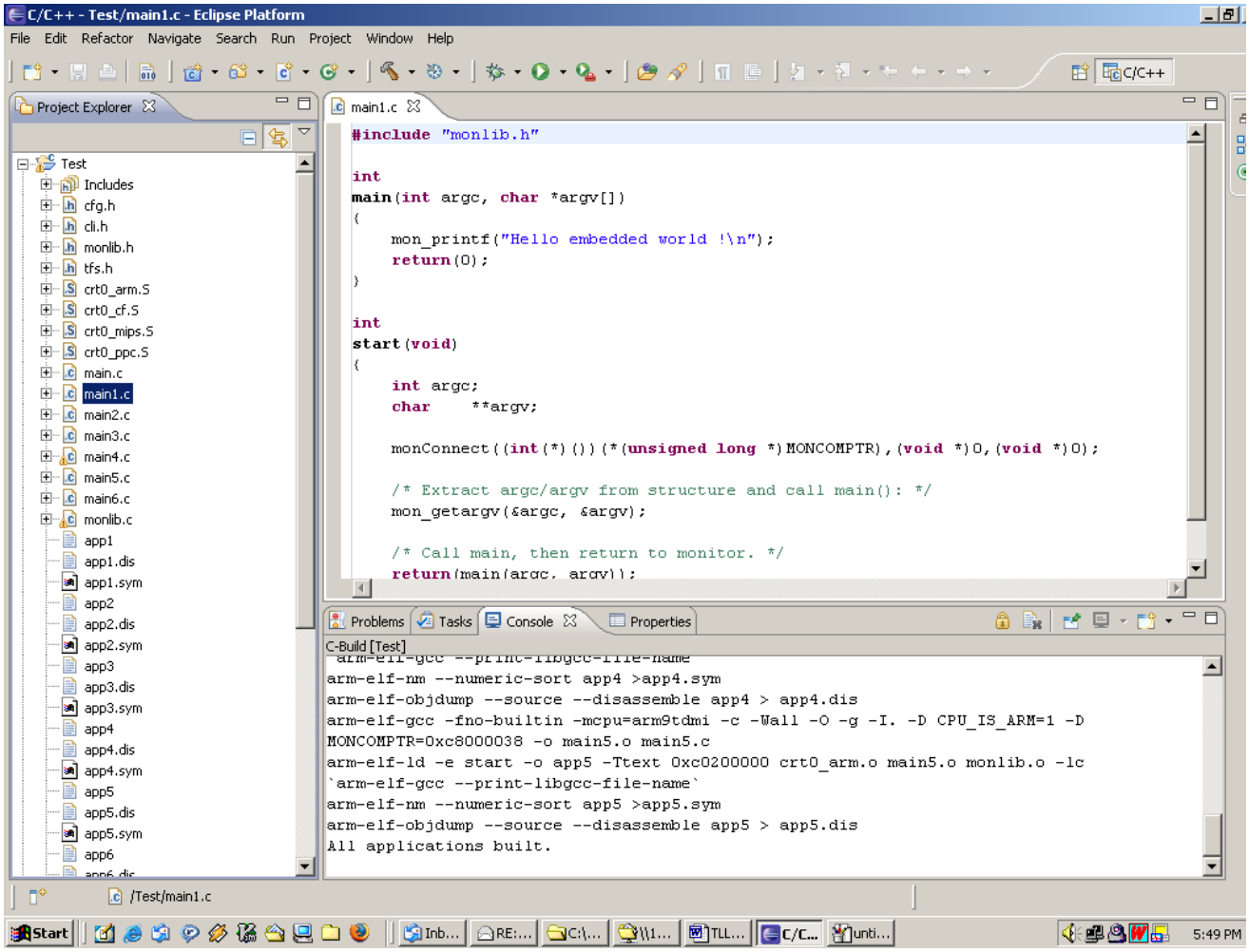

Figure 3-12
Eclipse Screen.

- Now this project shall be built / cleaned with the rules as per the Make file. Individual source codes shall be viewed in display window.
- The project shall be build by Right Click on Project name → Build
- With the above this command, the 5 applications (app1, app2, app3, app4 & app5) will be generated for source codes main1.c, main2.c, main3.c, main4.c and main5.c respectively.
- This executables shall be downloaded on to the target board for testing

Note: The Makefile shall be modified accordingly for adding the new source files

Downloading Procedure:

- Prerequisites – The TLL6219 is booted as per setting up operation explained in chapter 2
- Enter the command **xmodem –dy** in the HyperTerminal on desktop. The screen looks as below

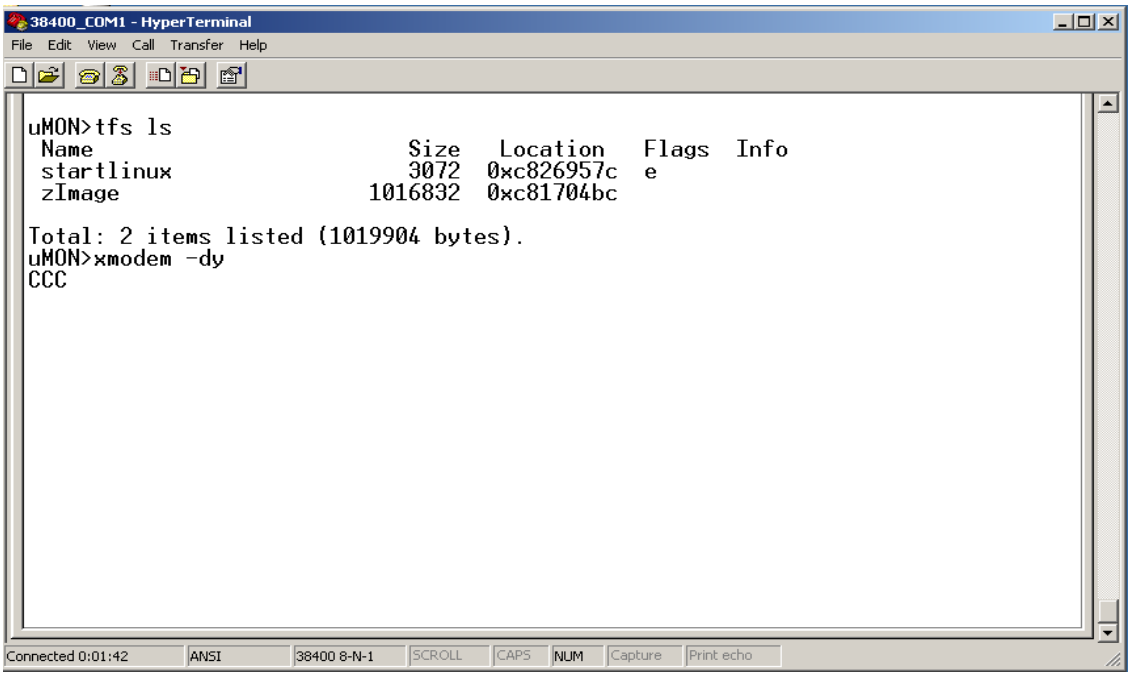

Figure 3-13
Booting in uMON.

- With xmodem command the TLL6219 is ready to receive the file. Now click on Transfer → sendfile in the HyperTerminal menu bar. The following window appears on your desktop

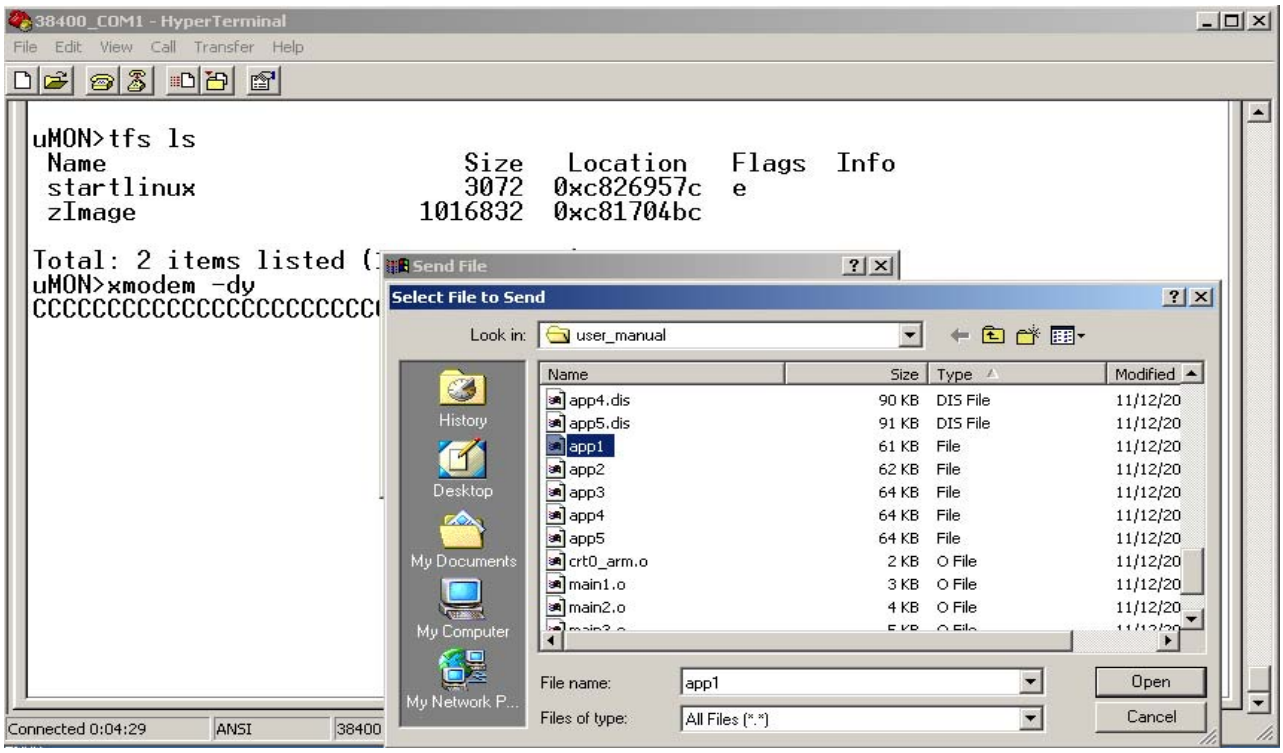

Figure 3-14
Choosing a file.

- Now browse to the app1 executable in the give directory and click open "D:\TLL6219\ARM_uMon-Examples\workspace\test"

Note: The file transfer is supported only in ymodem protocol option in send file operation

- After choosing right executable file (C:\TLL6219), the following window appears on your desktop

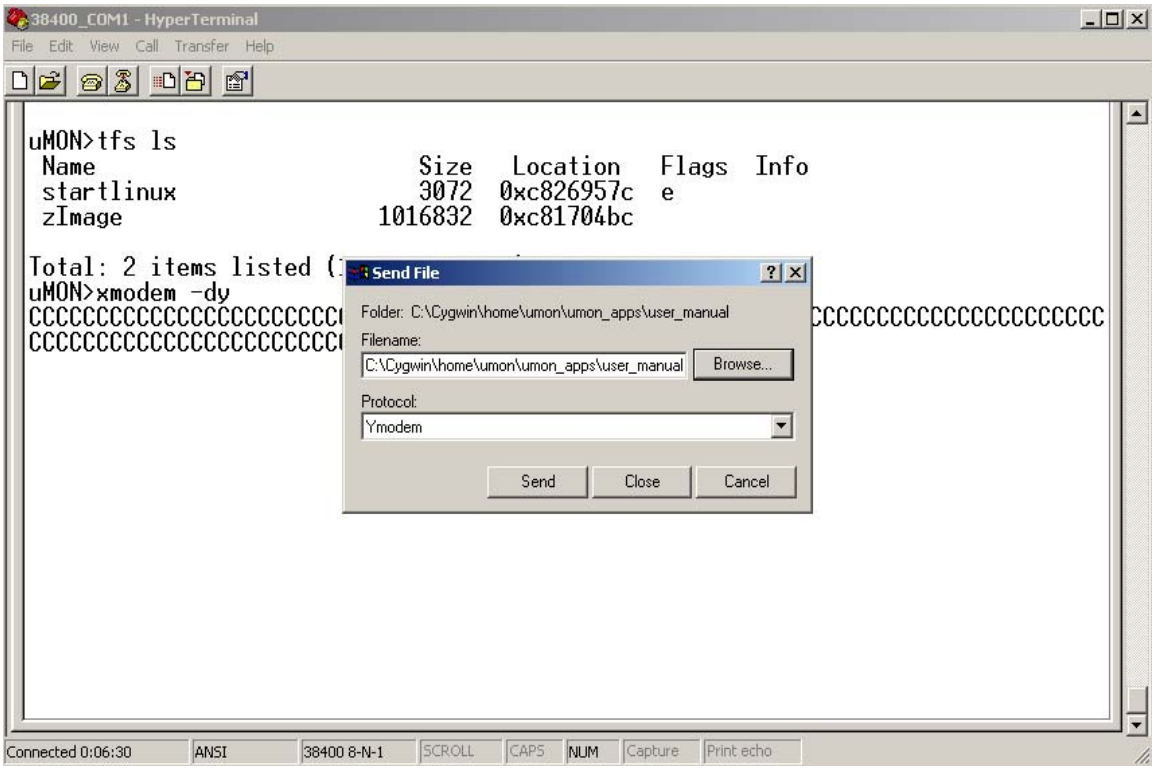

Figure 3-15
Sending a file.

- Click on Send button in the above window, the file transfer starts and after the completion the following window appears on desktop

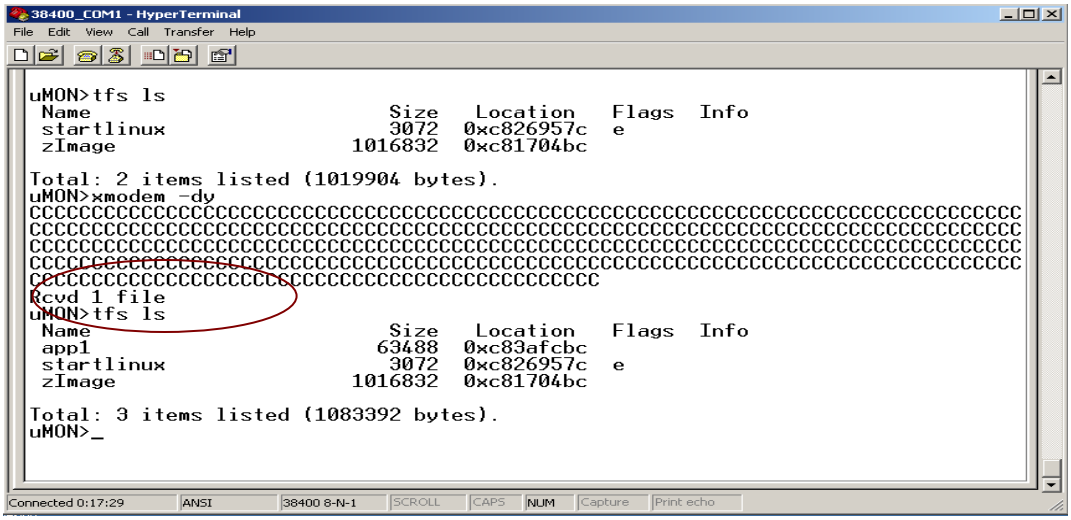

Figure 3-16
Reception of a file.

- The “Rcvd 1 file” shows the completion of receive process. This received file is stored in the Flash and shell be checked with “tfs ls” command. In the above screen we can see app1 listed.
- In the above screen, the app1 is not having FLAG. The “E” is an executable flag to execute as command. It is required to change the file as Executable with the command
- “tfs –fE cp app1 app1”
- After the above command the following screen appears on desktop

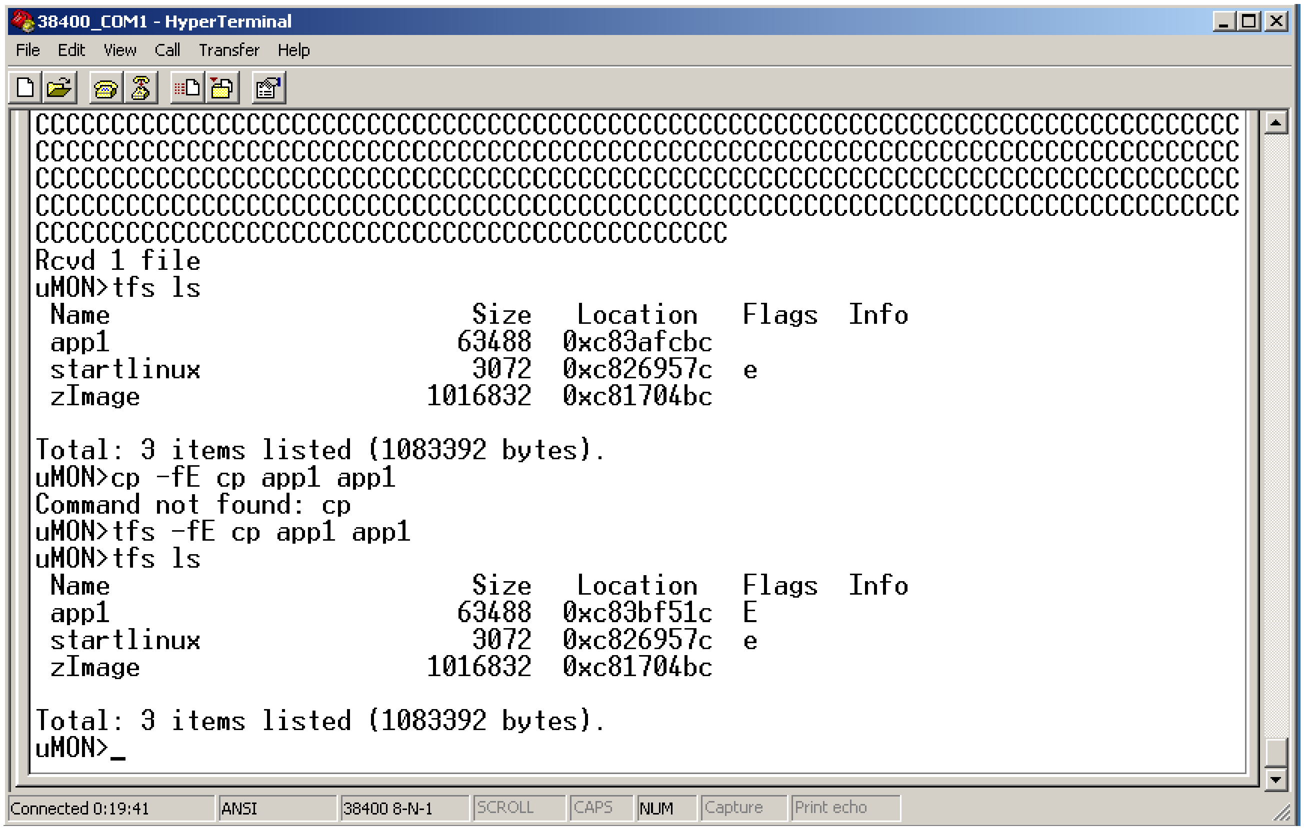

Figure 3-17
Flag setting of a file.

- Now we can see the “E” flag for app1 file

Executing the application on TLL6219 ARM mezzanine module:

- Type the application name in the command line (as app1) and then press enter. The following window appears on your screen

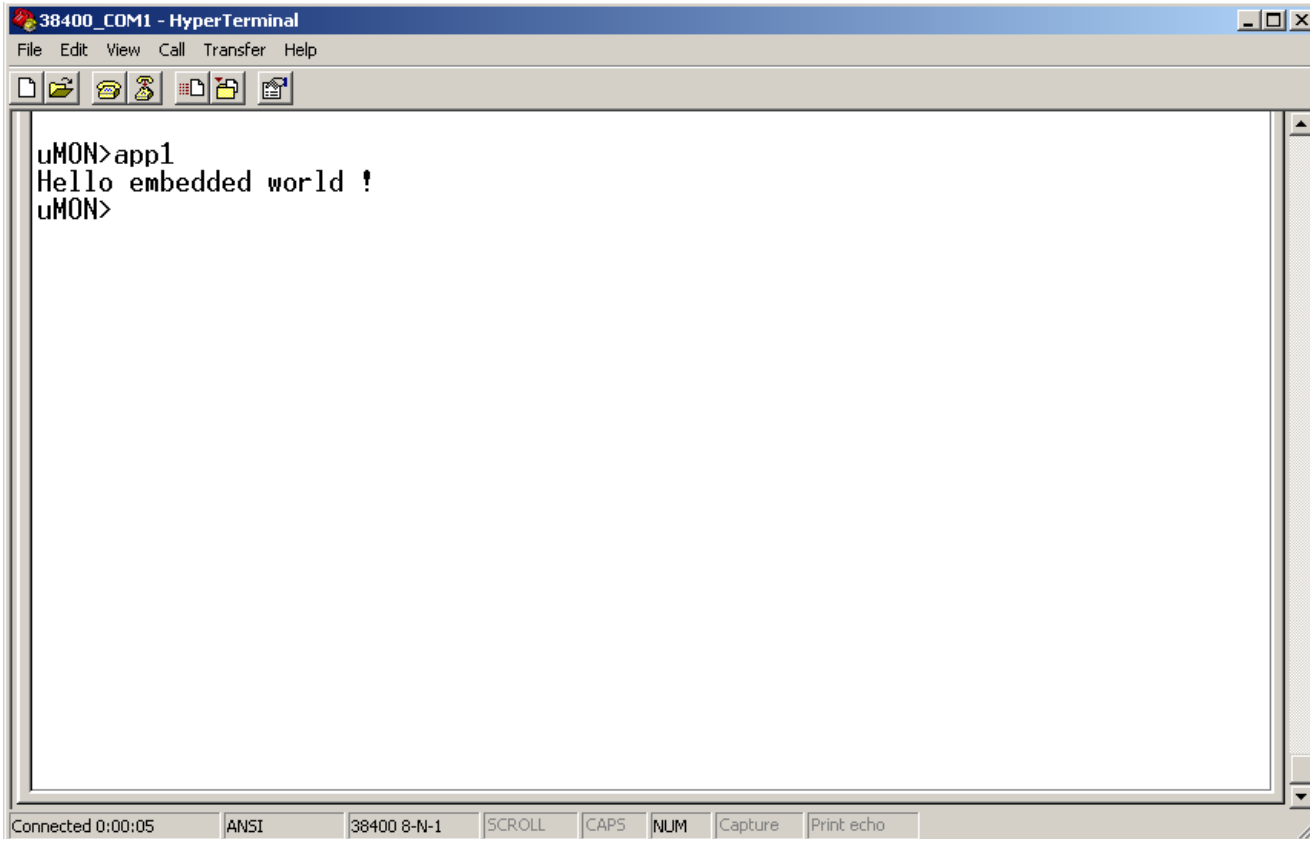

Figure 3-18
Choosing a Working Directory.

- The pre-stored data from the executable is sent to HyperTerminal and printed on console. For this operation the monlib.h header file in Micromonitor provides functions for communication and printing