phyCORE-MPC555

QuickStart Instructions

**Using Freescale CodeWarrior
Development Tool Chain**

Note: The PHYTEC Spectrum CD includes the electronic version of
the phyCORE-MPC555 English Hardware Manual

Edition: August 2009

	EUROPE	NORTH AMERICA
Address:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Ordering Information:	+49 (800) 0749832 order@phytec.de	1 (800) 278-9913 sales@phytec.com
Technical Support:	+49 (6131) 9221-31 support@phytec.de	1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Site:	http://www.phytec.de	http://www.phytec.com

5th Edition:August 2009

Figures

1 Introduction to the phyCORE-MPC555 Rapid Development Kit

This QuickStart provides:

- general information on the PHYTEC phyCORE-MPC555 Single Board Computer

- an overview of Freescale's CodeWarrior software development tool chain, and

- instructions on how to run example programs on the phyCORE-MPC555, mounted on the PHYTEC Development Board, in conjunction with Freescale's CodeWarrior tools.

Please refer to the phyCORE-MPC555 Hardware Manual for specific information on such board-level features as jumper configuration, memory mapping and pin layout. Selecting the links on the electronic version of this document links to the applicable section of the phyCORE-MPC555 Hardware Manual.

1.1 Rapid Development Kit Documentation

This "Rapid Development Kit" includes the following electronic documentation on the enclosed "PHYTEC Spectrum CD-ROM":

- PHYTEC phyCORE-MPC555 Hardware Manual and Development Board Hardware Manual

- PHYTEC phyCORE-MPC555 QuickStart Instructions

- MPC555 controller User's Manuals and Data Sheets

1.2 Overview of this QuickStart Instruction

This QuickStart Instruction provides a general "Rapid Development Kit" description, software installation hints, and example programs enabling quick out-of-the box start-up of the phyCORE-MPC555 in conjunction with Freescale's CodeWarrior tools. The *"Getting Started"* section uses the example program *"Hello_Fl"* to demonstrate the download of user code to the phyCORE-MPC555 external RAM, internal Flash, and external Flash, using Freescale CodeWarrior Debugger and Flash Programmer. This section also provides hints for creating a new phyCORE-MPC555 project in Freescale CodeWarrior.

1.3 System Requirements

Use of this "Rapid Development Kit" requires:

- the PHYTEC phyCORE-MPC555,

- the PHYTEC Development Board with the included DB-9 serial cable, DB-25 printer port cable and AC-to-DC adapter supplying 9 V DC/min. 1.5 A,

- the PHYTEC Spectrum CD,

- the Freescale CodeWarrior CD (the required evaluation license key needs to be requested from Freescale using the integrated registration wizard),

- an IBM-compatible host-PC (Pentium II or higher) running Microsoft Windows 2000/XP or Windows NT (Service Pack 6a or later), and 300 MB hard disk space plus project space.

For more information and example updates, please refer to the following sources:

http://www.phytec.com - or - http://www.phytec.de
support@phytec.com - or - support@phytec.de

http://www.freescale.com/

1.4 The PHYTEC phyCORE-MPC555

The phyCORE-MPC555 represents an affordable yet highly functional Single Board Computer (SBC) solution in the size 72 x 57 mm / 2.83 x 2.24 in. The standard board is populated with a Freescale MPC555 PowerPC Microcontroller featuring a 32-bit processor architecture with a 64-bit Floating-Point-Unit, 40 MHz processor speed, dual CAN 2.0B, dual UART, one SPI, two independent TPU's, ten channel Capture/Compare-Units with two 16-bit Timers, eight channel dedicated PWM output, 32 analog inputs and several I/O signals.

All applicable data/address lines and applicable signals extend from the underlying logic devices to two high-density 160-pin Molex SMT pin header connectors (pin width is 0.635 mm./25mil) lining the circuit board edges. This enables the phyCORE-MPC555 to be plugged like a "big chip" into target hardware.

The standard memory configurations of the phyCORE-MPC555 features 256 kByte external high-speed synchronous BURST-SRAM (2-1-1-1 cycles), 26 kByte internal SRAM and the 448 kByte internal Flash EEPROM (optionally external 512 kByte, 1/ 2/ 4 MB or 8 MB Flash memory are provided). The internal and external Flash supports direct on-board programming without additional programming voltages.

phyCORE-MPC555 Technical Highlights

- phyCORE dimensions 72 x 57 mm with two high-density 160-pin Molex SMT pin header connectors
- Processor: Freescale Embedded PowerPC MPC555, 40 MHz clock

- **Internal components of the MPC555:**
- 32-bit PowerPC core
- 64-bit floating point unit
- 26 kByte SRAM, buffered by battery voltage input
- 448 kByte Flash
- dual UART/SPI
- dual CAN 2.0B
- dual TPU with 16 I/O channels for each unit
- two 16-bit timer with 10 capture/compare units
- eight channel PWM system
- dual 16 channel 10-bit A/D converter with sample/hold, 10 µs conversion time, two trigger inputs
- multi-purpose I/O signals

- Memory:

-	SRAM:	256 kByte (optionally 1/2/4/8 MB) Flow-Through synchronous BurstRAM, 32-bit access, 0 Wait, 2-1-1-1 Burst Mode
-	Flash:	0/ 512 kByte / 1/ 2/ 4/ 8 MB, 32-bit wide
-	I^2C memory:	4 kByte EEPROM (up to 32 kByte), optional I^2C FRAM (512 Byte to 8 kByte) or I^2C SRAM (256 Byte) buffered by battery voltage input

- I^2C Real-Time Clock with calendar register and programmable alarm function
- power-down/wake-up support
- dual UART/SPI port: RS-232 transceiver for both channels (RxD/TxD); also configurable for TTL

- dual CAN port, CAN transceiver 82C251 for both channels; also TTL configurable
- JTAG/BDM test/debug port
- commercial (0...+70°C) and extended temperature range (-40...+85°C/+125°C) are available

The PHYTEC Development Board, in EURO-card dimensions (160 x 100 mm/6.25 x 4 in.), is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion of PHYTEC phyCORE-MPC555 Single Board Computer.

Development Board Technical Highlights

- reset push button, configurable via jumper for different reset signals
- IRQ push button, configurable via jumper for different interrupts
- wake-up push button
- boot jumper to select internal or external Flash memory
- jumper to enable the programming voltage for internal Flash memory
- two software programmable LED's (red/green)
- LEDs for supply voltage monitoring
- LED to monitor debug or run mode
- power supply for unregulated input voltage from 7 V to 13 V. It supplies regulated +5 V and +3.3 V for the phyCORE-MPC555. Additional +5 V is created for the VG96-connector.
- VG96 connector with all I/O signals of the MPC555
- two standard width pin header rows (3 x 54) which provide all phyCORE-MPC555 signals
- two DB-9 sockets for the RS-232 interface
- two DB-9 plugs for two separate CAN interfaces with configurable terminating resistors
- DB-25 plug for BDM interface to support direct connection to PC printer-port
- 2 x 5 standard width pin header rows to support third party BDM interfaces

1.5 The Freescale CodeWarrior Development Tool Chain

Freescale CodeWarrior is a cost-effective and highly functional Integrated Development Environment (IDE) containing a complete suite of development tools to create, compile, link, debug, and assemble fast, optimized embedded systems code for the PowerPC 5xx/6xx/8xx and other architectures.

The Freescale CodeWarrior Integrated Development Environment (IDE) is intuitive, easy to use and designed to provide a variety of tools that work in a cohesive manner regardless of the host, language, or target architecture. This means you can reach more markets and solve more problems without having to learn new tools. Freescale CodeWarrior IDE's graphical interface includes an editor, project manager, class browser, compiler, source-level and assembly-level debugger, stand-alone assembler, and linker.

In addition to keeping all your development tools a keystroke away, the IDE stores compiler settings and tracks all the dependencies for your project, simplifying even the most complex development build. The editing and software debugging tools in CodeWarrior are state-of-the art and feature syntax highlighting, the power to evaluate structures and complex expressions in the debugger, and the ability to correct source code in the error message window.

Freescale CodeWarrior consists of the IDE and Debugger core tools, which are structured as follows:

IDE:

Windows-based Integrated Development Environment housing the following tools:

- **Project Manager:** shows the collection of files used to build an output file
- **Editor:** double-clicking on a file in the Project Manager window invokes the Editor
- **Build System:** IDE plug-in "tools" to create and then convert source code to output code
- **C compiler:** IDE plug-in
- **Linker:** IDE plug-in
- **Make:** IDE plug-in
- **Debugger:** Fully active source-level and assembly-level Windows-based debugger

Upon installation of CodeWarrior, the IDE and Debugger executables are located in the ***C:\Program Files\Freescale\CodeWarrior EPPC 5xx V8.7\Bin*** folder. All tool commands are easily accessible via intuitive pull-down menus with prompted selections.

As depicted below, CodeWarrior's editor, C/C++ compiler, linker and debugger comprise the basic tool chain. The remaining plug-in tool components assist in code navigation and build control.
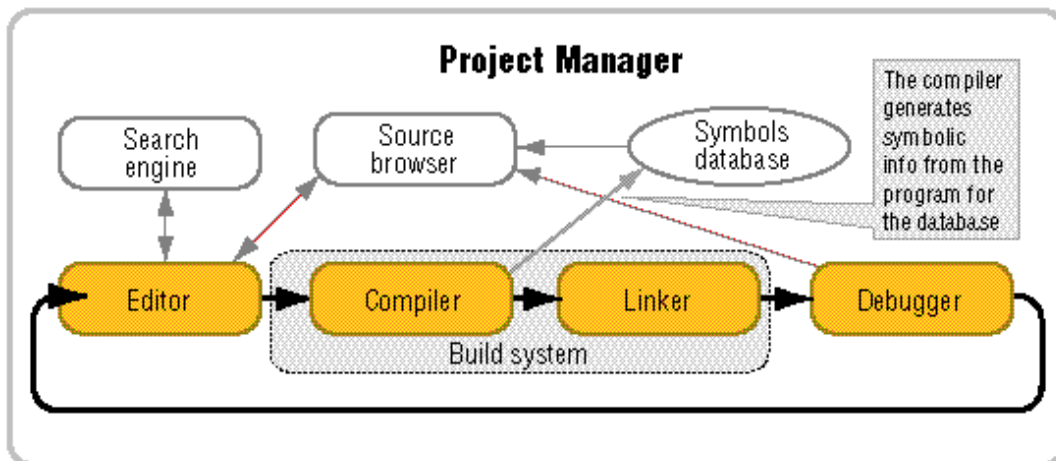

Figure 1:   Freescale CodeWarrior IDE Architecture

All IDE commands and functions are accessible via pull-down menus or short cuts. An extensive Help utility is included.

The **Project Manager** gathers all necessary source files, library and project information - such as various output files (called targets) with different memory maps for RAM-download or ROM-imaging - and stores it in a project file assigned the suffix *.mcp*. From the project manager window, all other tool components are easily accessible via mouse control.

The **Editor** is a powerful tool to compose source code. It includes a context-sensitive coloring of keywords for easy navigation and recognition. All colors and custom keyword sets can be user-defined. Pop-up menus provide an overview of all available functions and a navigation to the start of a desired function or its corresponding header file. A graphic difference engine displays the differences between two source code files. The search engine enables location of a specified text string and implements find-and-replace operations.

The **Build System** contains the C/C++ compiler, assembler, linker and code converter. The CodeWarrior compiler produces fast, optimized embedded systems code. The C compiler is NIST Certified ANSI C, and the C++ compiler closely tracks the emerging ANSI/ISO C++ standard. To increase the compilation speed the compiler can generate precompiled headers. For syntax check, the compiler provides pre-processing of single source files or for the complete project. For code exploration, a disassembly command displays the corresponding assembler code that the C code compiles to.

The compiler and linker support *.elf* and Freescale S-record output file formats. A map file is also generated, providing detailed memory map and module information.

The **CodeWarrior Debugger** provides C/C++ source-level, assembly-level and mixed mode debugging. For code exploration, absolute or conditional breakpoints can be accessed or single-step operation can be performed. Step-over, step-into and step-out of code function capabilities are provided. The contents of variables, arrays and structures can be monitored and manually modified.

The Freescale CodeWarrior Debugger supports the following debug options of the phyCORE-MPC555 target.

- A hardware-level debugging option is provided via a PC-to-target connection using the MPC555 BDM-port (Background Debug Mode), a hardware debugger module integrated on the processor. The parallel printer port of the host-PC communicates to the phyCORE via a BDM interface, such as the Macraigor Wiggler or the BDM signal conversion logic integrated on the PHYTEC Development Board. Furthermore for extensive software development, more sophisticated BDM interfaces are available for increase download speed.

- A connection via RS-232 to a software debug Monitor kernel (MetroTRK, SDS Monitor) resident in the external or internal Flash on the phyCORE. The phyCORE-MPC555 is delivered without the MetroTRK monitor kernel resident in the Flash. It can be programmed via BDM connection to the desired Flash memory.

2 Getting Started

What you will learn with this Getting Started example:

- installing the Rapid Development Kit software
- installing Freescale CodeWarrior for PowerPC Embedded Systems
- interfacing the phyCORE-MPC555, mounted on the Development Board, to a host-PC
- downloading example user code from a host-PC to the phyCORE-MPC555 on-board memory

2.1 Installing the Freescale CodeWarrior Tool Chain

When you insert the Freescale CodeWorrior CD into the CD-ROM drive of your host-PC, the Freescale CodeWarrior CD should automatically launch a setup program that installs the required software. Otherwise the setup program *launch.exe* can be manually executed from the root folder of the Freescale CodeWarrior CD.

The applicable CodeWarrior tool chain must be installed to ensure successful completion of this QuickStart Instruction. Failure to install the proper software could lead to possible version conflicts, resulting in functional problems.

A 15-day evaluation license key is included with the installation. After that, the full license key needs to be requested from Freescale using the integrated registration wizard. This is required to use Freescale CodeWarrior Tool Chain.

For installation on WinNT, Win2000 or WinXP hosts you must have administrator privileges.

- Install the Freescale CodeWarrior for Embedded PowerPC Targets tool chain from the enclosed Freescale CD, following the steps indicated in the install procedure.

The default destination location is
C:\Program Files\Freescale\CodeWarrior EPPC 5xx V8.7. All path and file names within this QuickStart Instruction are based on the assumption that you accept the default install paths and drives. We recommend that you accept the default destination location.

Once installation of CodeWarrior is complete, you need to manually copy additional support files from the **PHYTEC Spectrum CD** installation to the Program Files for CodeWarrior on your host-PC. These files enable initialization of and communication to the PHYTEC phyCORE-MPC555 target hardware from within the CodeWarrior IDE.

- After installing the PHYTEC Spectrum CD, replace the files **mpc555.elf** and **mpc555sh.elf** located in:

  C:\ProgramFiles\Freescale\CodeWarrior EPPC 5xx V8.7\bin\ Plugins\Support\Flash_Programmer\EPPC

  with those out of the path:

  C:\phyBASIC\phyCORE- MPC555\Tools\CW8.1\bin\Plugins \Support\Flash_Programmer\Eppc

Note:
In order to allow proper communication between the host-PC's parallel printer port and the BDM interface on the phyCORE Development Board for debugging and code download configure the printer port to be one of the following: SPP, Normal, Standard, Output Only, Unidirectional, AT. Try to avoid ECP, EPP or PS/2 bi-directional. The printer port mode is selected in your System BIOS.

2.2 Interfacing the phyCORE-MPC555 to a Host-PC

Connecting the phyCORE-MPC555 (part # PCM-001-xxxx, mounted on the PHYTEC Development Board (part # PCM-995), to your computer is simple:

- If the phyCORE module is not already connected, mount it onto the Development Board's Molex connector (X1) as shown in the figure below. Ensure that pin 1 of module matches pin 1 of the receptacle on the Development Board.

Ensure that there is a solid connection between the module's pins and the Development Board receptacle. If the phyCORE module is removed from the Development Board, take precautions to properly mount the module when it is reattached to the Development Board. Pin 1 on the phyCORE module (denoted by the hash stencil mark on the PCB) should be matched to the footprint receptacle on the Development Board marked by "X1" on the PCB. Also take precautions not to damage the connectors when the phyCORE is removed from and inserted onto the Development Board.
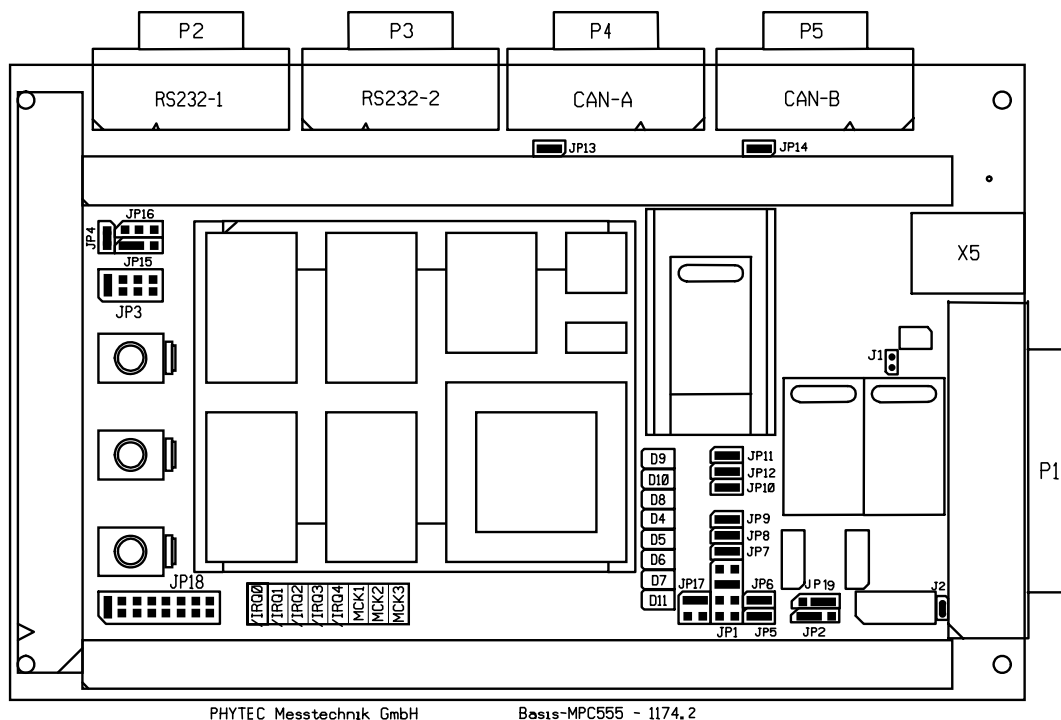
Figure 2:   Development Board Overview and Suitable Jumper Settings

- Configure the jumpers on the phyCORE Development Board as indicated above. This correctly routes the interface signals to applicable connectors and establishes signal connection between on-board peripherals and the phyCORE-MPC555 module.
- Connect the DB-25 connector (P1) of the Development Board to the LPT1 port of your host-PC using the included DB-25 cable. This connection is used for the communication of the Freescale CodeWarrior IDE with the phyCORE-MPC555.
- Connect the DB-9 connector (P2) on the Development Board to a free serial port of your host-PC using the included serial cable. This will enable you to monitor board-host communication via a terminal emulation program, such as Windows HyperTerminal.

- Using the included 9VDC power adapter, connect the power socket on the board (X5) to a power supply (*refer to Figure* 3 *for the correct polarity*). The phyCORE module/Development Board combination requires a 9VDC@1A power supply.
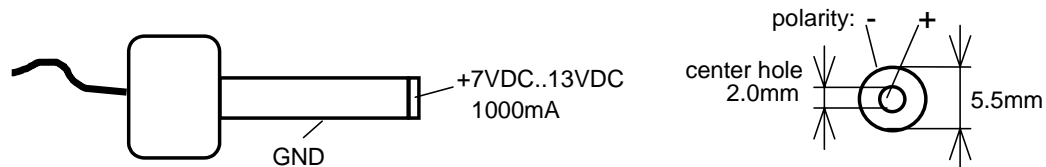
Figure 3:   Power Connector

The LEDs D8, D9 and D10 (three green LEDs) should light, indicating that all voltages are supplied to the board. D11 (red LED) will illuminate, indicating that the MPC555 processor is waiting in debug mode.

The phyCORE-MPC555 should now be properly connected via the Development Board to a host-PC and power supply and you are now ready to use the Freescale CodeWarrior tool chain to establish communication. This phyCORE/Development Board combination shall also be referred to as "target hardware".

2.3 Launching the CodeWarrior IDE

CodeWarrior should have been installed during the initial setup procedure as described in the previous section.

Now please download the Example **ftp://ftp.phytec.de/pub/ Products/phyCORE-MPC555/Metrowerks/CW8.7/Hello_FL.zip** to your hard disc. Navigate to *C:\PHYBasic\phyCORE-MPC555\ Demos*, create the folder CW8.7 and decompress the example into this folder.

- Launch the Freescale *CodeWarrior IDE* by double-clicking on the *CodeWarrior* icon on your desktop or by selecting *CodeWarrior* from within the

*Programs\Freescale CodeWarrior\PowerPC 5xx V8.7\
CodeWarrior IDE* folder.

- Open the ***Hello_Fl.mcp*** project file by selecting *File \ Open* from the pull-down menu and navigating to
PHYBasic\phyCORE-MPC555\Demos\CW8.7\Hello_Fl.
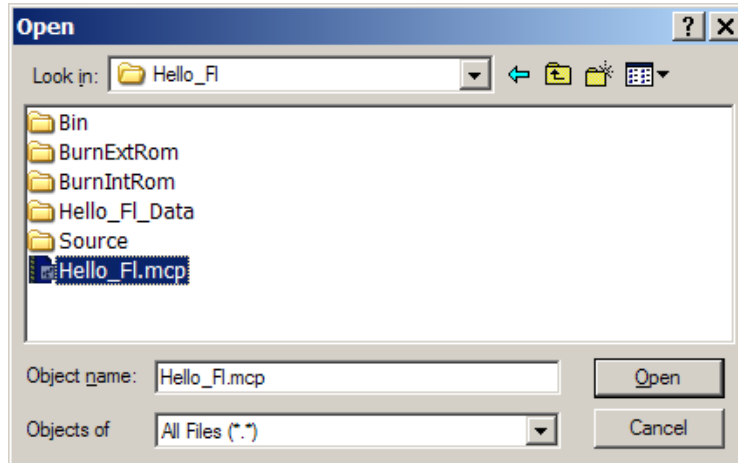
- The ***Hello_Fl.mcp*** project file will open in a project window.
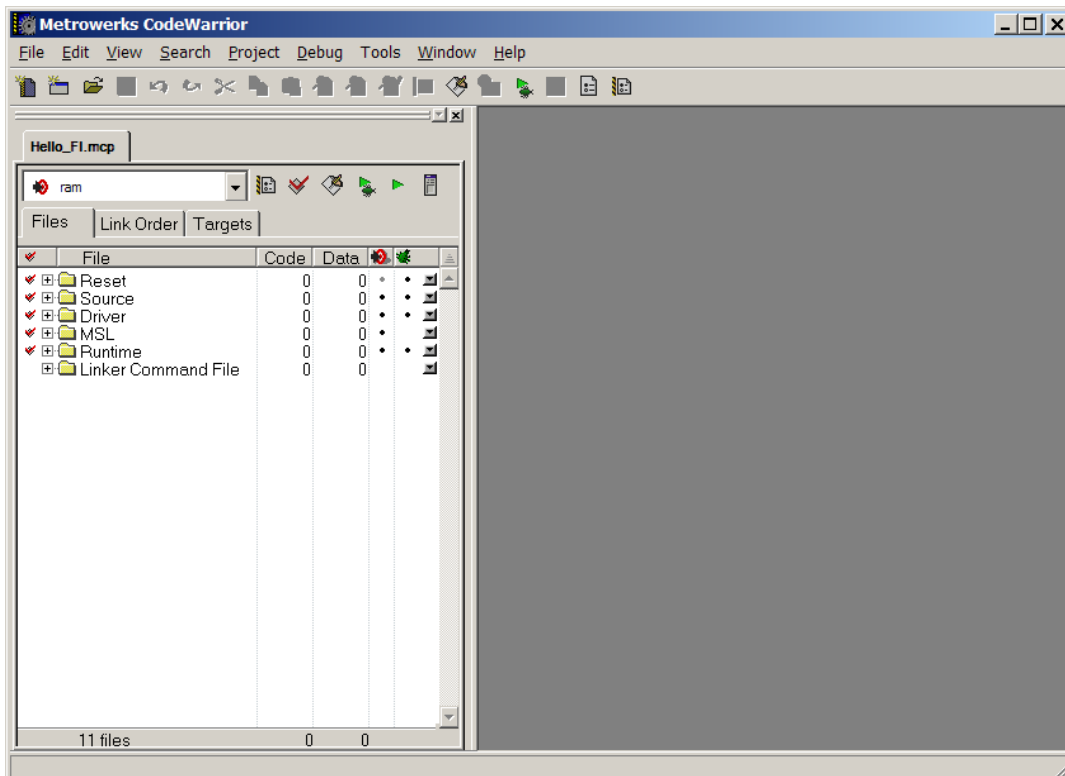
2.4 Downloading Example Code with CodeWarrior IDE

The *Hello_Fl* example downloads code to the target hardware that, when executed, sends a character string over RS-232 from the target to the host-PC. The character string can be viewed with a terminal program. The program also flashes the user LEDs D4 and D5.

The **"Hello_Fl"** project consists of two software targets, *ram* and *rom.*

"ram" target
The ram target is configured for downloading to the external SRAM. All program code, data and stack space is linked to the external SRAM. The external SRAM is available from address 0x10000000 to 0x1003FFFF (256 kByte). The end address depends on the actual SRAM capacity available on the phyCORE-MPC555.

"*rom"* target
The rom target is configured to generate an output that is linked for operation in ROM (either internal or external Flash memory). This target consists of all components, such as exception table or initialization procedures, which are necessary for bootable code. If this target is programmed into one of the Flash memories, it can be used to debug itself. This is suitable because of the special ROM debug capability of the MPC555 on-chip debug agent. If you start the debugger with reference to this target, only the database of this target is used to correlate and show the actual position and status of program execution and data contents.

Flash memory is available from address 0x00000000. The reset exception resides at address 0x00000100, from where the software execution starts. Data and stack space is located in the external SRAM starting at address 0x10000000.

2.4.1 Configuring HyperTerminal Program for Hello_Fl example

Monitoring the execution of the ***Hello_Fl*** demo requires use of a terminal program, such as the HyperTerminal program included within Windows.

- Start the HyperTerminal program within the *Programs \ Accessories \ Communicatio n* bar.
- The HyperTerminal main window will now appear.
- Double-click on the HyperTerminal icon *Hypertrm* to create a new HyperTerminal session.
- Enter a name such as "*Connect to Com1*" (be sure to use an available COM port for your system) and select an icon of your choice to establish a new connection using HyperTerminal.

- Click *OK* to advance to the connection window.

- Click *OK* again to advance to the next window (COM1 Properties). Then set the following COM parameters: Bits per second = *57600*; Data bits = *8*; Parity = *None*; Stop Bits = *1*; Flow Control = *None*. Not shown: Start Bits = *1*.
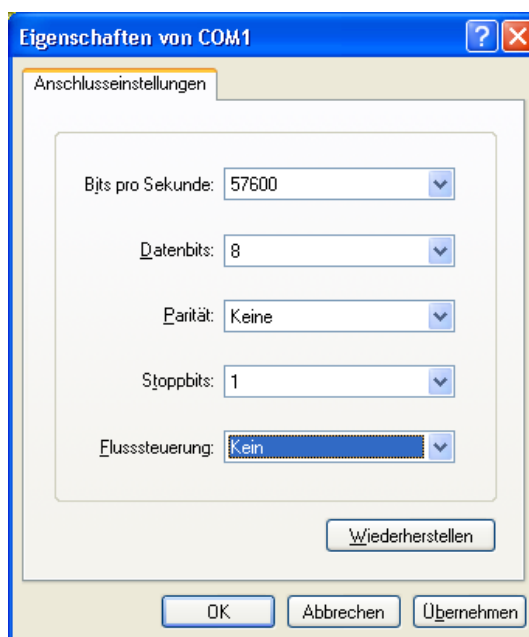
- Selecting *OK* advances you to the COM Direct – HyperTerminal monitoring window. Notice the connection status report in the bottom lower corner of the window.
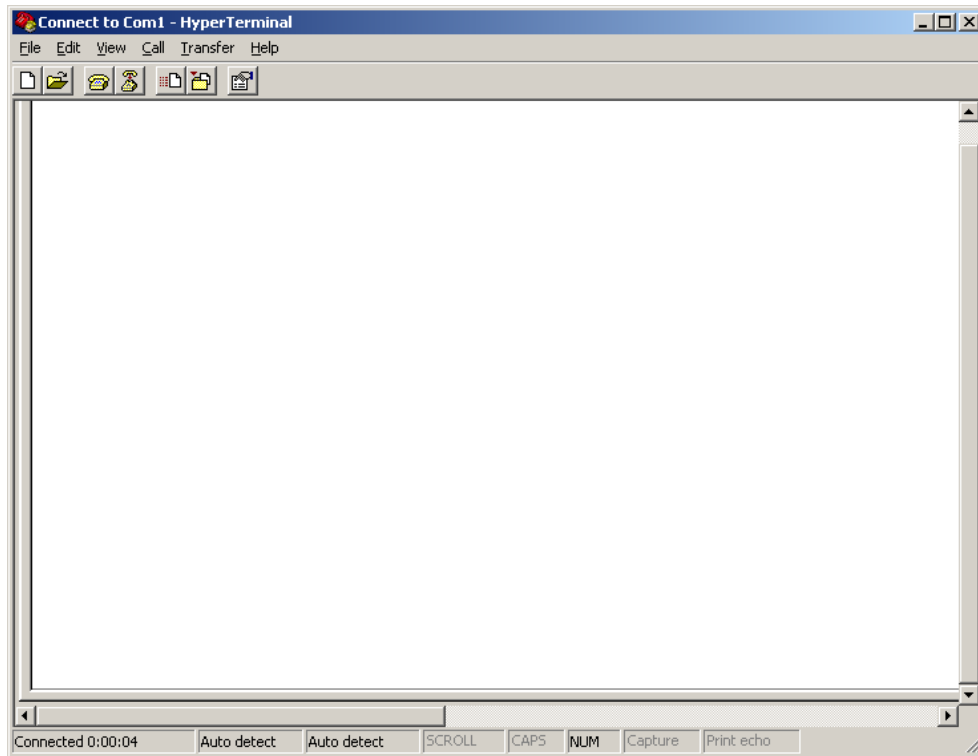
HyperTerminal is now prepared to view the output of the example program once it is executed.

2.4.2 Running the Hello_Fl.mcp Demo Project in RAM

- Return to the Freescale CodeWarrior IDE, leaving the HyperTerminal session open and connected to the target hardware.
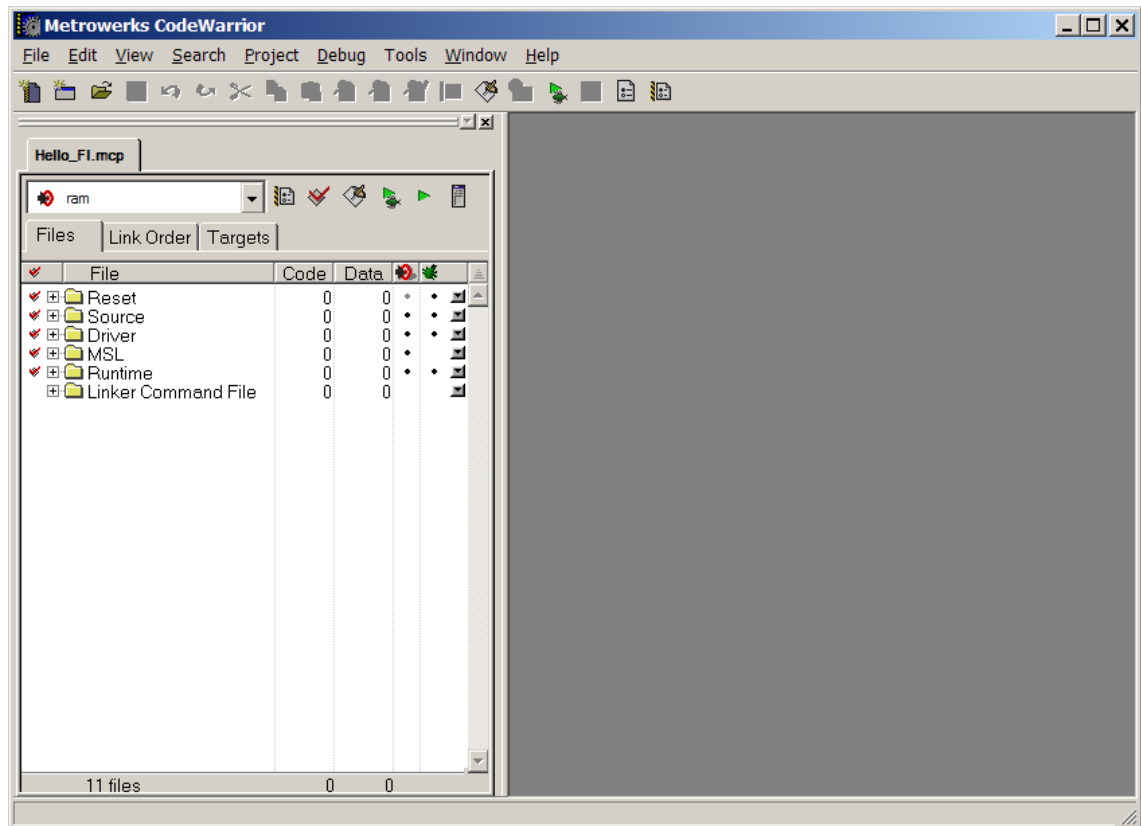
If not already selected, select the software target "**ram**" from the pull-down at the top of the project window.

- Notice the red check marks in front of the individual files. This indicates that the project is not up to date and a new build should be performed. This typically happens when you open an existing project for the first time.
- Perform the build process by clicking the *Make* icon   or select *Project / Make* from the IDE pull-down menu.

- The Make process generates an output binary (***ram.elf***). The Make status window will appear within the CodeWarrior IDE while the project is being compiled and linked. Once complete the project window will be visible.

Notice that the red check marks in front of the individual file groups in the project window have disappeared showing the project is up to date and all necessary output files have been created.

Hint:
If an error message should appear "Cannot open file" go into the EPPC Debugger Settings menu within your target settings and make sure the file "***phyCORE-MPC555_init.cfg***" is entered. Click on *Apply* and *OK* to save the setting. Also make sure that the source files don't have "read-only" attributes.

- Start the download process by clicking on green debug arrow in the upper right section of the project window. You can also perform the download by pressing the <F5> function key or selecting the *Project / Run* item in the main menu bar.
- The following status window will appear during the download.
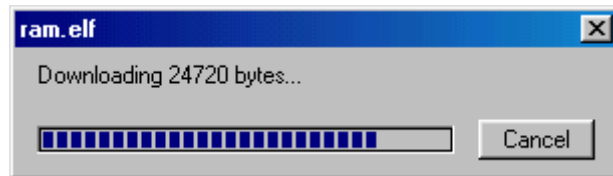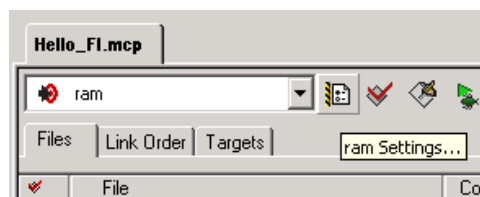
- Once download is complete, you will advance to a more detailed project window that displays the source code in its lower panel.
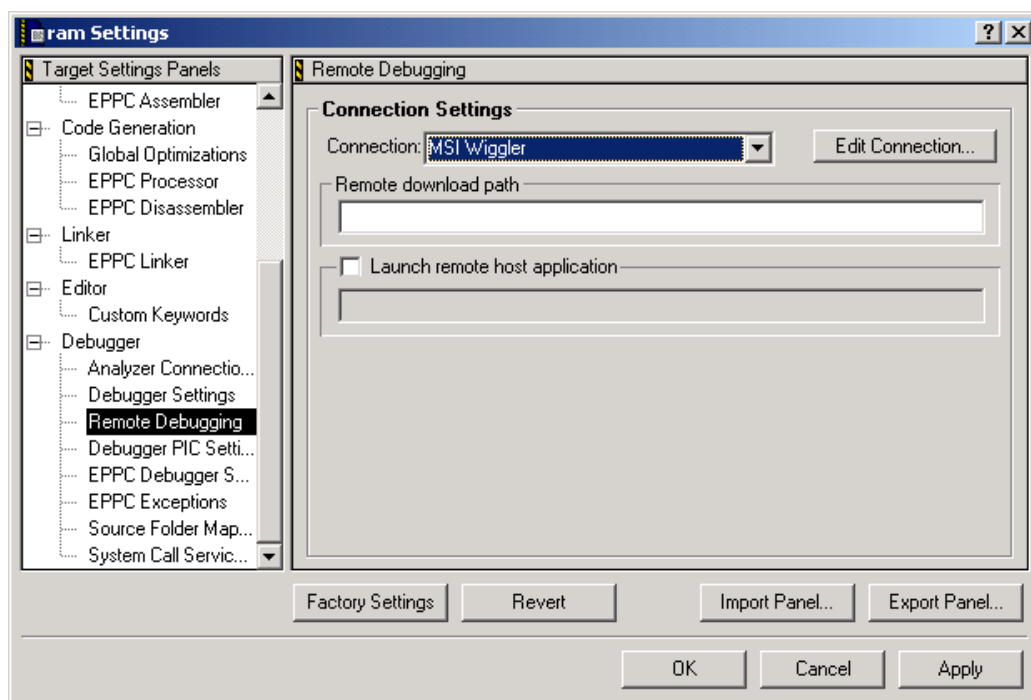
If you get the following error message after starting the program download you need to check your target settings and configure the correct Remote Debugging connection:
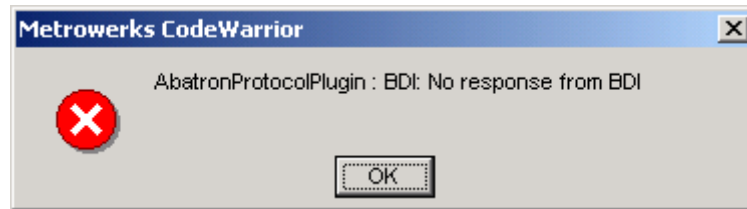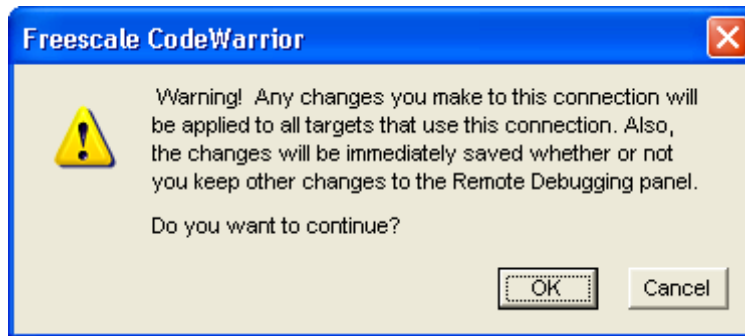

- Select *Edit \ ram Settings* or click on the *ram Settings..* icon in the project window.
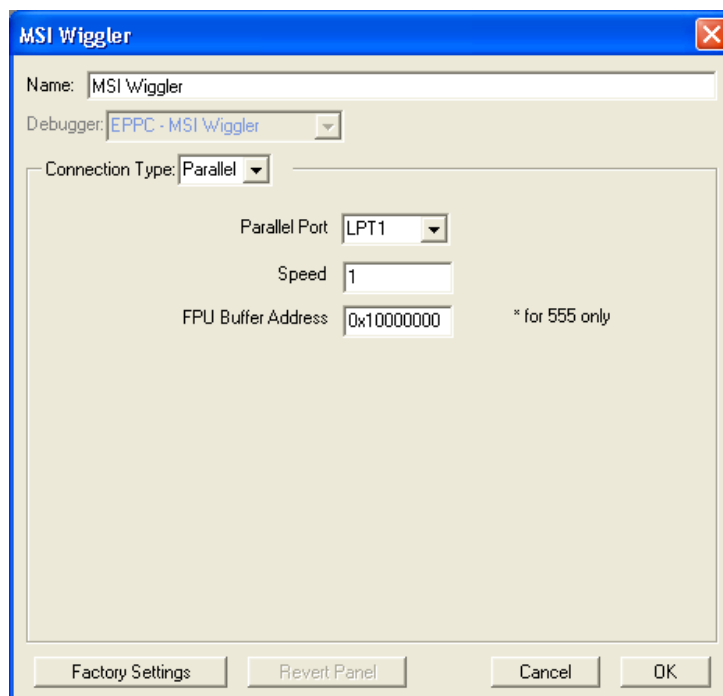

- Open the *Debugger \ Remote Debugging* window and select **MSI Wiggler** under *Connection Settings*.
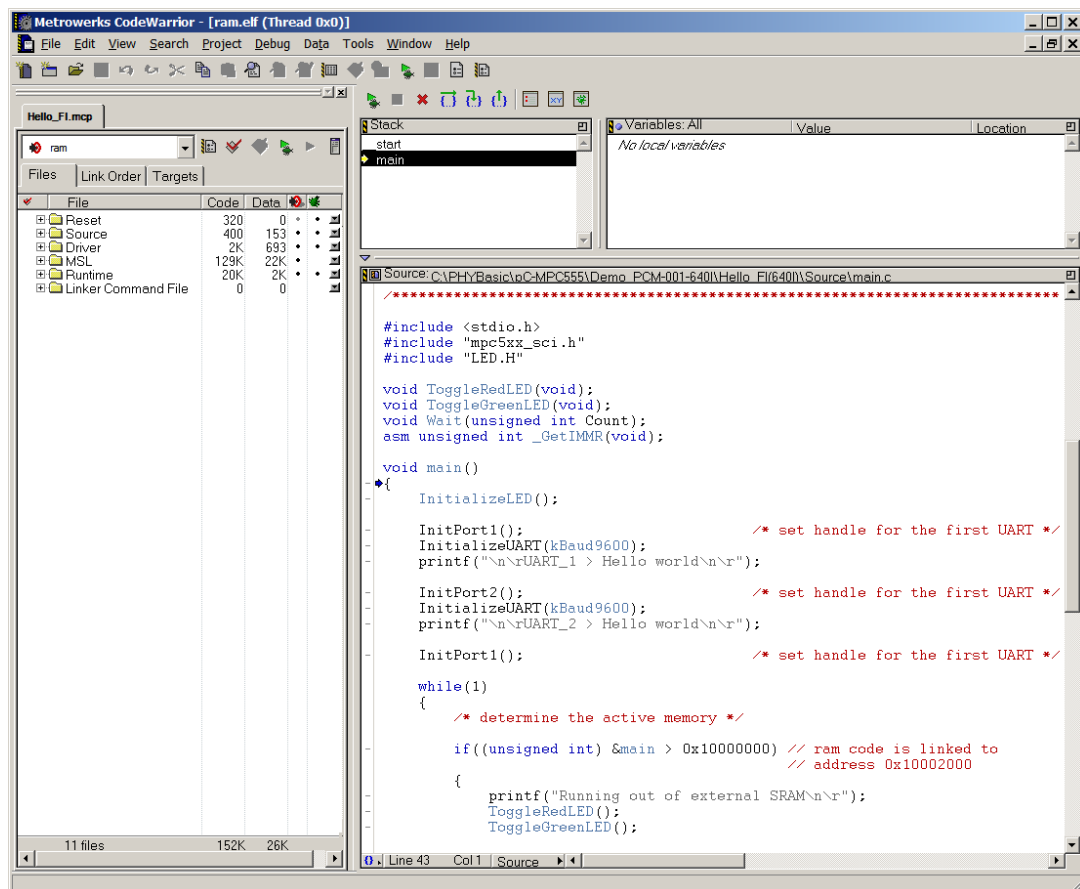

- Click on *Edit Connection…* then *OK*

- Enter *0x10000000* for *FPU Buffer Address* and click *OK*.

- Click on *Apply*, then *OK* and repeat the download.

- Once download is complete, you will advance to a more detailed project window that displays the source code in its lower panel.

- Click on the green debug arrow [icon] at the top of the project window (or press <F5> or select *Project / Run* in the main tool bar). This will run the main code, starting from the location indicated by the blue arrow in the source window [first line after main()]. The lower portion of the ***ram.elf (Thread 0x0)*** window will indicate that the demo program is executing out of RAM.
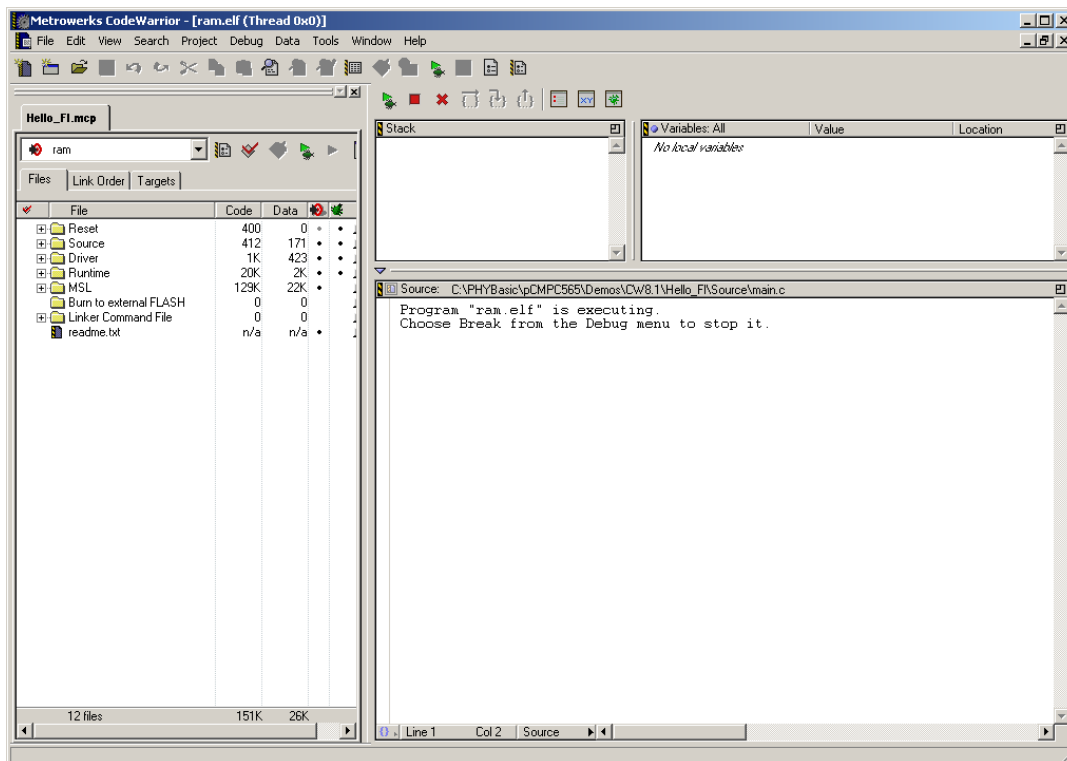
- Switch back to the HyperTerminal program to view the output of the demo program, which will print the character string,

  "UART_1> Hello world
  Running out of external SRAM"

LED D4 (red) and LED D5 (green) will flash on and off simultaneously.

```
Connect to Com1 - HyperTerminal
File  Edit  View  Call  Transfer  Help

UART_1 > Hello world
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
Running out of external SRAM
_

Connected 0:00:13    Auto detect    9600 8-N-1    SCROLL    CAPS    NUM    Capture    Print echo
```

- The *Hello_Fl* project implements both UART_1 and UART_2. To view the output of UART_2, connect the RS-232 cable to the DB-9 connector at P3 on the Development Board. After restarting the code execution UART_2 outputs the following:


```
Connect to Com1 - HyperTerminal
File  Edit  View  Call  Transfer  Help

UART_2 > Hello world

Connected 0:00:58    Auto detect    9600 8-N-1    SCROLL    CAPS    NUM    Capture    Print echo
```

- To end the debug session click on the *Kill thread* ✖ button at the top of the project window (or press <Shift>+<F5> or select *Debug / Kill* in the main tool bar).

2.4.3 Downloading Hello_Fl to the MPC555 on-chip Flash

- Make sure Jumper JP15 on the phyCORE-MPC555 Development Board is closed at position 1+2 (*refer to Figure 2*).
- From the pull-down menu, at the top of the **Hello_Fl.mcp** project window, select the software target "**rom**".
- The project window should look as follows:
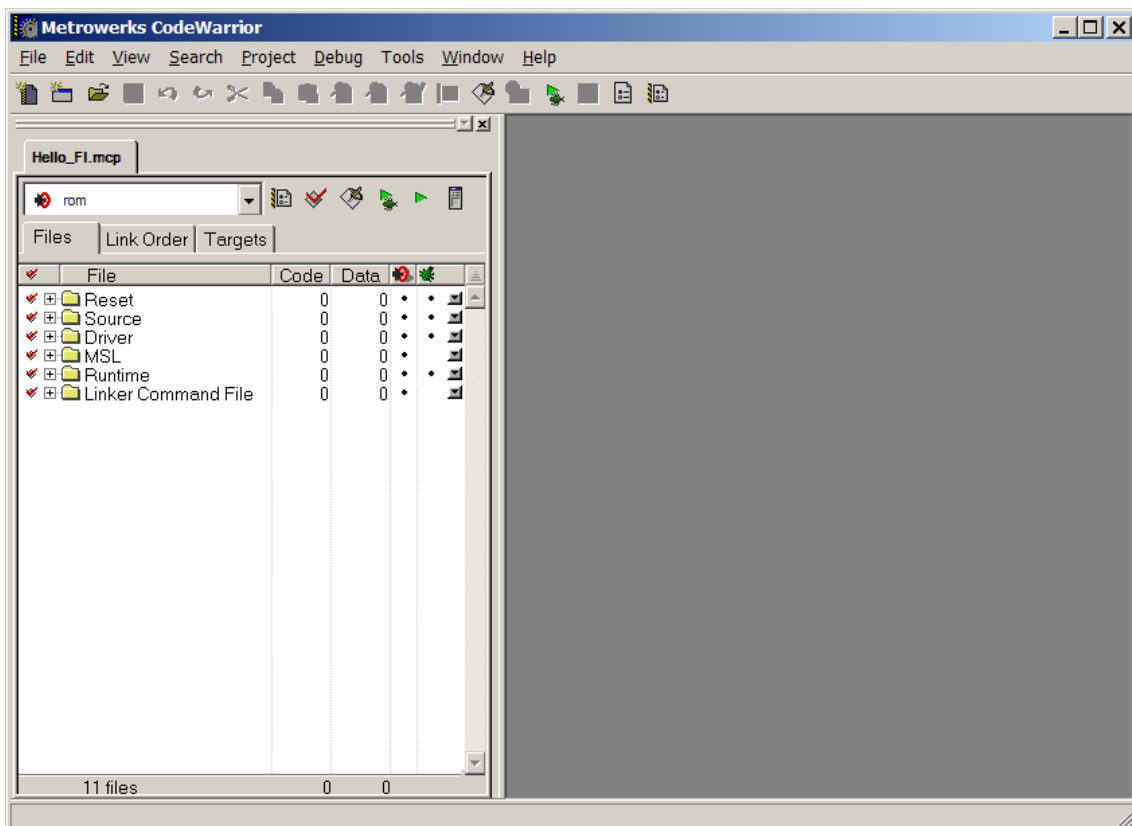
- Perform the build process by clicking the *Make* icon ✧ or select *Project / Make* from the IDE pull-down menu or use the <F7> function key.

- Open the **Flash Programmer** window by selecting *Tools / FlashProgrammer* from the pull-down menu. The following window will appear:
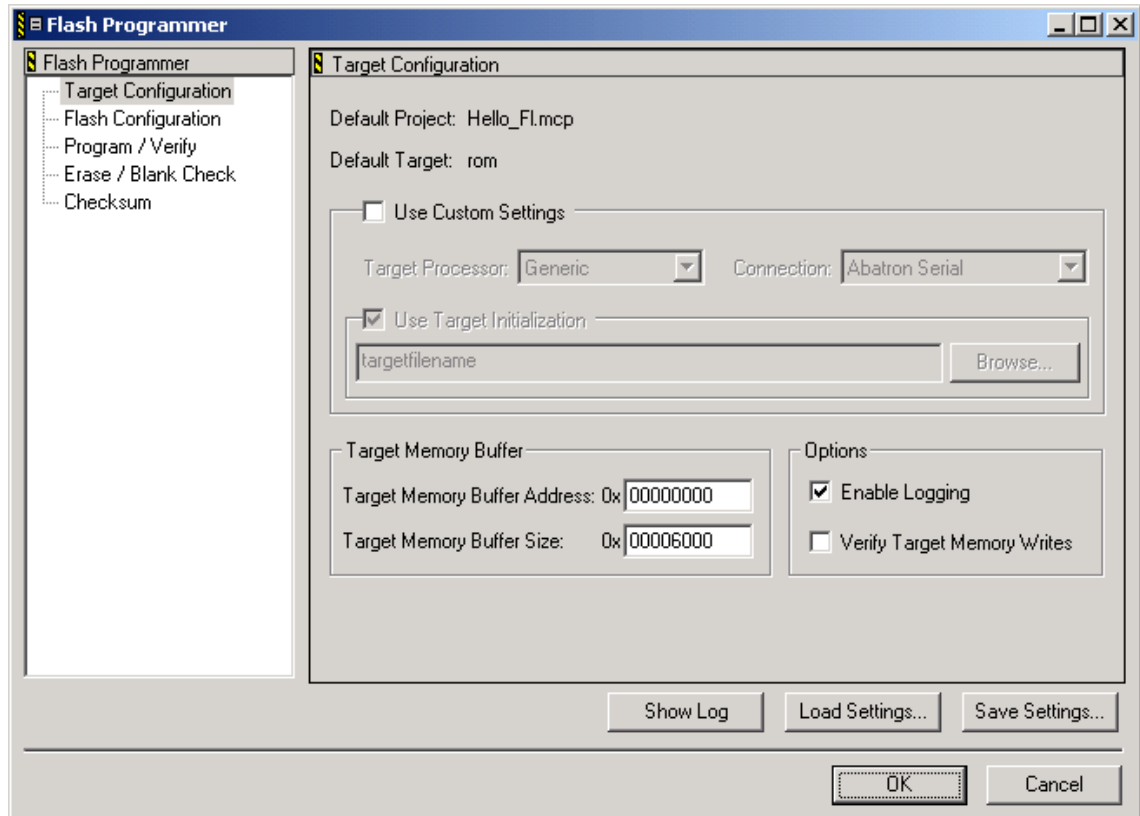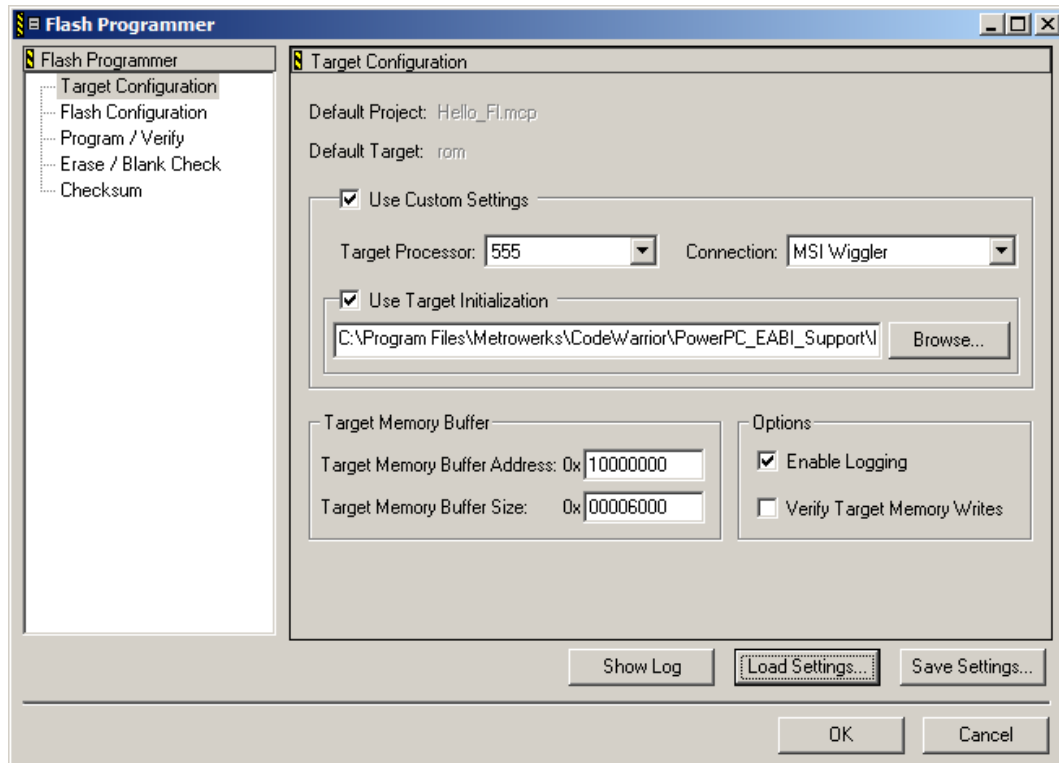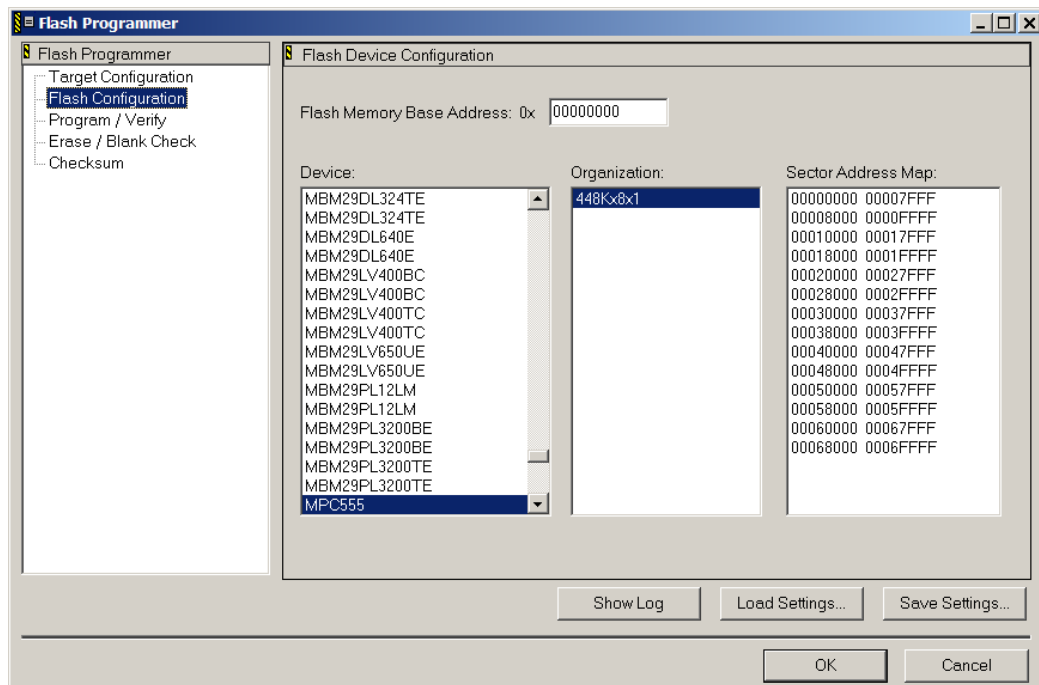

- Click on the *Load Settings...* button.

- Browse to the ***phyCORE-MPC555_internal_flash.xml*** file located in the
C:\Freescale\CodeWarrior EPPC 5xx V8.7\bin\Plugins\Support\Flash_Programmer\EPPC folder and open the file.
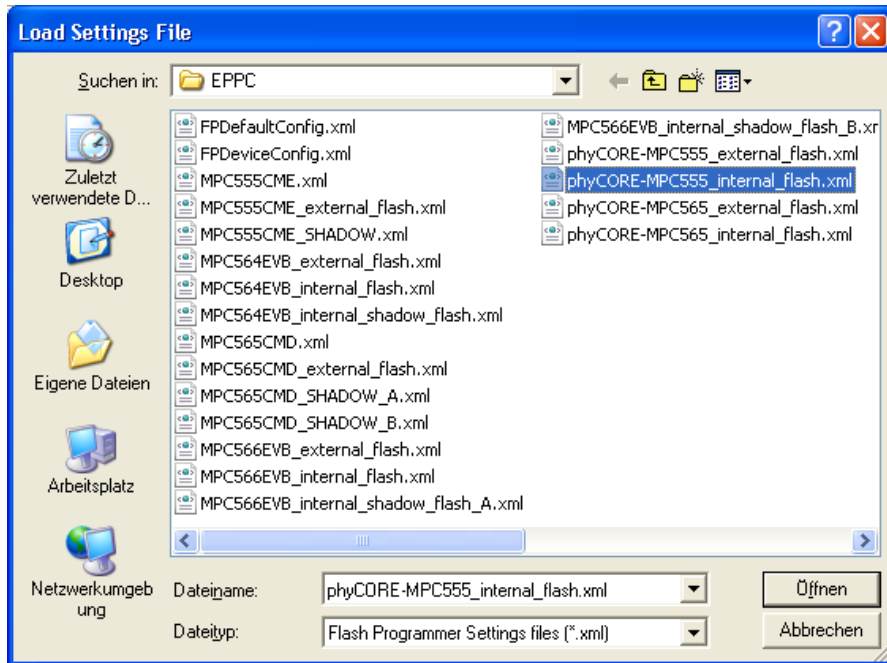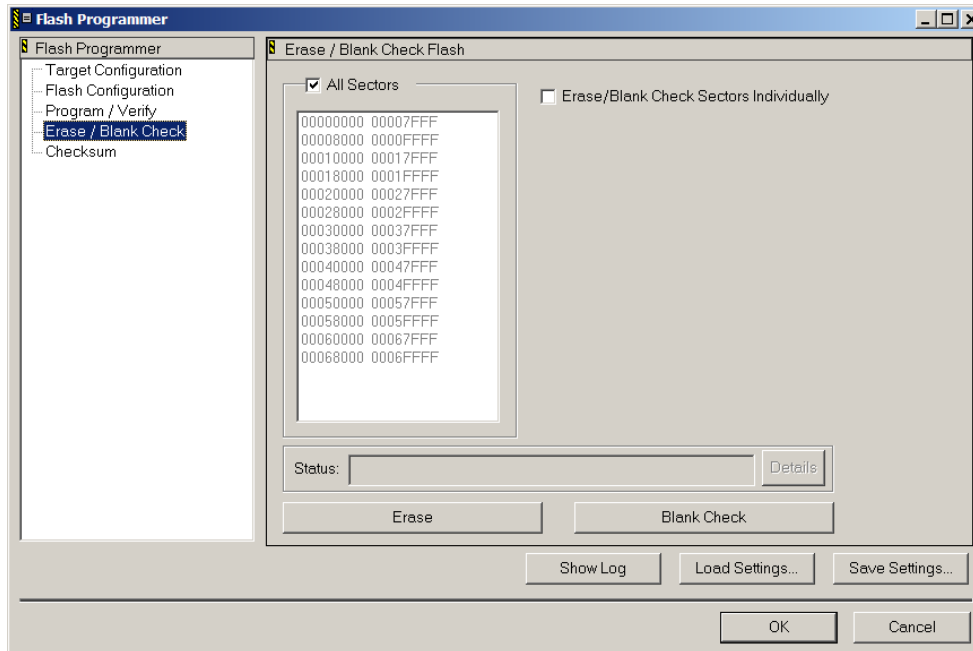

- The *Target Configuration* window will appear as follows:

　　　　　　　　　　　　　© PHYTEC Messtechnik GmbH 2009　　L-524e_5

- Click on *Flash Configuration* to check that the correct Flash device is as follows:

- Back in the **Flash** Programmer main window click on *Erase/Blank Check*.

- Click on the *Erase* button to erase the entire on-chip Flash memory.

- The Flash erase procedure will take up to 10 seconds. The *Status:* line shows the individual steps that are performed during the Flash erase.

Note:
If the *Status* indicates that the *Erase Command Failed* check that Jumper JP15 is set to position 1+2.

- If you wish, you can perform a *Blank Check* at the end of the Flash erase sequence to verify the Flash sectors are blank. Simply click on the applicable button.

- Now go to the *Program/Verify* section of the **Flash Programmer** window.
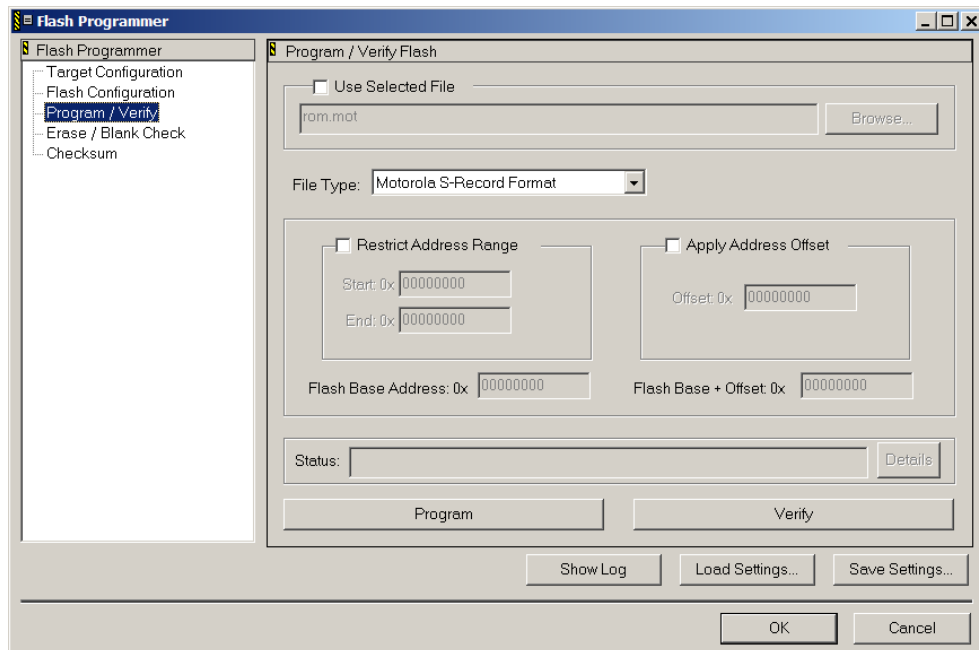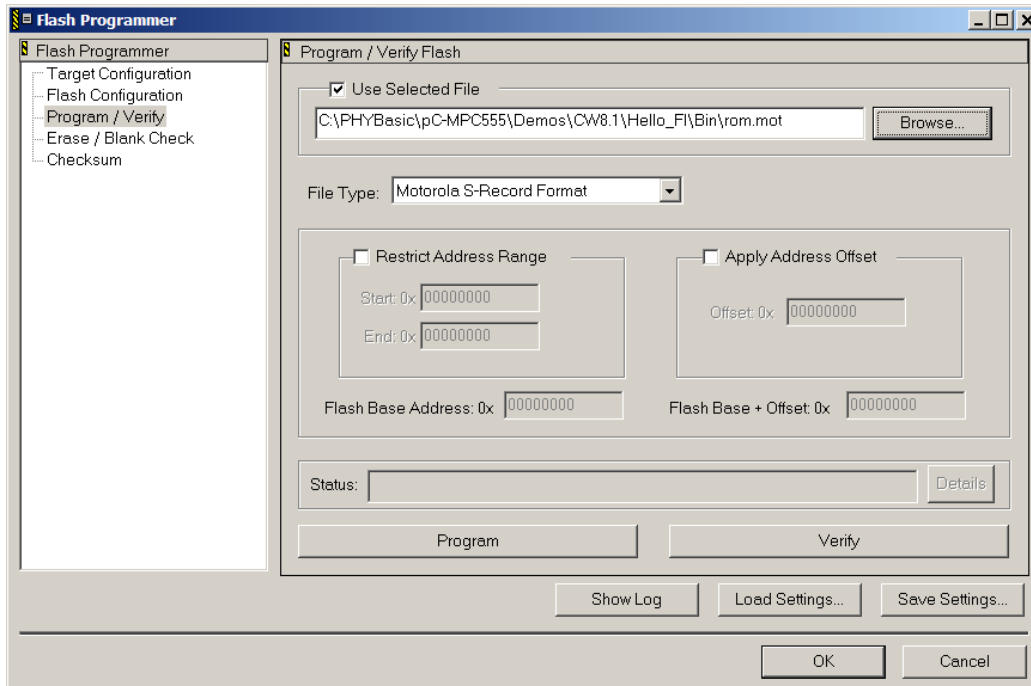

- Check *Use Selected File* and navigate to the **PHYBasic\pc-MPC555\Demos\CW8.7\Hello_Fl\Bin** folder in the *Select File to Program* window.
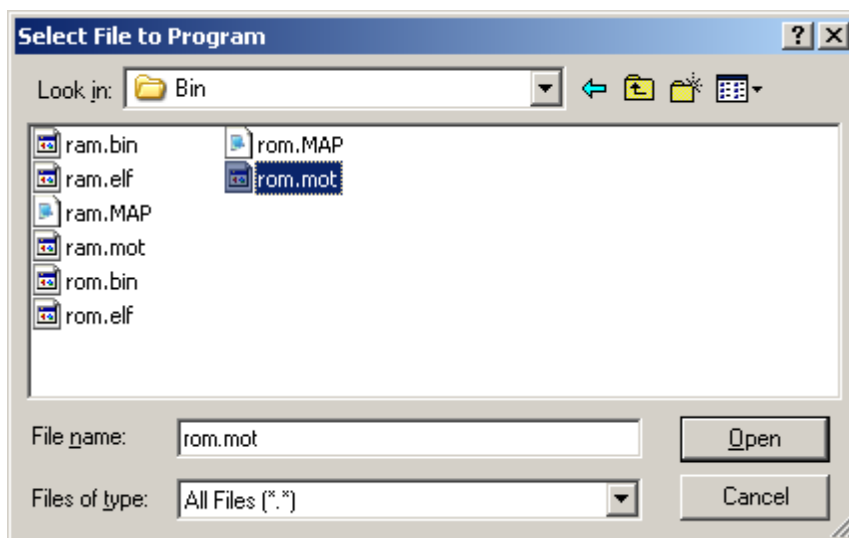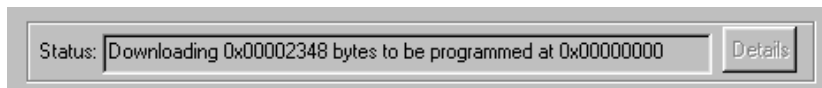

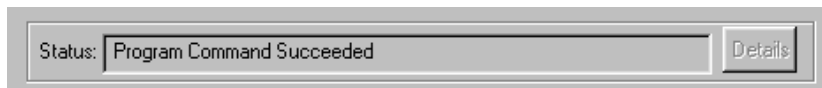- Select the **rom.mot** file from the list of available files and *Open* it.

- The **Flash Programmer** window will reappear and should look like this:

- Now click on the *Program* button to start downloading the **rom.mot** file into the on-chip Flash.
- The *Status:* line shows the individual steps that are performed during the Flash download.

- The following status line will appear at the end of the download sequence:

- If you wish to verify the Flash download simply click on the *Verify* button to start this procedure.
- Click on the *OK* button to exit the **Flash Programmer** tool.
- Click *No* when you are prompted to save these new settings.

You have now successfully downloaded the pre-existing **Hello_Fl** example program to the MPC555 on-chip Flash.

- Disconnect the power adapter from Development Board.
- **Important:** Remove Jumper JP6.
- Disconnect the DB-25 parallel cable.
- Reconnect the power adapter to the Development Board.

Only the red LED D4 on the Development Board will now flash.

- Switch back to the HyperTerminal program to view the output of the demo program, which will print the character string,
  "UART_1> Hello world
  Running out of internal FLASH"

2.4.4 Downloading Hello_Fl to the External Flash

This example demonstrates burning code to the external Flash on the phyCORE-MPC555. The example program will be downloaded to the Flash devices at U2/U3 (AM29LV type).

- Disconnect the power adapter from Development Board.
- Reconnect the DB-25 parallel cable.
- Change Jumper JP15 to position 2+3 (*refer to Figure 2*).
- Close Jumper JP6.
- Reconnect the power adapter to Development Board.
- From the pull-down menu, at the top of the ***Hello_Fl.mcp*** project window, make sure the software target "**rom**" is selected.
- Open the ***Flash Programmer*** window by selecting *Tools / FlashProgrammer* from the pull-down menu.
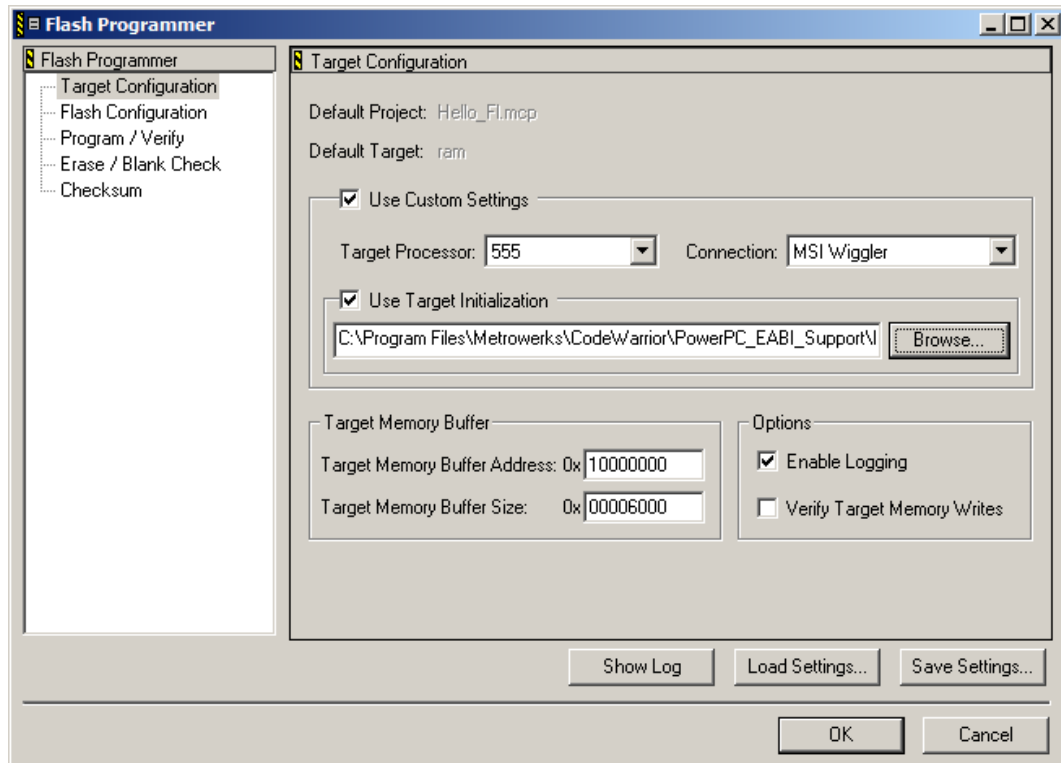
- The following window will appear:
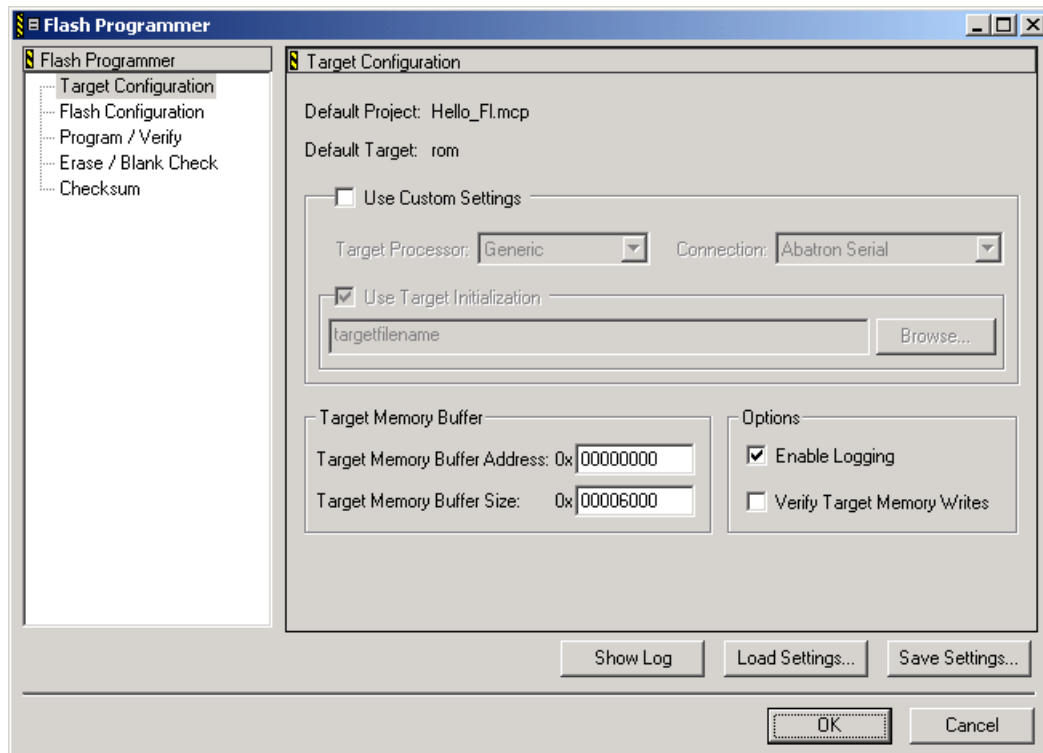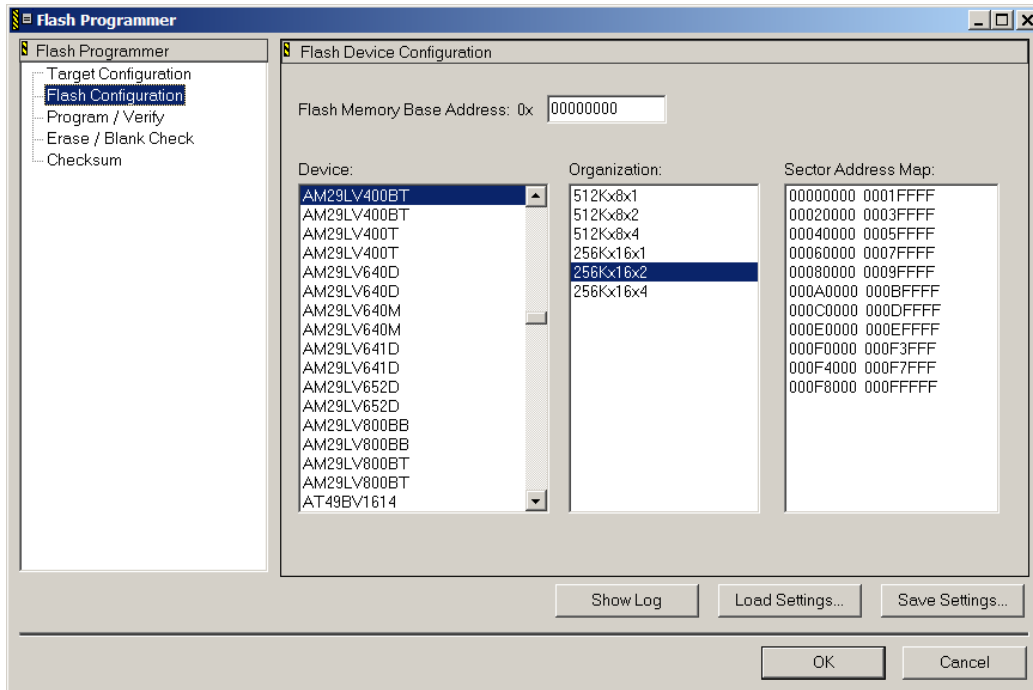

- Click on the *Load Settings...* button

- Browse to the ***phyCORE-MPC555_external_flash.xml*** file located in the
C:\Freescale\CodeWarrior EPPC 5xx V8.7\bin\Plugins\Support\ Flash_Programmer\EPPC folder and *Open* the file.
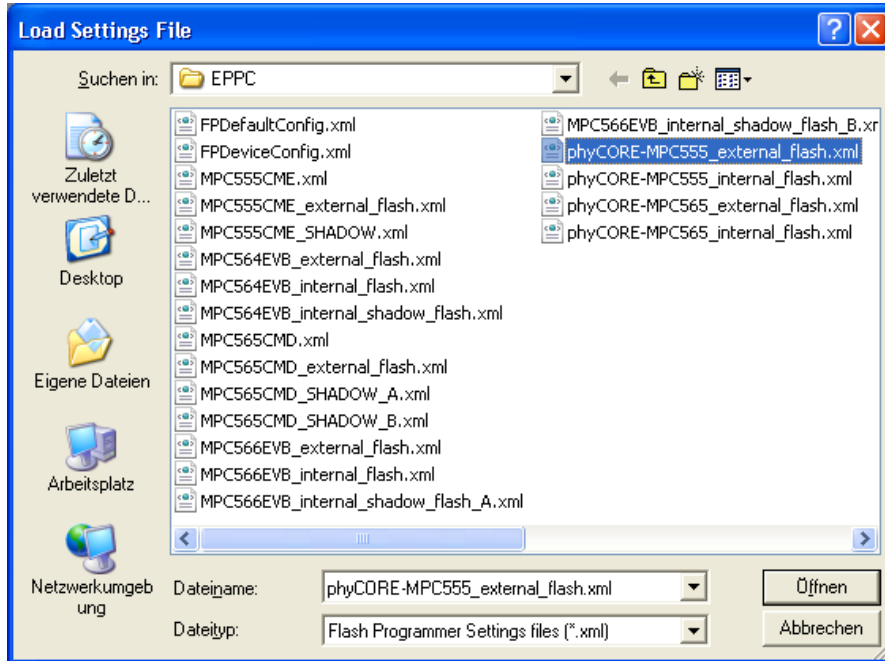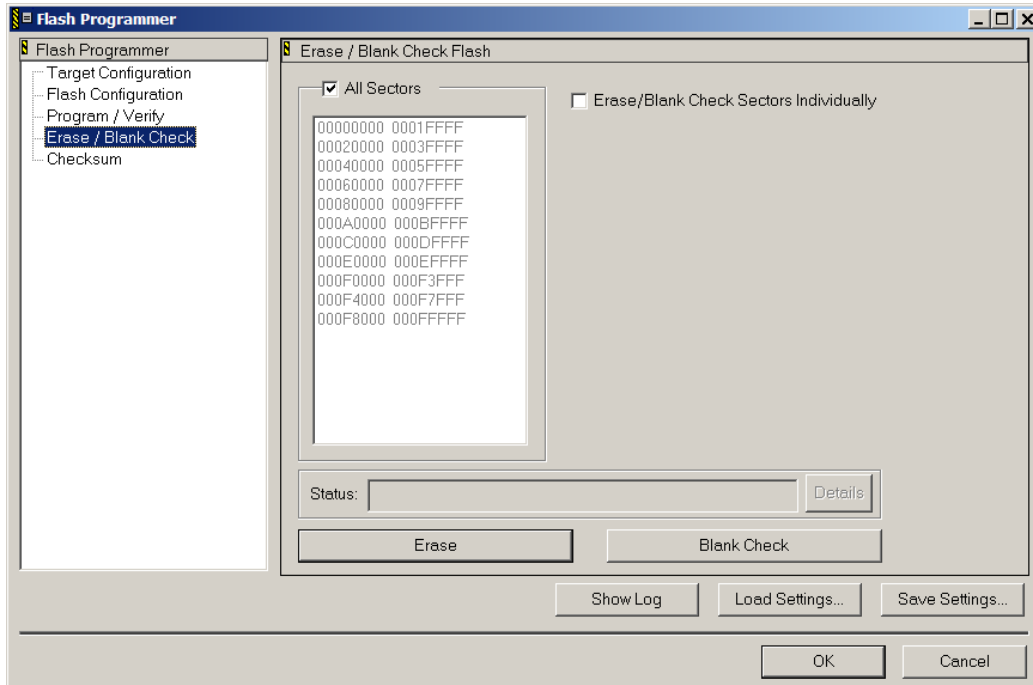

- The *Target Configuration* window will appear as follows:

- Click on *Flash Configuration* to check that the correct Flash device is as follows:

- Click on *Erase / Blank Check*. Next click on the *Erase* button.

- The Flash erase procedure will take up to 15 seconds. The *Status:* line shows the individual steps that are performed during the Flash erasure.

Note:
If the *Status* indicates that the *Erase Command Failed* check that Jumper JP15 is set to position 2+3.

- You can perform a *Blank Check* at the end of the Flash erase sequence to verify the Flash sectors are blank. Simply click on the applicable button.
- Now go to the *Program/Verify* section of the **Flash Programmer** window.
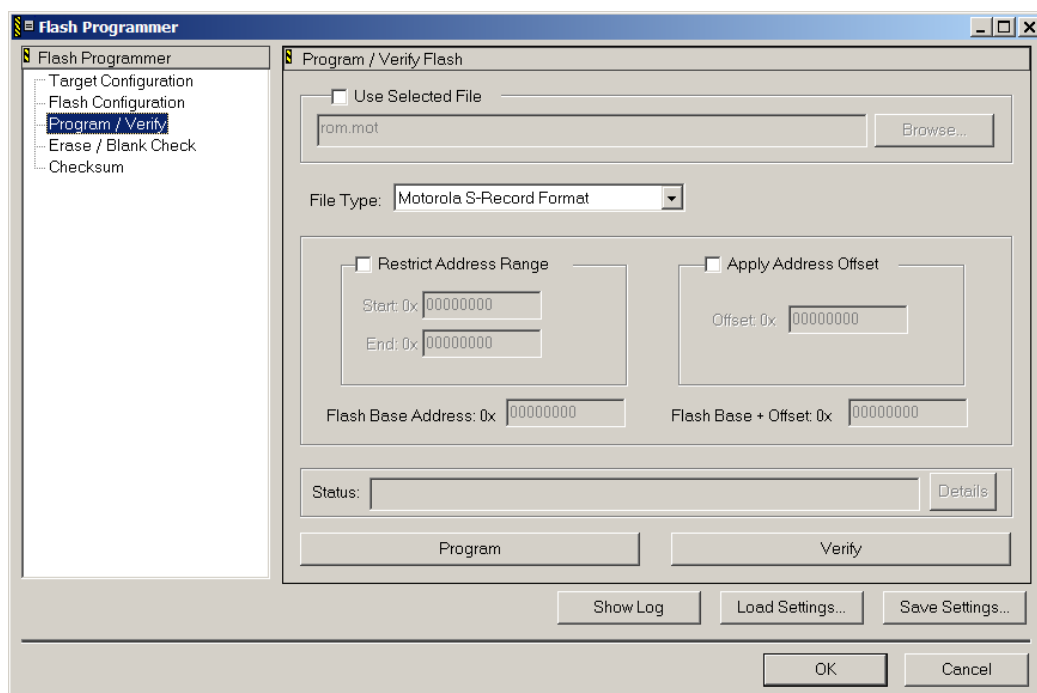
- Select the output file that will be downloaded into the external Flash. Check the *Use Selected File* box. Click on the *Browse...* button in the upper right-hand corner of this window.
- Navigate to the
  PHYBasic\pc-MPC555\Demos\CW8.7\Hello_Fl\Bin
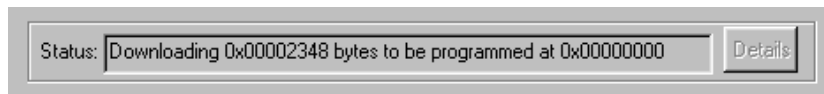 folder in the *Select File to Program* window.
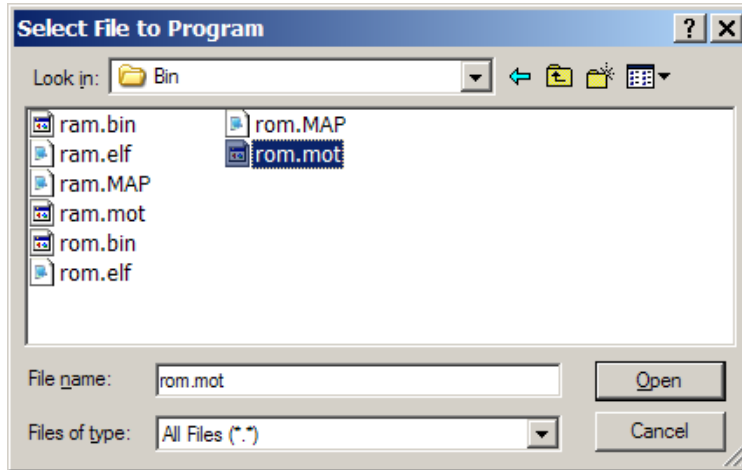
- Select the **rom.mot** file from the list of available files and *Open* it. The **Flash Programmer** window should look as follows:
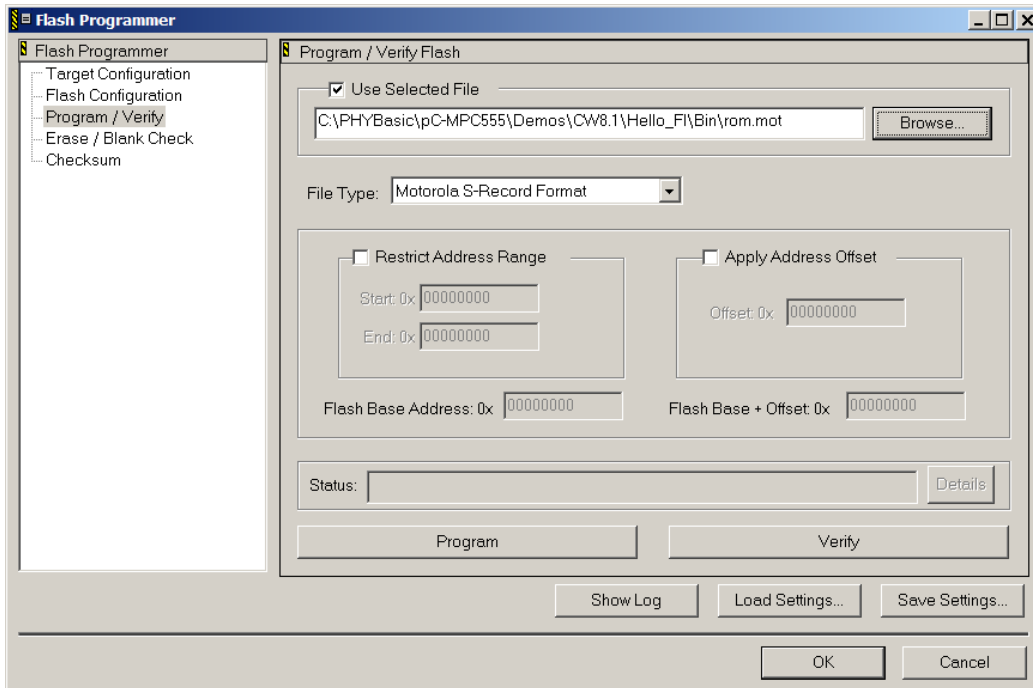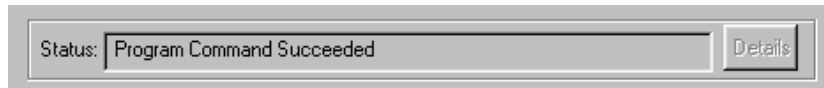

- Now click on the *Program* button to start downloading the **rom.mot** file into the external Flash.

- The *Status:* line shows the individual steps that are performed during the Flash download.

  Status: | Downloading 0x00002348 bytes to be programmed at 0x00000000 | Details

- The following status line will appear at the end of the download sequence:

  Status: | Program Command Succeeded | Details

- If you wish to verify the Flash download simply click on the *Verify* button to start this procedure.
- Click on the *OK* button to exit the **Flash Programmer** tool.
- Click *No* when you are prompted to save these new settings.

You have now successfully downloaded the **Hello_Fl** example program to the external Flash on the phyCORE-MPC555.

- Disconnect the power adapter from Development Board.
- **Important:** Remove Jumper JP6.
- Disconnect the DB-25 parallel cable.
- Reconnect the power adapter to the Development Board.

Only LED D5 (green) on the Development Board will now flash.

- Switch back to the HyperTerminal program to view the output of the demo program, which will print the character string,
  "UART_1> Hello world
  Running out of external FLASH"

2.5 Creating a New Project

For ease of creating a new project for the phyCORE-MPC555 it is recommended that the given **Hello_Fl** project is used as a template.

The **Hello_Fl** project already has the target, linker, compiler, debugger, and startup code correctly configured for the phyCORE-MPC555 target.

- Create a new directory for your project and copy the contents of **Hello_Fl** into the new directory.

- From the project window you can add or remove source files, add code libraries, modify compiler and linker settings, and compile and link a project.
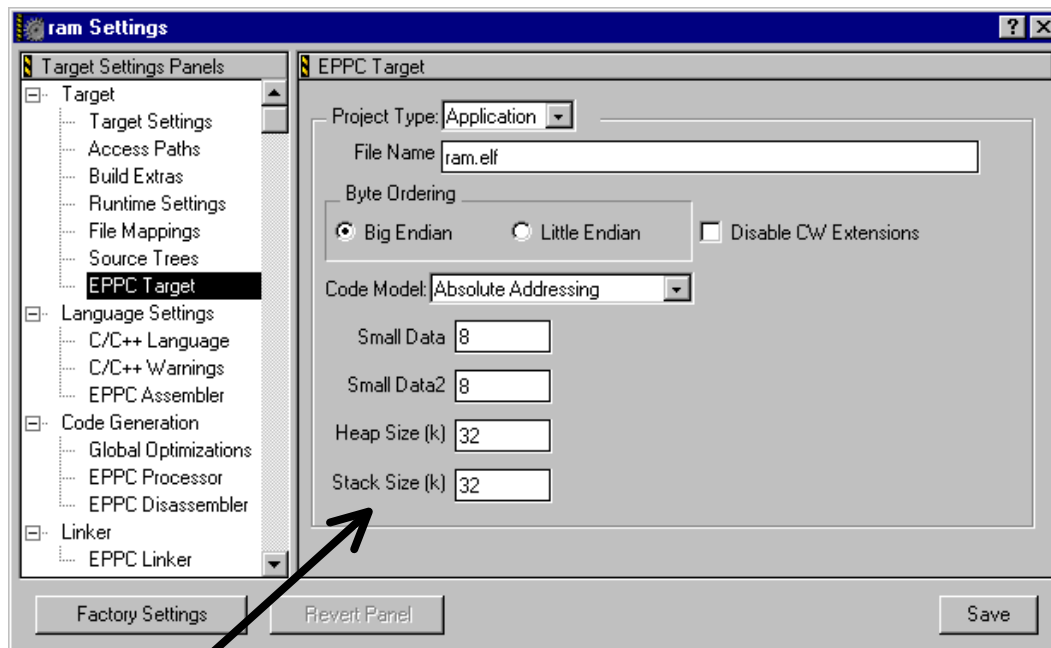
3 Advanced User Information

3.1 Stack Allocation

- Click on the *Project Settings Button* 🗐 within the project window to open the *Target Settings Panel*
- The stack location is specified by the content of the *Stack Address* edit field in the *Linker | EPPC Linker* Panel within the *Target Settings Panel*.
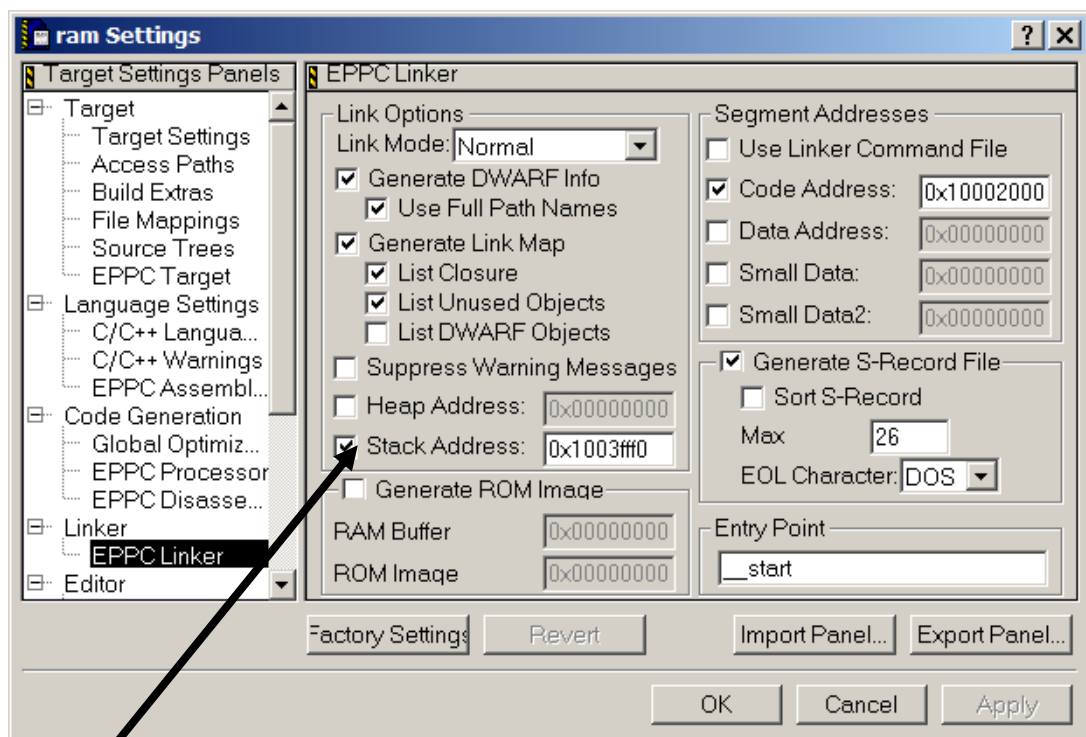

- The marked checkbox should always be active. If not, a default absolute stack address will be used, that may not adequate for your requirements.
- The *Stack Address* determines the top of the stack area. The stack grows down the number of kilobytes you specify in the *Stack Size* edit field. It is located in the *Target | EPPC Target* panel within the *EPPC Settings Panel*.

- The default value is 32 kByte. You can adjust it to your own requirements.
- For stack pointer handling the PowerPC General Purpose Register R1 is reserved and should never used for other purposes.

3.2 Dynamic Memory Allocation

- Click on the *Project Settings Button* [icon] within the project window to open the *Target Settings Panel*
- For dynamic memory allocation (malloc or new function) the heap start address and amount can be specified in the target settings *(ALT+F7)*. The heap must always reside in *RAM*.
- The amount is controlled by the setting:
    - Target – EPPC Target: Heap Size (kB), 32 (default)
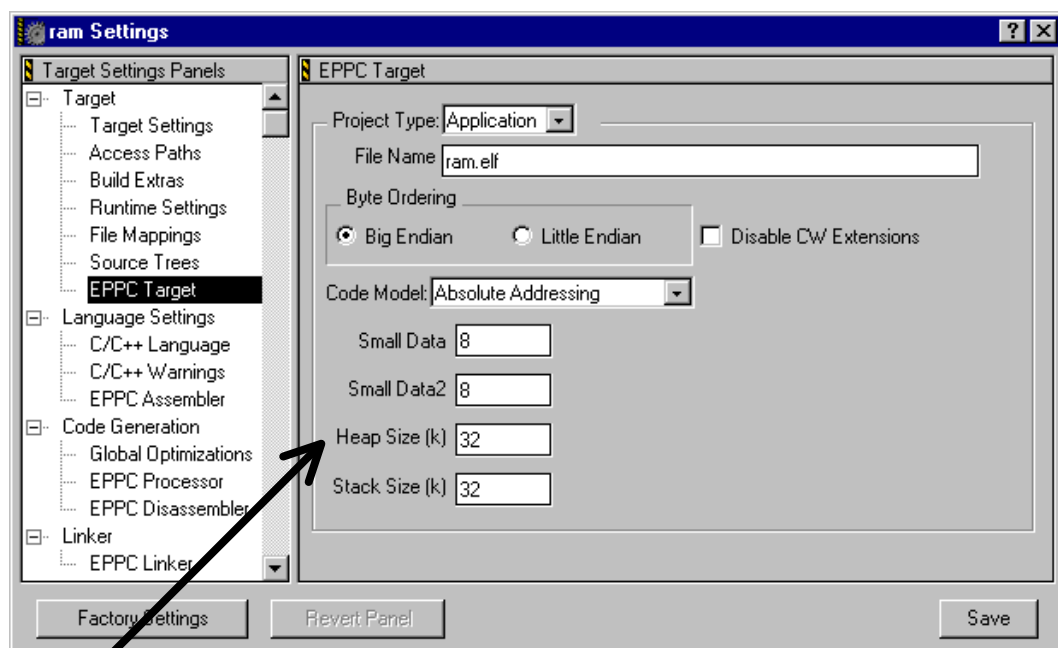
- If you do not use dynamic memory allocation, you can set this value to *zero*.
- The location of the heap is specified in the Target Linker Setting.
    - *Linker – EPPC Linker: Heap Address*
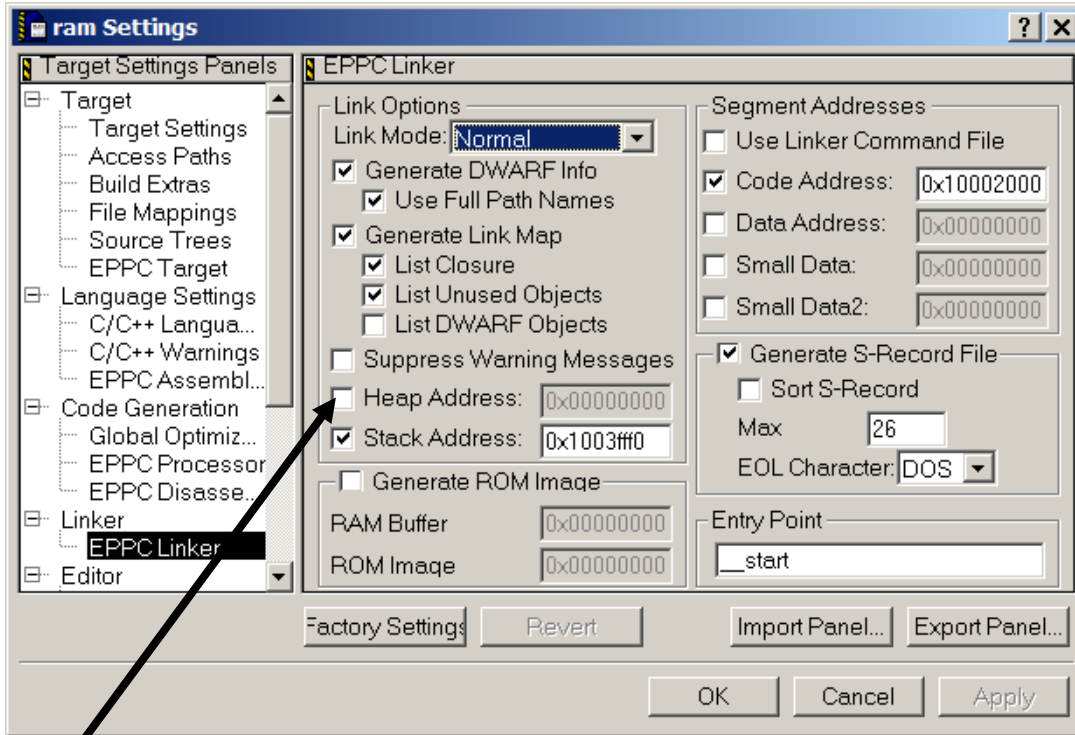
- If the check box is disabled, the top of the heap will equal the bottom of the stack. To enter a specific *heap address*, enable the check box and specify your desired address in the edit field. The address will be aligned up to the nearest *8*-byte boundary.

Note:
The memory allocation returns at runtime not as much memory space as in the Heap Size field specified. A small portion is used for buffer management.

Document:	**phyCORE-MPC555 QuickStart Instructions**
Document number:	**-524e_5, August 2009**

How would you improve this manual?

Did you find any mistakes in this manual?   page

Submitted by:

Customer number:

Name:

Company:

Address:

Return to:

      PHYTEC Technologie Holding AG
      Postfach 100403
      D-55135 Mainz, Germany
      Fax : +49 (6131) 9221-26