

phyCORE-MPC565

QuickStart Instructions

**Using Metrowerks CodeWarrior
Development Tool Chain**

Note: The PHYTEC Tool-CD includes the electronic version of the
[phyCORE-MPC565 English Hardware Manual](#)

Edition: March 2005

A product of a PHYTEC Technology Holding company

In this manual are descriptions for copyrighted products that are not explicitly indicated as such. The absence of the trademark (™) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, PHYTEC Meßtechnik GmbH assumes no responsibility for any inaccuracies. PHYTEC Meßtechnik GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. PHYTEC Meßtechnik GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, PHYTEC Meßtechnik GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. PHYTEC Meßtechnik GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

© Copyright 2005 PHYTEC Meßtechnik GmbH, D-55129 Mainz.

Rights - including those of translation, reprint, broadcast, photomechanical or similar reproduction and storage or processing in computer systems, in whole or in part - are reserved. No reproduction may occur without the express written consent from PHYTEC Meßtechnik GmbH.

	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

1st Edition: March 2005

1	Introduction to the phyCORE-MPC565	
	Rapid Development Kit	1
1.1	Rapid Development Kit Documentation	1
1.2	Overview of this QuickStart Instruction.....	2
1.3	System Requirements	2
1.4	The PHYTEC phyCORE-MPC565	4
1.5	The Metrowerks CodeWarrior Development Tool Chain	8
2	Getting Started.....	13
2.1	Installing the Metrowerks CodeWarrior Tool Chain.....	13
2.2	Installing Rapid Development Kit Software.....	16
2.3	Interfacing the phyCORE-MPC565 to a Host-PC.....	16
2.4	Launching the CodeWarrior IDE.....	20
2.4.1	Setting up a New Remote Connection.....	20
2.5	Downloading Example Code with CodeWarrior IDE	22
2.5.1	Configuring HyperTerminal Program for Hello_Fl example.....	22
2.5.2	Running the Hello_Fl.mcp Demo Project in RAM	25
2.5.3	Downloading Hello_Fl to the MPC565 On-Chip Flash.....	32
2.5.4	Downloading Hello_Fl to the External Standard Flash (PCM-019-1010E Only)	38
2.5.5	Downloading Hello_Fl to the External Burst Flash (PCM-019-3230 Only).....	45
2.6	Creating a New Project.....	53

Figures

Figure 1:	Metrowerks CodeWarrior IDE Architecture	9
Figure 2:	Development Board Overview and Suitable Jumper Settings...	17
Figure 3:	Default Jumper Settings of the phyCORE Development Board PCM-991 with phyCORE-MPC565.....	18
Figure 4:	Power Connector	19

1 Introduction to the phyCORE-MPC565 Rapid Development Kit

This QuickStart provides:

- general information on the PHYTEC phyCORE-MPC565 Single Board Computer
- an overview of Metrowerks' CodeWarrior software development tool chain, and
- instructions on how to run example programs on the phyCORE-MPC565, mounted on the PHYTEC Development Board, in conjunction with Metrowerks' CodeWarrior tools.

Please refer to the [phyCORE-MPC565 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-MPC565 Hardware Manual.

1.1 Rapid Development Kit Documentation

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

- [PHYTEC phyCORE-MPC565 Hardware Manual and Development Board Hardware Manual](#)
- [PHYTEC phyCORE-MPC565 QuickStart Instructions](#)
- [MPC565 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-MPC565 in conjunction with Metrowerks' CodeWarrior tools. The "*Getting Started*" section uses the example program "*Hello_Fl*" to demonstrate the download of user code to the phyCORE-MPC565 external RAM, internal Flash, and external Flash, using Metrowerks CodeWarrior Debugger and Flash Programmer. This section also provides hints for creating a new phyCORE-MPC565 project in Metrowerks CodeWarrior.

1.3 System Requirements

Use of this "Rapid Development Kit" requires:

- the PHYTEC phyCORE-MPC565,
- the PHYTEC Development Board with the included DB-9 serial cable, DB-25 printer port cable and AC-to-DC adapter supplying 5 V DC/min. 1.5 A,
- the PHYTEC Tool-CD,
- the Metrowerks CodeWarrior CD (the required evaluation license key needs to be requested from Metrowerks 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:

PHYTEC

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



<http://www.metrowerks.com>
sales@metrowerks.com



<http://www.freescale.com/>

1.4 The PHYTEC phyCORE-MPC565

The phyCORE-MPC565 module integrates the 32-bit Motorola MPC565 PowerPC microcontroller on an advanced PCB layout. All applicable controller signals extend to two 200-pin, high-density Molex connectors. In addition to the on-chip Flash (1 MByte), the phyCORE-MPC565 can be populated with up to 8 MByte of asynchronous standard Flash and up to 16 MByte of synchronous Burst-SRAM, addressable without wait states. Both memory device types support of the fastest interface mode of the MPC565, thereby guaranteeing maximized processing speed of externally stored code and data. The internal and external Flash supports direct on-board programming without additional programming voltages. A serial EEPROM (up to 32 KByte) is available for storing operating parameters. A Real-Time Clock with a calendar and alarm function serves as a real-time reference.

The phyCORE-MPC565 is especially suited for applications requiring processing of calculation-intensive algorithms. The MPC565's integrated 64-bit Floating Point Unit (FPU) enables hardware-level execution of elementary operations with single or double precision floating point numbers. Other controller features supported by the SBC module include three on-chip CAN 2.0B controllers; four UARTs; two SPI and one J1850 interfaces; dual 10-bit A/D with 40 (56) channels; twelve 16-bit PWM systems and three Time Processing Units (TPU) that process microcode independently of the PowerPC core. The MPC565 provides enhanced operational flexibility in that it can manage complex control and regulation tasks without connection to any peripheral devices. Each TPU has 16 programmable channels (pins) that route data to and from peripheral features, enabling generation of extremely complex timings without burdening user code. This flexibility allows implementation of the phyCORE-MPC565 in a wide range of embedded applications.

phyCORE-MPC565 Technical Highlights

- Single Board Computer in subminiature form factor (84 x 57 mm) according to phyCORE specifications
- all applicable controller and other logic signals extend to two high-density 200-pin Molex connectors
- Motorola embedded PowerPC MPC565 (40/56 MHz clock)

- **Internal Features of the MPC565:**
 - 32-bit PowerPC core, 40 / 56 MHz CPU speed
 - 64-bit Floating Point Unit
 - 36 kByte SRAM; capable of battery buffering
 - 1 MByte Flash (two independent 512 kByte blocks)
 - four UARTs
 - two SPI interfaces and one J1850 interface
 - three CAN 2.0B interfaces
 - three TPU with 16 channels each
 - six 16-bit timer systems (MOIS14)
 - twelve 16-bit PWM systems (MIOS14)
 - dual 10-bit ADC (5 μ s) with 40 (65) channels (ext. MUX)
 - multi-purpose I/O signals
 - JTAG/BDM/Nexus test/debug port

- **Memory Configuration¹:**
 - SRAM: 1 MByte to 16 MByte flow-through synchronous burst-RAM, 32-bit access, 0 wait states, 2-1-1-1 burst mode
 - Flash-ROM: 2/4 MB synchronous burst mode Flash-EEPROM, 32-bit access
2 MByte to 8 MByte asynchronous standard Flash-EEPROM, 32-bit access
 - I²C Memory: 4 kByte EEPROM (up to 32 kByte, alternatively I²C FRAM, I²C SRAM)

¹: Please contact PHYTEC for more information about additional module configurations.

- **Other Board-Level Features:**

- UART port: two RS-232 transceivers for channel A and B (RxD/TxD); also configurable as TTL
- CAN port: two CAN transceivers 82C251 for channels A and B; also configurable as TTL
- Ethernet port: 10 MBit/s CS8900A controller (I/O mode)
- System-EPLD: Lattice device ispMACH LC4064, in-system programmable, 10 free I/O signals, e.g. for additional /CS generation
- I²C Real-Time Clock with calendar and alarm function
- power-down/wake-up support via RTC or external signal
- JTAG/BDM/Nexus test/debug port
- industrial temperature range (-40...+85°C)

The phyCORE-MPC565 Development Board is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion and subsequent programming of the phyCORE-MPC565 module with high-density (0.635 mm pitch) pin header connectors. RS-232 and CAN interface signals extend from the module to dual-stacked DB-9 connectors on the Development Board, while Ethernet signals extend to an RJ-45 connector. A 40-pin Nexus connector provides a debugging interface (an unpopulated 50-pin Nexus connector is also available).

Development Board Technical Highlights

- Development Board in EURO-card dimensions (100 x 160 mm)
- high-density (0.635 pitch) Molex connector for speedy and secure insertion, and subsequent programming, of the phyCORE-MPC565 Single Board Computer module
- socket for +5VDC power supply connectivity
- 2x RS-232 at DB-9 sockets
- 2x CAN interfaces at DB-9 plugs
- RJ-45 Ethernet interface
- DB-25 plug for BDM interface to support direct connection to PC printer port, on-board signal conversion
- 10-pin header connector for interfacing to external BDM devices
- 40-pin Nexus interface (populated)
- space available for population of 50-pin Nexus connector
- various power and status LEDs for monitoring all available supply voltages and specific controller signals
- reset and boot push buttons
- Expansion Bus: address, data, interface and all applicable I/O signals route from implemented phyCORE module to 2 x 80-pin Molex connectors, enabling connectivity to PHYTEC **Add-On hardware**

1.5 The Metrowerks CodeWarrior Development Tool Chain

Metrowerks 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 Metrowerks 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. Metrowerks 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.

Metrowerks 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\Metrowerks\CodeWarrior\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.

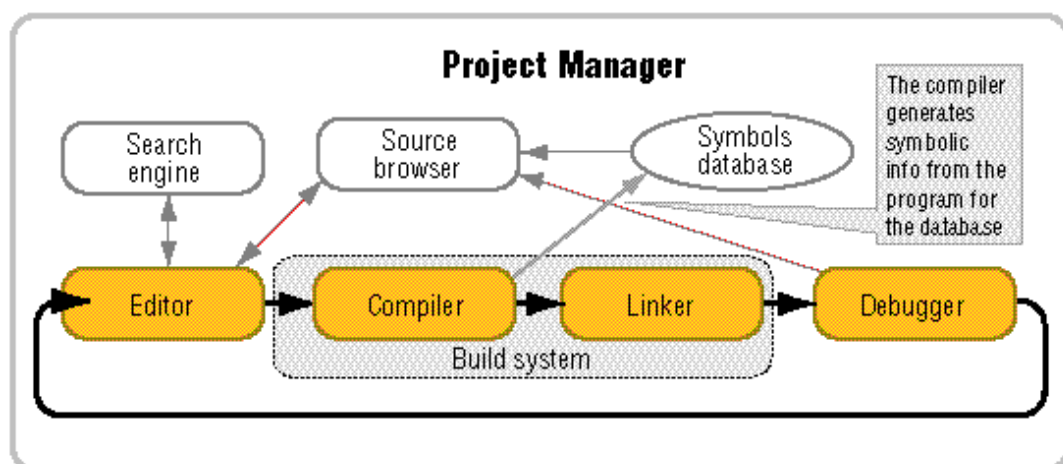


Figure 1: Metrowerks CodeWarrior IDE Architecture

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 Motorola 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 Metrowerks CodeWarrior Debugger supports the following debug options of the phyCORE-MPC565 target.

- A hardware-level debugging option is provided via a PC-to-target connection using the MPC565 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-MPC565 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 Metrowerks CodeWarrior for PowerPC Embedded Systems
- installing the Rapid Development Kit software
- interfacing the phyCORE-MPC565, mounted on the Development Board, to a host-PC
- downloading example user code from a host-PC to the phyCORE-MPC565 on-board memory

2.1 Installing the Metrowerks CodeWarrior Tool Chain

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

The required 30-day license key needs to be requested from Metrowerks using the integrated registration wizard. This is required to use Metrowerks CodeWarrior Tool Chain.

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

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

The default destination location is

C:\Program Files\Metrowerks\CodeWarrior. 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 enclosed **PHYTEC phyCORE-MPC565 Tool-CD** to appropriate locations in the CodeWarrior default installation on your host-PC. These files enable initialization of and communication to the PHYTEC phyCORE-MPC565 target hardware from within the CodeWarrior IDE.

- Copy the files ***phyCORE-MPC565_BDM_init.cfg*** and ***5xx_IOCTL_init.cfg*** located on the PHYTEC phyCORE-MPC565 Tool-CD under:
Tools\CW8.1\PowerPC_EABI_Support\Initialization_Files\Bdm
to the following default location in your the Metrowerks CodeWarrior installation path:

***C:\Program Files\Metrowerks
PowerPC_EABI_Support\Initialization_Files\Bdm\...***

- Next, copy the entire folder called ***Phytec_phyCORE-MPC565_serial*** located on the Tool-CD under:
Tools\CW8.1\PowerPC_EABI_Tools\MetroTRK\Transport\ppc
to the following default location in your the Metrowerks CodeWarrior installation path:

***C:\Program Files\Metrowerks
PowerPC_EABI_Tools\MetroTRK\Transport\ppc\...***

- Next, copy the file *FPDeviceConfig.xml* located on the Tool-CD under: *Tools\CW8.1\bin\Plugins\Support\Flash_Programmer\Eppc* to the destination folder:
C:\Program Files\Metrowerks_MPC5xx\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 you need to configure the printer port to either EPP or SPP mode. It is not recommended to use the ECP mode in conjunction with the BDM interface. The printer port mode is selected in your System BIOS.

2.2 Installing Rapid Development Kit Software

The current version of the PHYTEC Tool-CD does not provide a setup program for automatic installation. You will need to manually copy the demos, manuals, and other support documents for this Rapid Development Kit to your local hard drive.

All path and file statements within this QuickStart Instruction are based on the assumption that you choose *C:\PHYBasic* as the default path when copying files from Tool-CD to hard drive.

- Copy the the entire *PHYBasic* folder from the **PHYTEC phyCORE-MPC565 Tool-CD** to you local hard drive. Make sure to remove the read-only attribute from all files copied to you hard drive.

2.3 Interfacing the phyCORE-MPC565 to a Host-PC

Connecting the phyCORE-MPC565 (part # PCM-019-xxxx), mounted on the PHYTEC Development Board (part # PCM-991), to your computer is simple:

- If the phyCORE module is not already installed, 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.

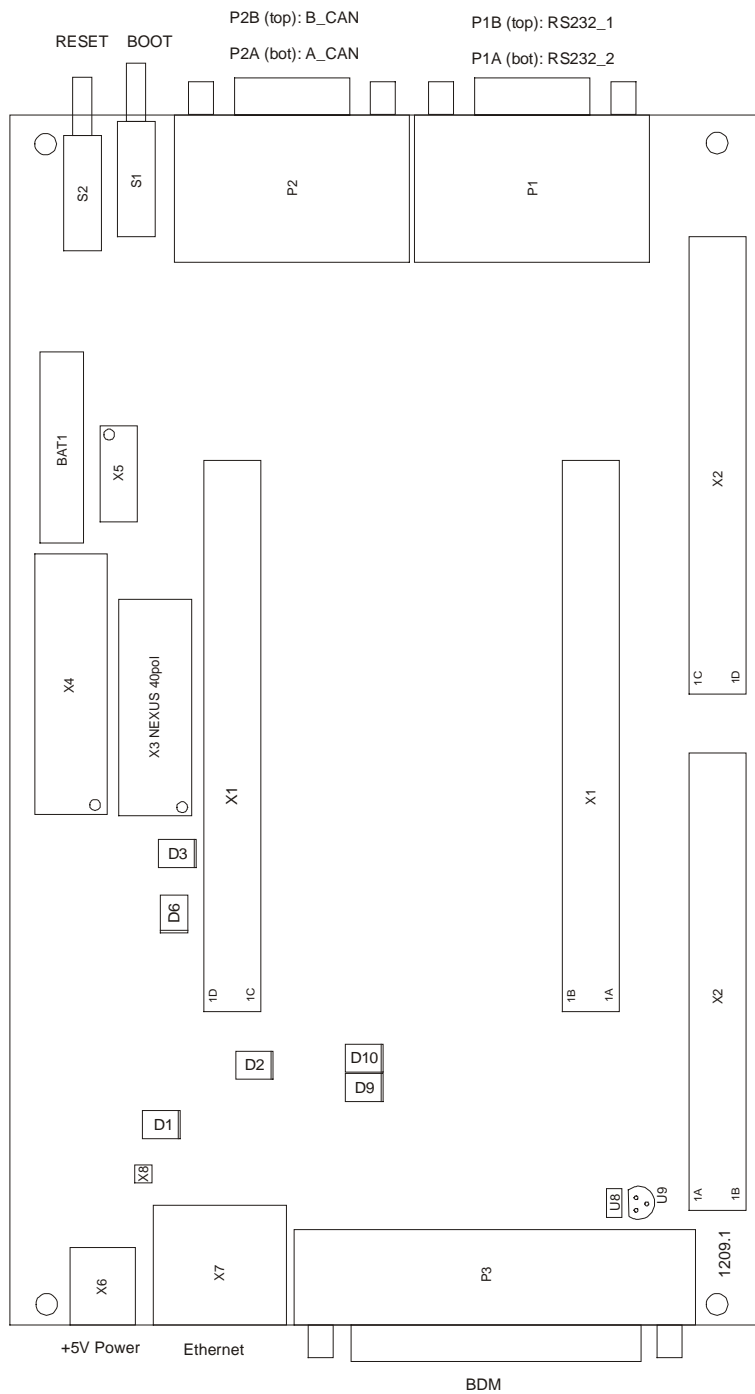


Figure 2: Development Board Overview and Suitable Jumper Settings

- Configure the jumpers on the phyCORE Development Board as indicated in the figure below. This correctly routes the interface signals to applicable connectors and establishes signal connection between on-board peripherals and the phyCORE-MPC565 module.

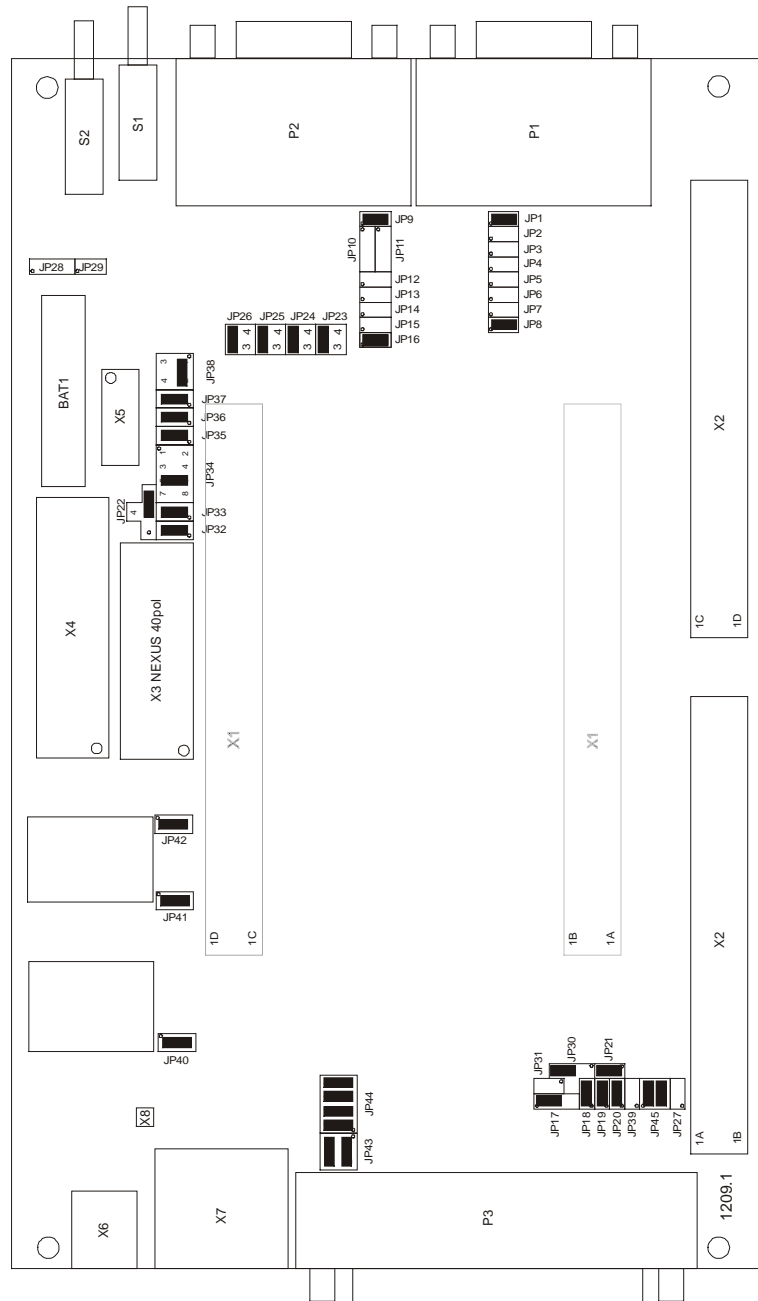


Figure 3: Default Jumper Settings of the phyCORE Development Board PCM-991 with phyCORE-MPC565

- Connect the DB-25 connector of the Development Board (P1) to the LPT1 port of your host-PC using the included DB-25 cable. This connection is used for the communication of the Metrowerks CodeWarrior IDE with the phyCORE-MPC565.

Be sure that your printer port is configured for EPP or SPP mode. This setting is typically done in the BIOS setup of the host-PC.

- Connect the upper socket P1B of the double DB-9 connector 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 5VDC power adapter, connect the power socket on the board (X6) to a power supply (*refer to Figure 4 for the correct polarity*). The phyCORE module/Development Board combination requires a 5VDC@1.5A $\pm 3\%$ regulated supply.

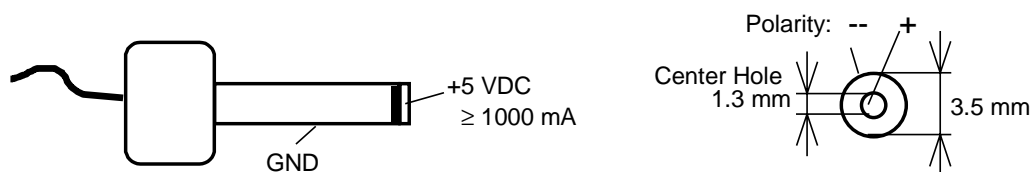


Figure 4: Power Connector

The LEDs D1, D2 and D3 (three green LEDs) should light, indicating that all voltages are supplied to the board. Also D10 (red LED) is illuminating which indicates that the MPC565 processor is waiting in debug mode.

The phyCORE-MPC565 should now be properly connected via the Development Board to a host-PC and power supply and you are now ready to use the Metrowerks CodeWarrior tool chain to establish communication. This phyCORE/Development Board combination shall also be referred to as "target hardware".

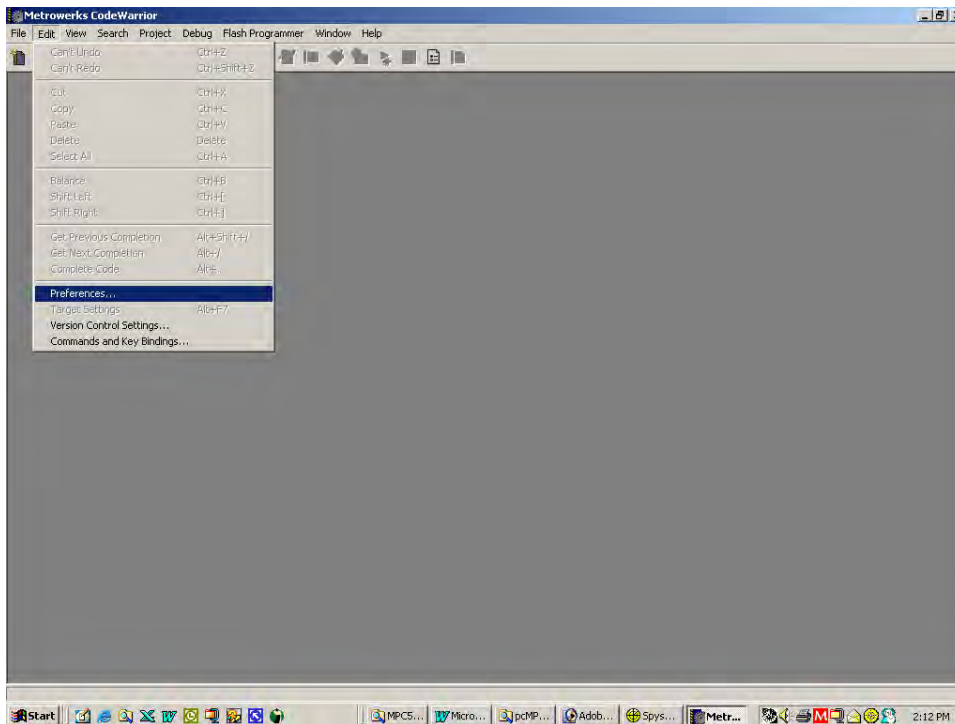
2.4 Launching the CodeWarrior IDE

- Launch the Metrowerks *CodeWarrior IDE* by double-clicking on the *CodeWarrior* icon or by selecting *CodeWarrior* from within the *Programs\Metrowerks CodeWarrior\ CodeWarrior MPC5xx Edition V8.1\ CodeWarrior IDE*.

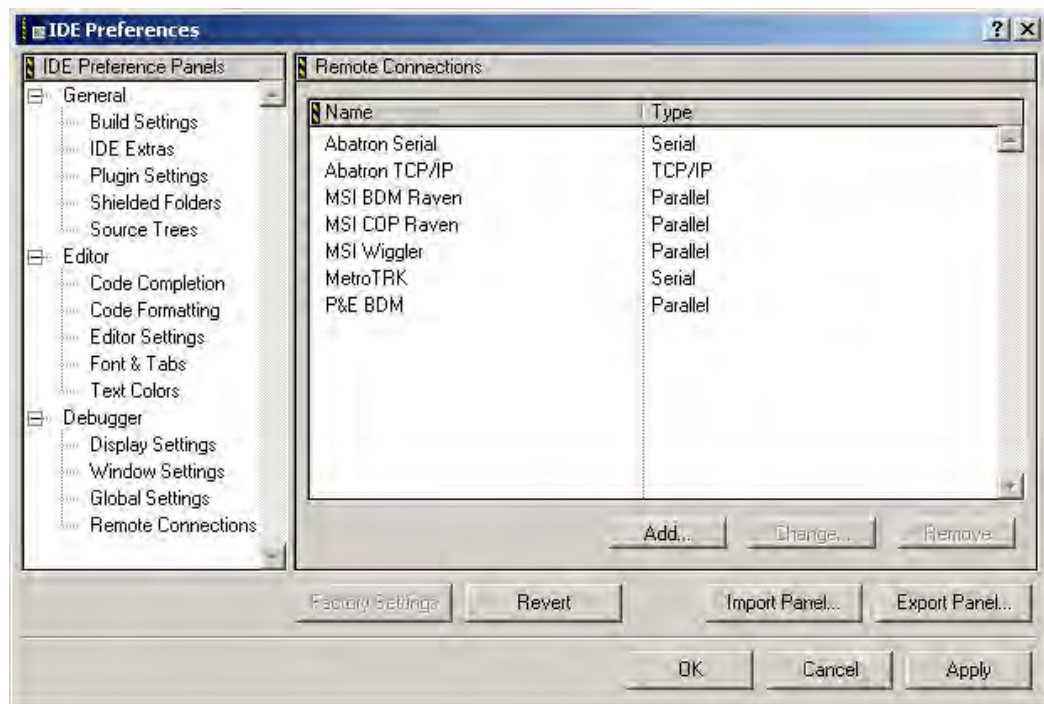
The following steps are required to configure necessary debugger communication preferences.

2.4.1 Setting up a New Remote Connection

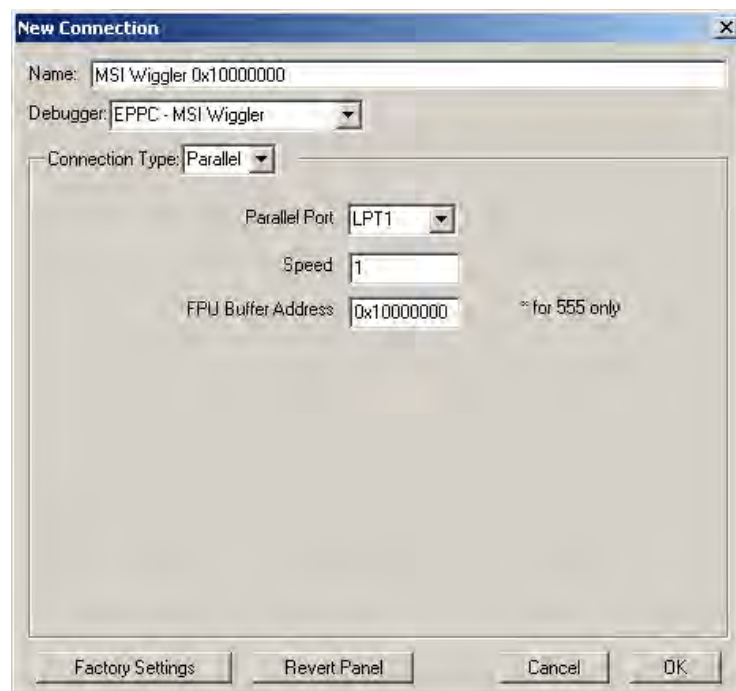
- Select *Edit / Preferences* from the pull-down menu bar:



- In the *IDE Preference Panels* window, select *Debugger -> Remote Connections*.



- Click on the *Add* button and enter the settings shown below in the *New Connection* window:



Name: MSI Wiggler 0x10000000
Debugger: EPPC - MSI Wiggler
Connection Type: parallel
Parallel Port: LPT1
Speed: 1
FPU Buffer Address: 0x10000000

- Confirm by clicking the *OK* button.
- You should now be able to view the new connection *MSI Wiggler 0x10000000* in the *Remote Connections* window.
- Confirm with the *Apply* button and click *OK*.

2.5 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 LED D6.

2.5.1 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* bar.
- The HyperTerminal main window will now appear.
- Double-click on the HyperTerminal icon *Hypertrm* to create a new HyperTerminal session.

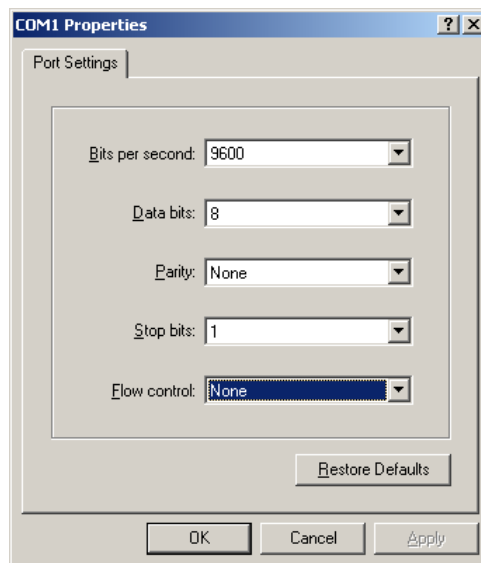
- Enter a name such as "*Connect to Com1*" (be sure to use an available COM port for your system) and select an icon of your choice to establish a new connection using HyperTerminal.



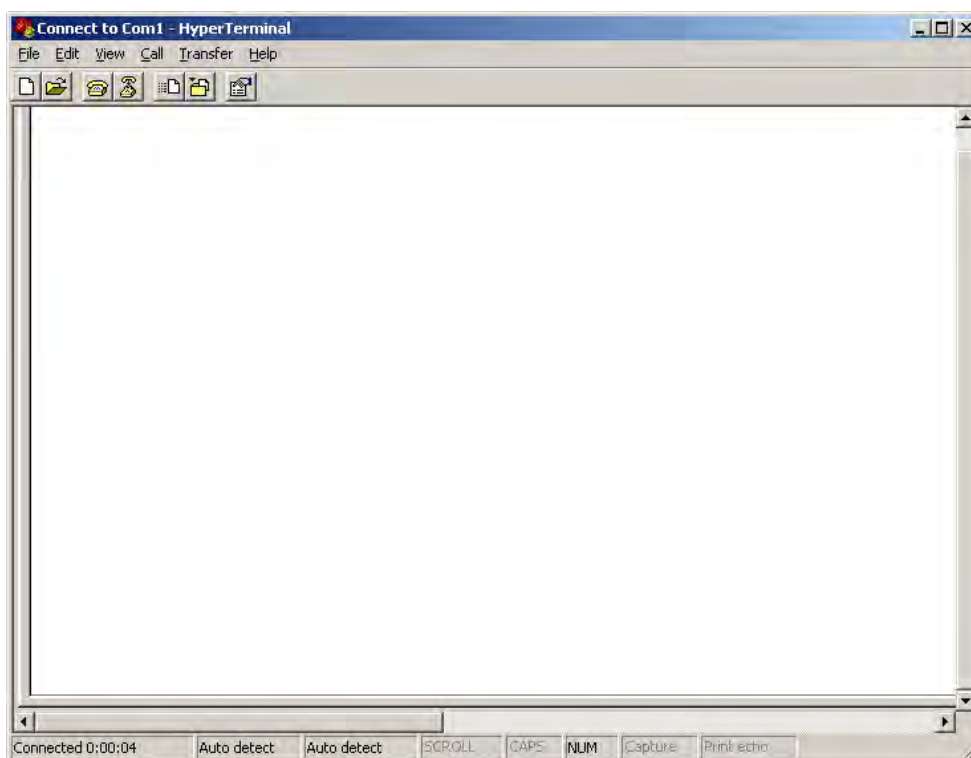
- Click *OK* to advance to the connection window.



- Click *OK* again to advance to the next window (COM1 Properties). Then set the following COM parameters: Bits per second = 9600; Data bits = 8; Parity = *None*; Stop Bits = 1; Flow Control = *None*.



- Selecting *OK* advances you to the COM Direct – HyperTerminal monitoring window. Notice the connection status report in the bottom lower corner of the window.



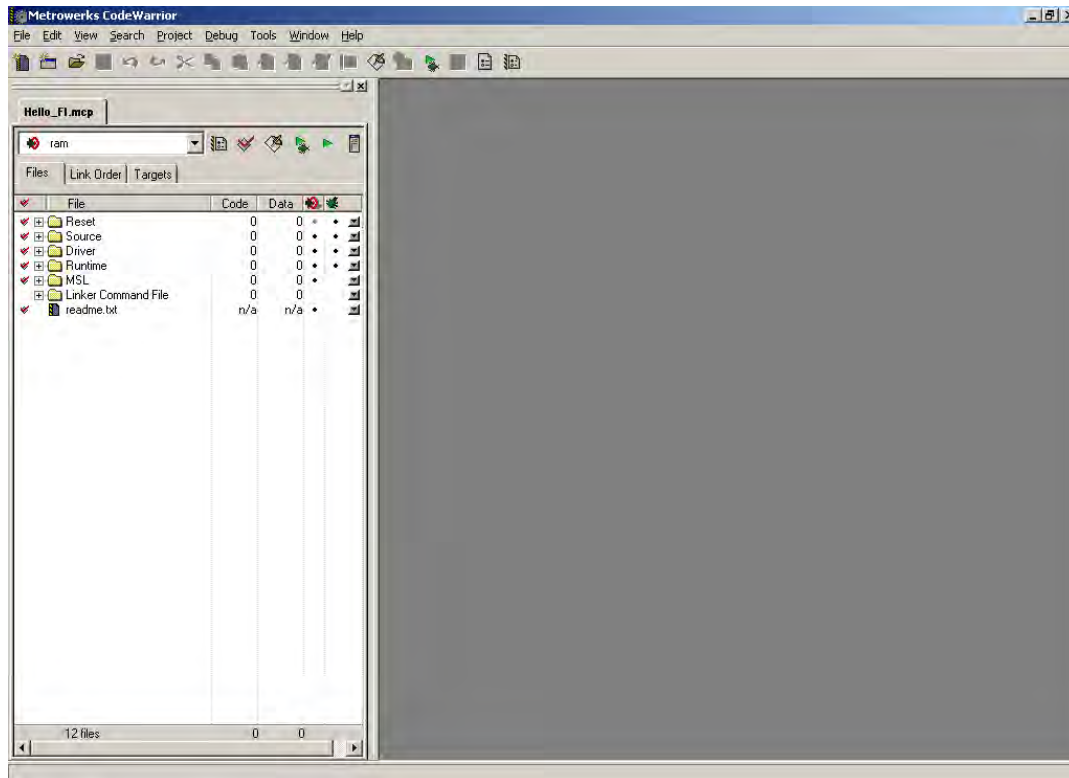
HyperTerminal is now prepared to view the output of the example program once it is executed.

2.5.2 Running the Hello_Fl.mcp Demo Project in RAM

- Return to the Metrowerks CodeWarrior IDE, leaving the HyperTerminal session open and connected to the target hardware.
- Open the ***Hello_Fl.mcp*** project file by selecting *File \ Open* from the pull-down menu and navigating to ***PHYBasic\pc-MPC565\Demos\CW8.1\Hello_Fl.***




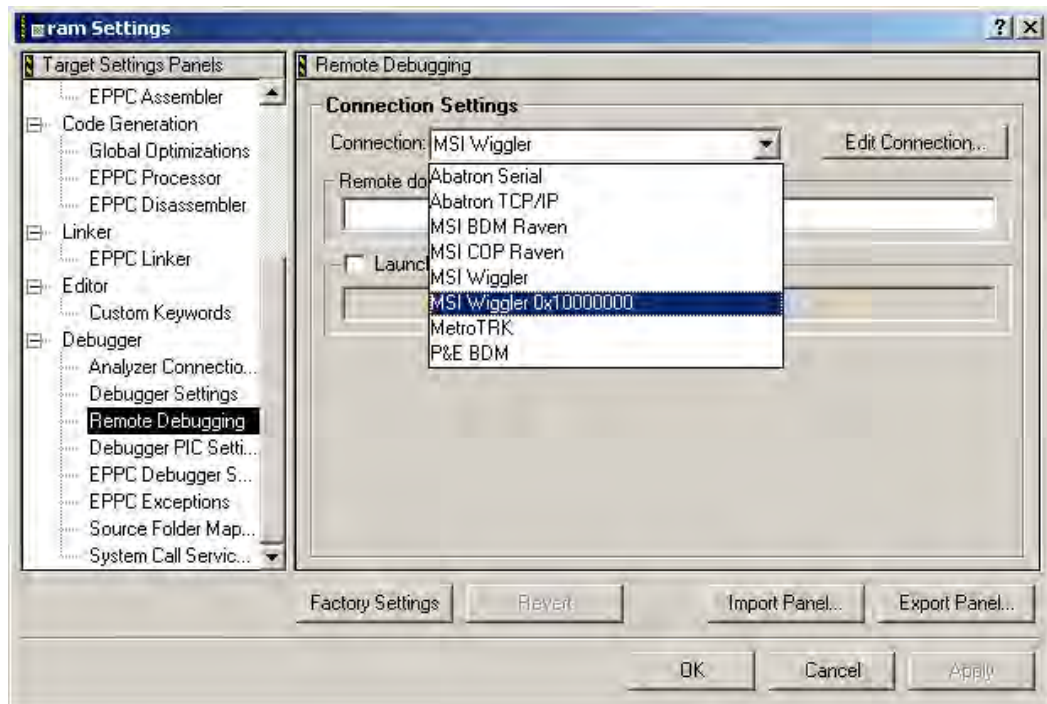
- The ***Hello_Fl.mcp*** project file will open in a project window.




If not already selected, select the software target "**ram**" from the pull-down at the top of the project window.

The new MSI Wiggler connection, which was added in *section 2.4.1*, must be set in the *ram Settings* as shown below.

- Open the *ram Settings* by clicking on the *ram Settings* icon  on the project toolbar or in the main menu select *Edit / ram Settings*.
- In the *Remote Debugging, Connection Settings*, select the *MSI Wiggler 0x10000000* and click *OK*.




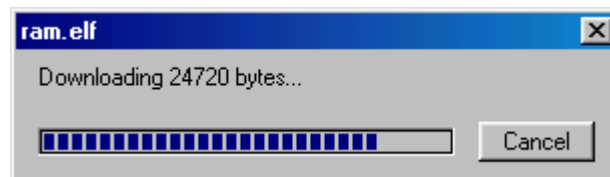
- Back in the project window, notice the red check marks in front of the individual files. This indicates that the project is not up to date and a new build should be performed. This typically happens when you open an existing project for the first time.
- Perform the build process by clicking the *Make* 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.

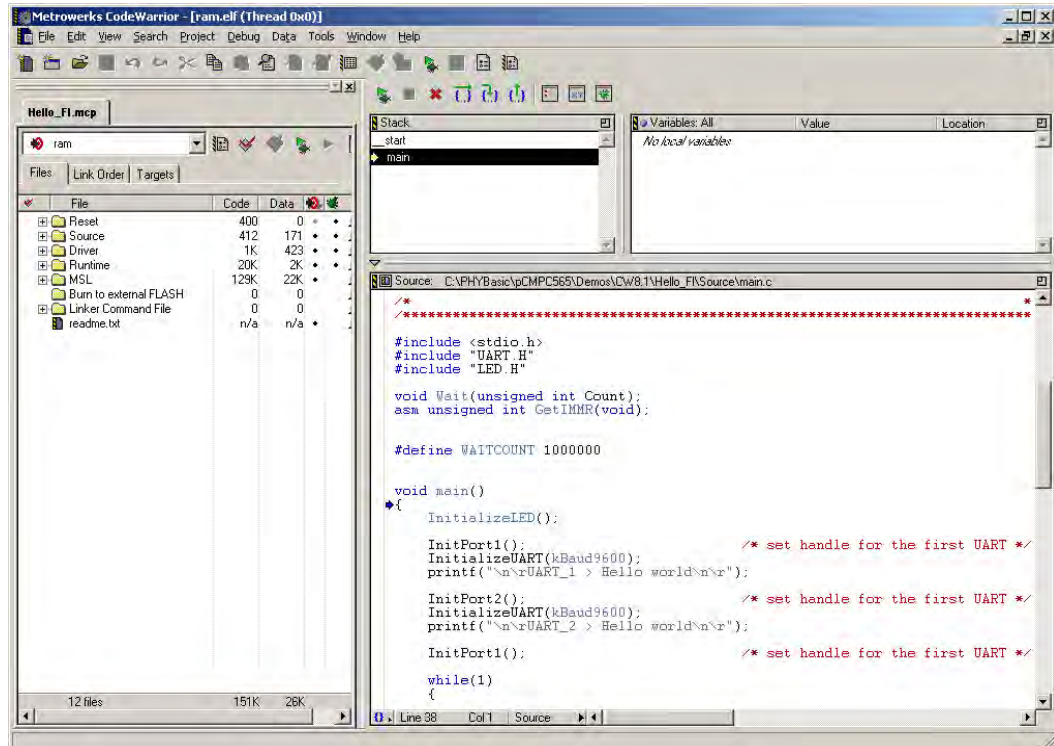
Hint:


If an error message should appear "Cannot open file" go into the EPPC Debugger Settings and make sure the file "phyCORE-MPC565_BDM_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.

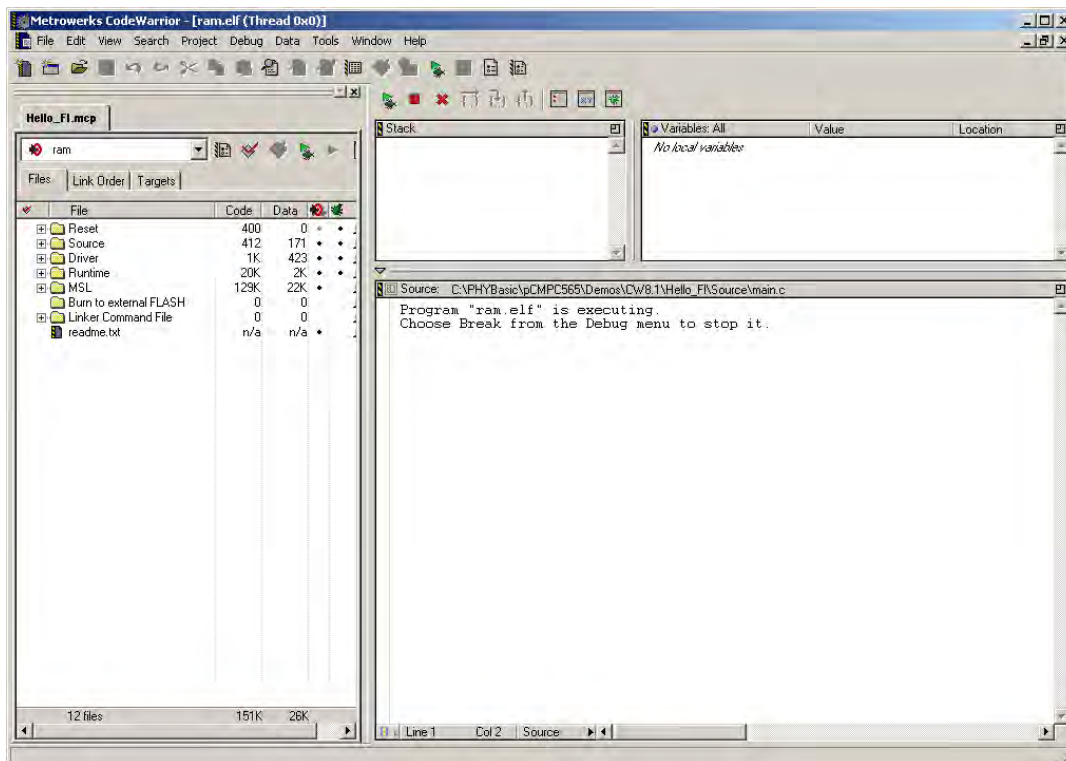
- Start the download process by clicking on green debug  arrow in the upper right section of the project window. You can also perform the download by pressing the <F5> function key or selecting the *Project / Run* item in the main menu bar.
- The following status window will appear during the download.



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

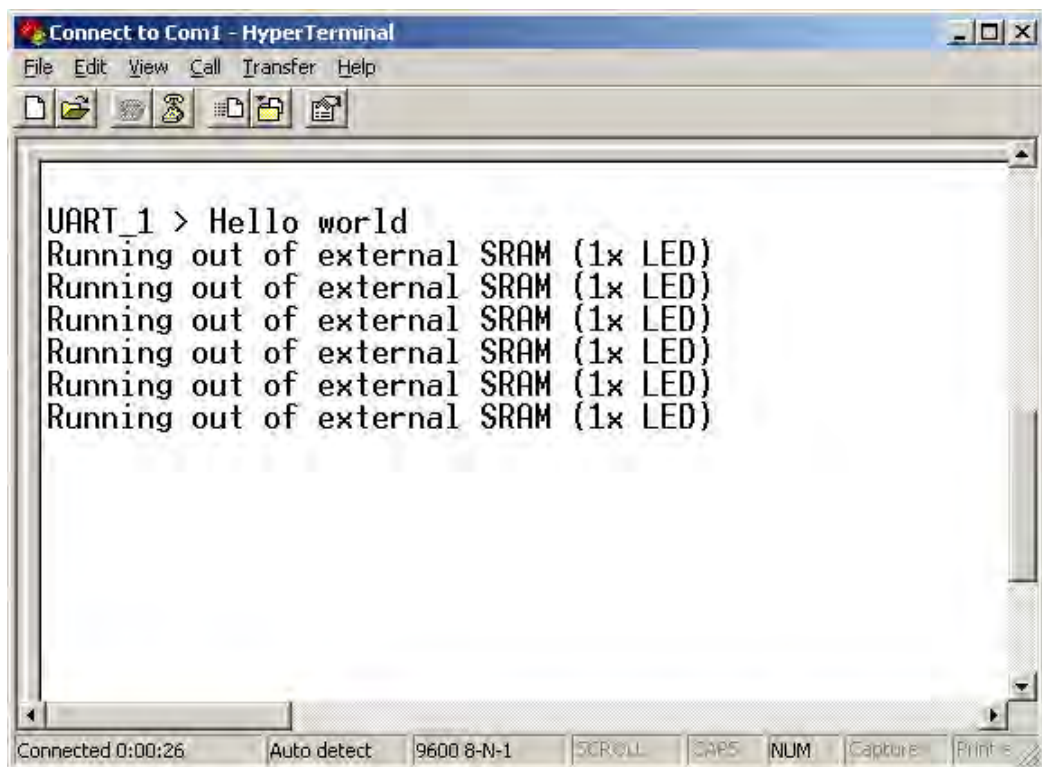


- Click on the green debug arrow  at the top of the project window (or press <F5> or select *Project / Run* in the main tool bar). This will run the main code, starting from the location indicated by the blue arrow in the source window [first line after main()]. The lower portion of the *ram.elf (Thread 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 (1x LED)"***

The red LED D6 will flash in a pattern with short on and long off intervals.

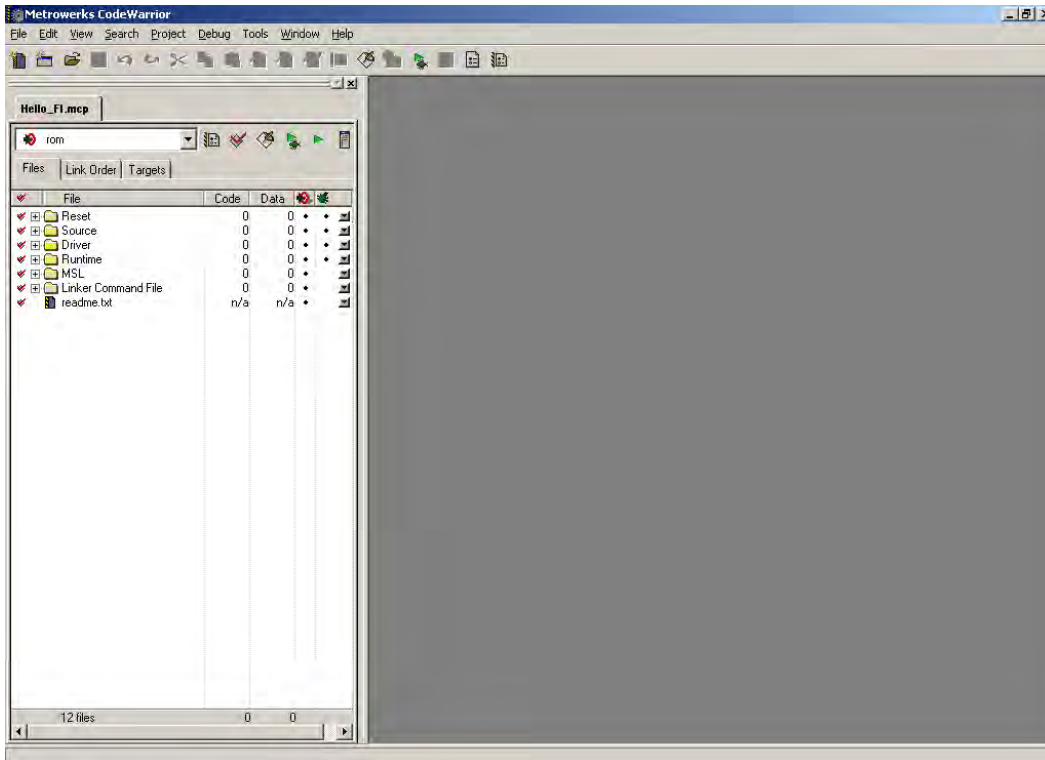


```
UART_1 > Hello world
Running out of external SRAM (1x LED)
Running out of external SRAM (1x LED)
Running out of external SRAM (1x LED)
Running out of external SRAM (1x LED)
Running out of external SRAM (1x LED)
Running out of external SRAM (1x LED)
```

- 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).

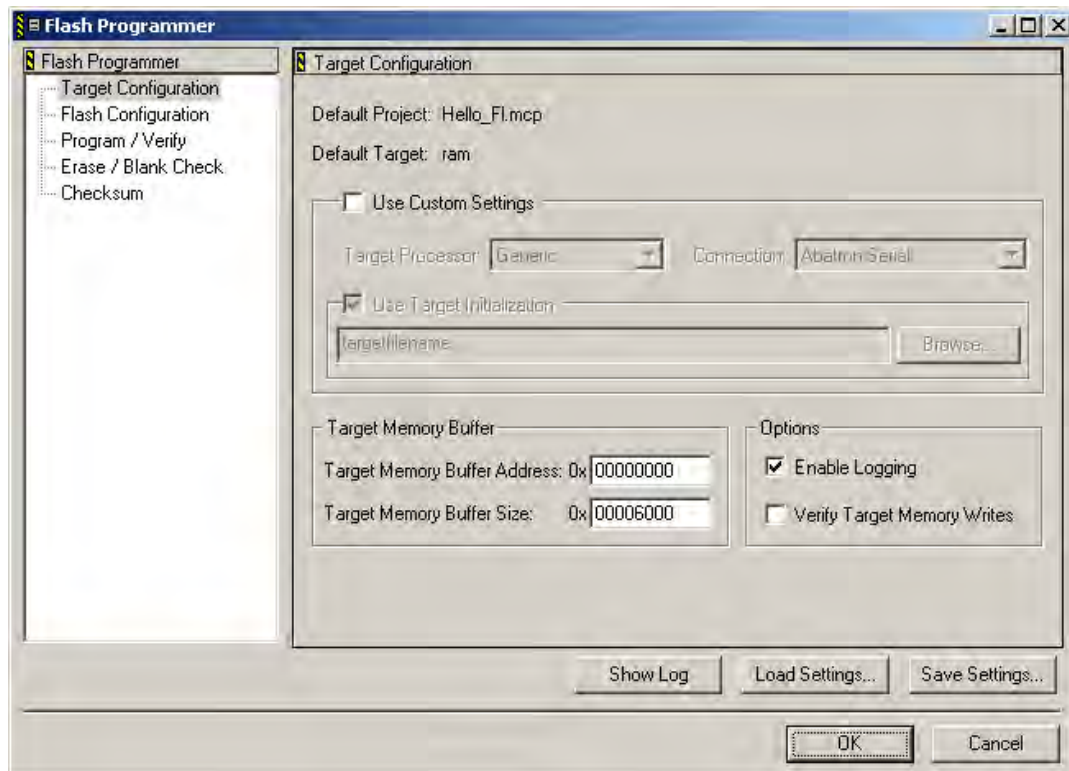
2.5.3 Downloading Hello_Fl to the MPC565 On-Chip Flash

- Make sure Jumper JP17 on the phyCORE-MPC565 Development Board is closed at default position 1+2 (*refer to Figure 3*).
- 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:

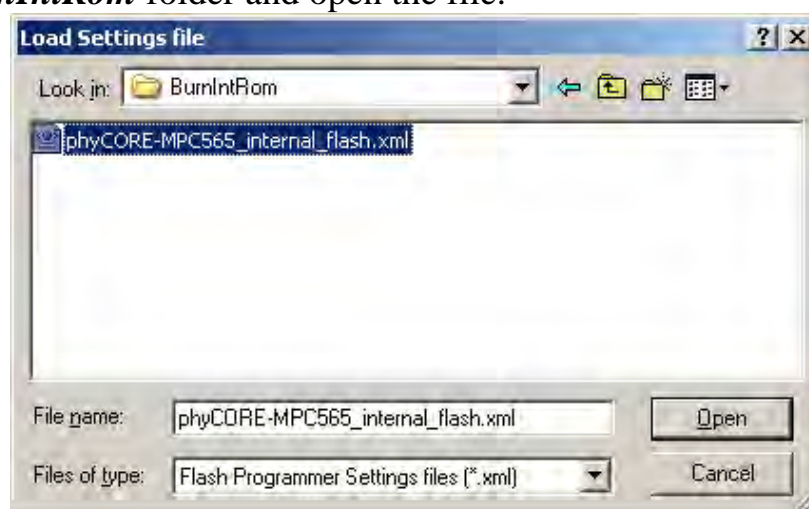


- Perform the build process by clicking the *Make* icon  or select *Project / Make* from the IDE pull-down menu or use the <F7> function key.

- 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-MPC565_internal_flash.xml** file located in the **PHYBasic\pc-MPC565\Demos\CW8.1\Hello_Fl\BurnIntRom** folder and open the file.

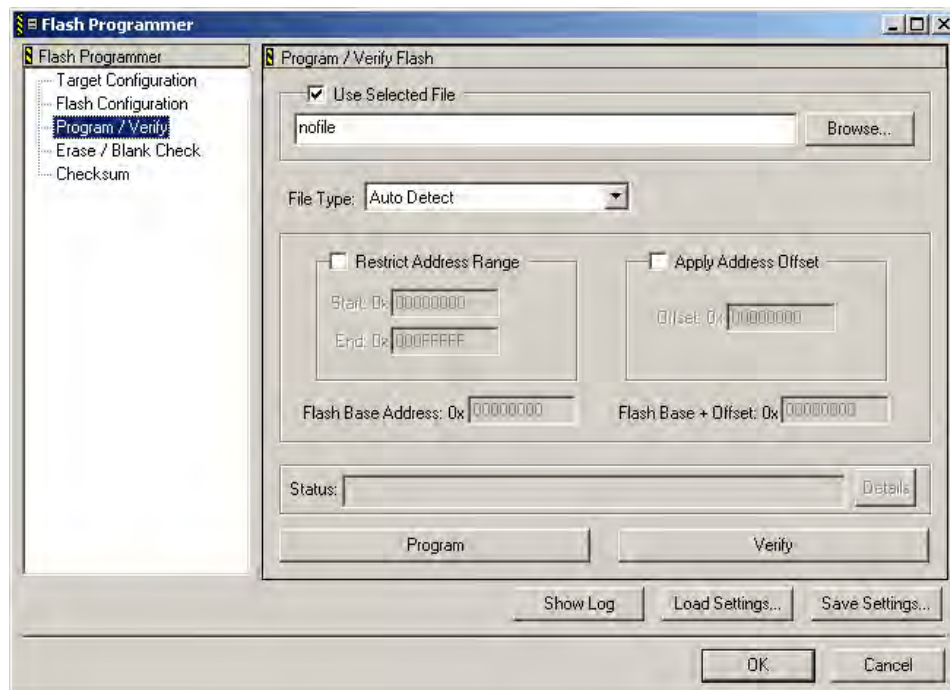


- Back in the *Flash Programmer* main window click on *Erase/Blank Check*.
- Make sure both check boxes are enabled (as shown below) and click on the *Erase* button to erase the entire on-chip Flash memory.



- The Flash erase procedure will take up to 10 seconds. The *Status:* line shows the individual steps that are performed during the Flash erasure.
- If you wish you can do a *Blank Check* at the end of the Flash erase sequence to verify the Flash sectors are blank. Simply click on the applicable button.

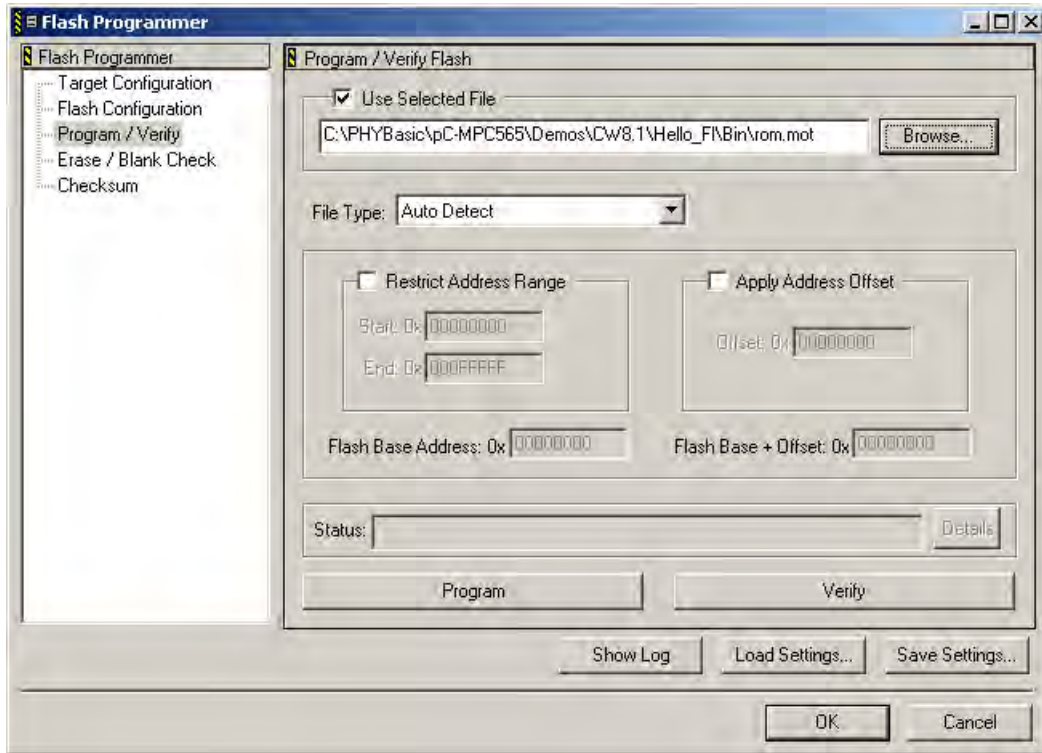
- Now go to the on *Program/Verify* section of the *Flash Programmer* window.



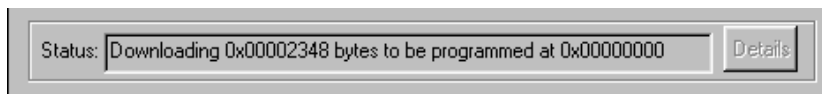
- Select the output file that will be downloaded into the on-chip Flash. Click on the *Browse...* button in the upper right-hand corner of this window.
- Navigate to the *PHYBasic\pc-MPC565\Demos\CW8.1\Hello_FL\Bin* folder in the *Select File to Program* window.



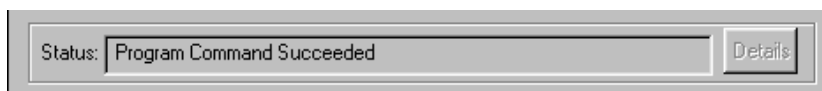
- Select the *rom.mot* file from the list of available files and *Open* it. The *Flash Programmer* window will reappear and should look like this:



- 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:



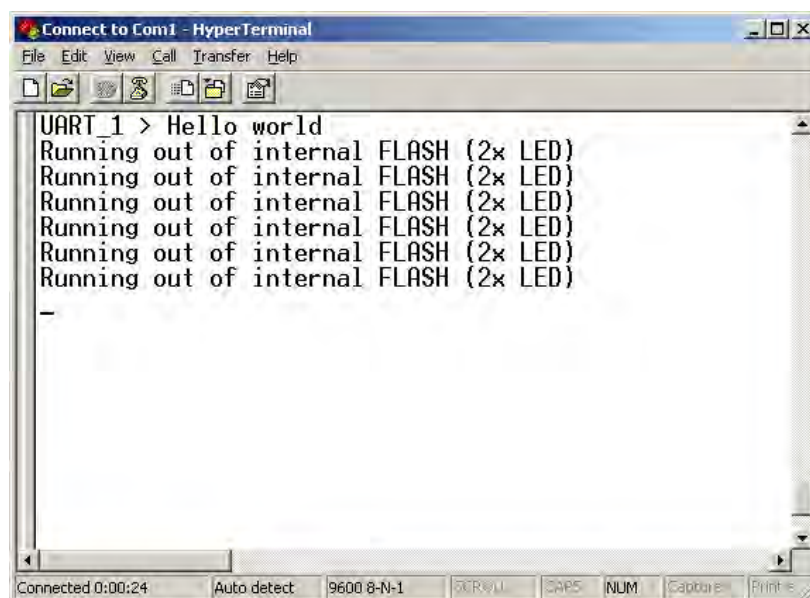
- If you wish to verify the Flash download simply click on the *Verify* button to start this procedure.
- Click on the OK button to exit the *Flash Programmer* tool.
- Click *No* when you are prompted to save these new settings.

You have now successfully downloaded the pre-existing *Hello_Fl* example program to the MPC565 on-chip Flash.

- Disconnect the power adapter from Development Board.
- Disconnect the DB-25 parallel cable.
- Reconnect the power adapter to the Development Board.
- Start the program execution by simultaneously pressing the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board, first releasing the Reset and then, two or three seconds later, release the Boot button.

The red LED D6 on the Development Board will now flash in a different pattern with two short on/off sequences followed by a longer pause.

- 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 (2x LED)"**



```
Connect to Com1 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
UART_1 > Hello world
Running out of internal FLASH (2x LED)
Running out of internal FLASH (2x LED)
Running out of internal FLASH (2x LED)
Running out of internal FLASH (2x LED)
Running out of internal FLASH (2x LED)
Running out of internal FLASH (2x LED)
-
Connected 0:00:24 Auto detect 9600 8-N-1 CTRL CAPS NUM Capture Print
```

2.5.4 Downloading Hello_Fl to the External Standard Flash (PCM-019-1010E Only)

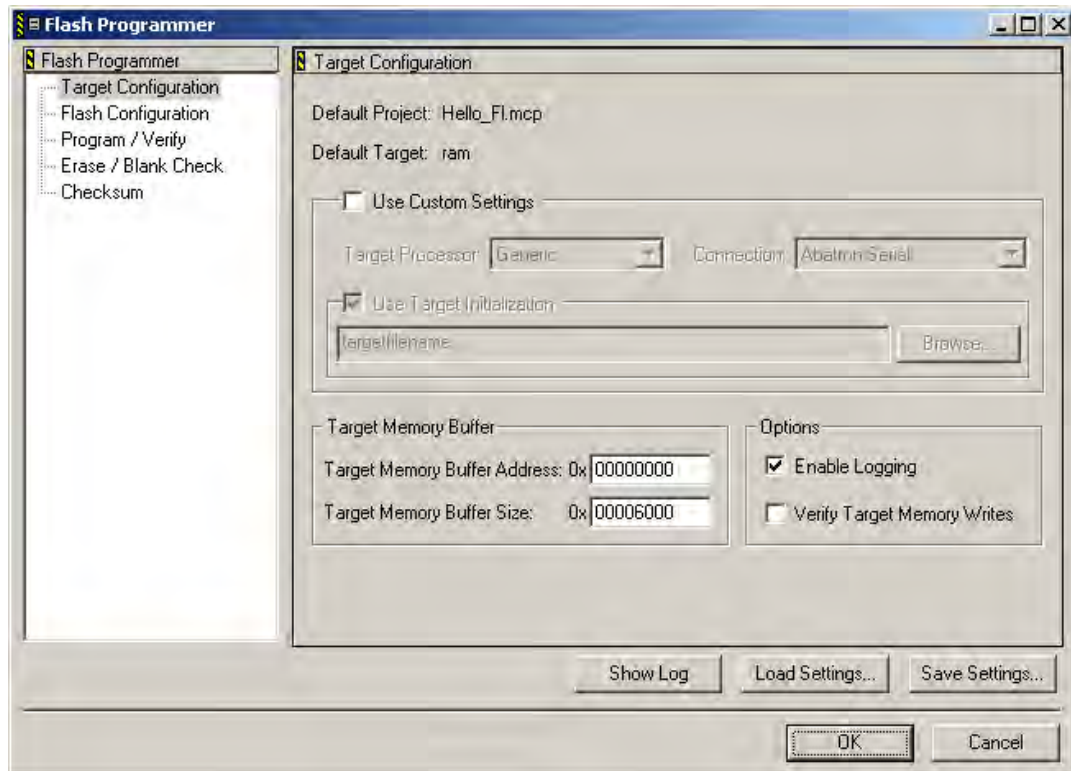
This example demonstrates burning code to the external standard Flash on the phyCORE-MPC565. The example program will be downloaded to the standard Flash devices at U5/U6 (AM29LV type).

Note:

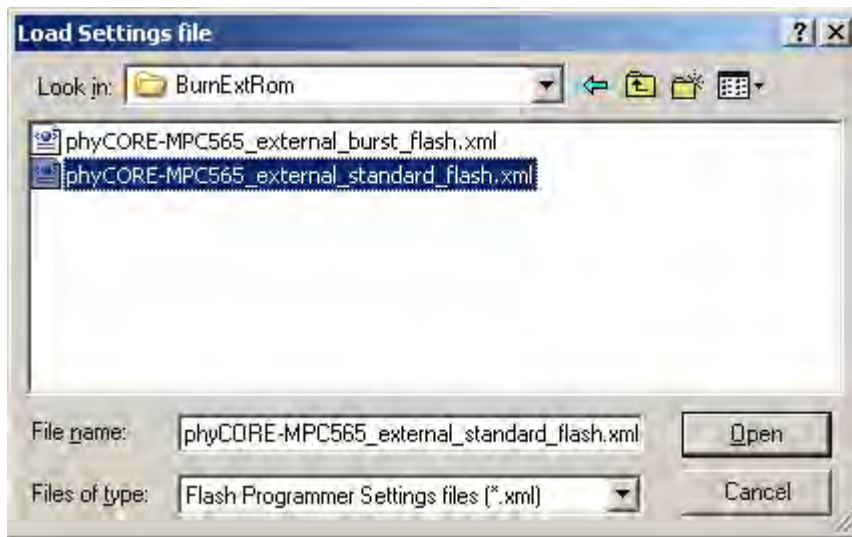
This example only applies to phyCORE-MPC565 modules with part number PCM-019-1010E and the latest EPLD firmware. For modules sold prior to 2/10/05 please see *Application Note LAN-033e for instructions on updating the EPLD firmware.*

- Disconnect the power adapter from Development Board.
- Reconnect the DB-25 parallel cable.
- Change Jumper JP17 to position 2+3 (*refer to Figure 3*).
- 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.

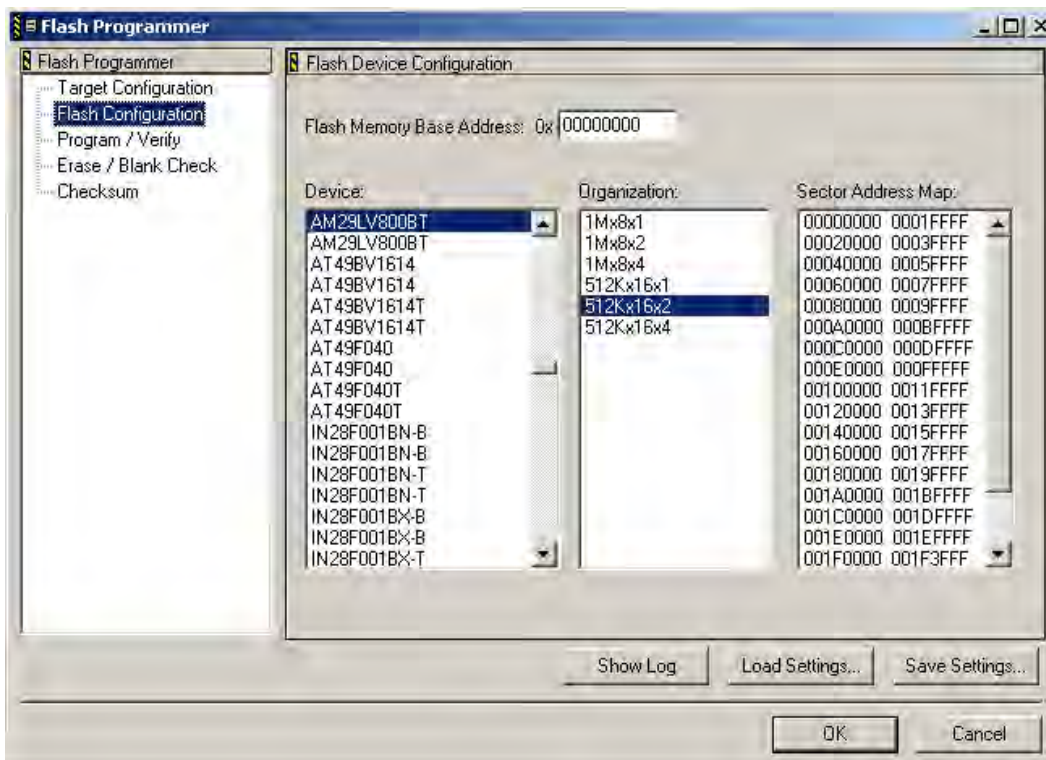
- The following window will appear:



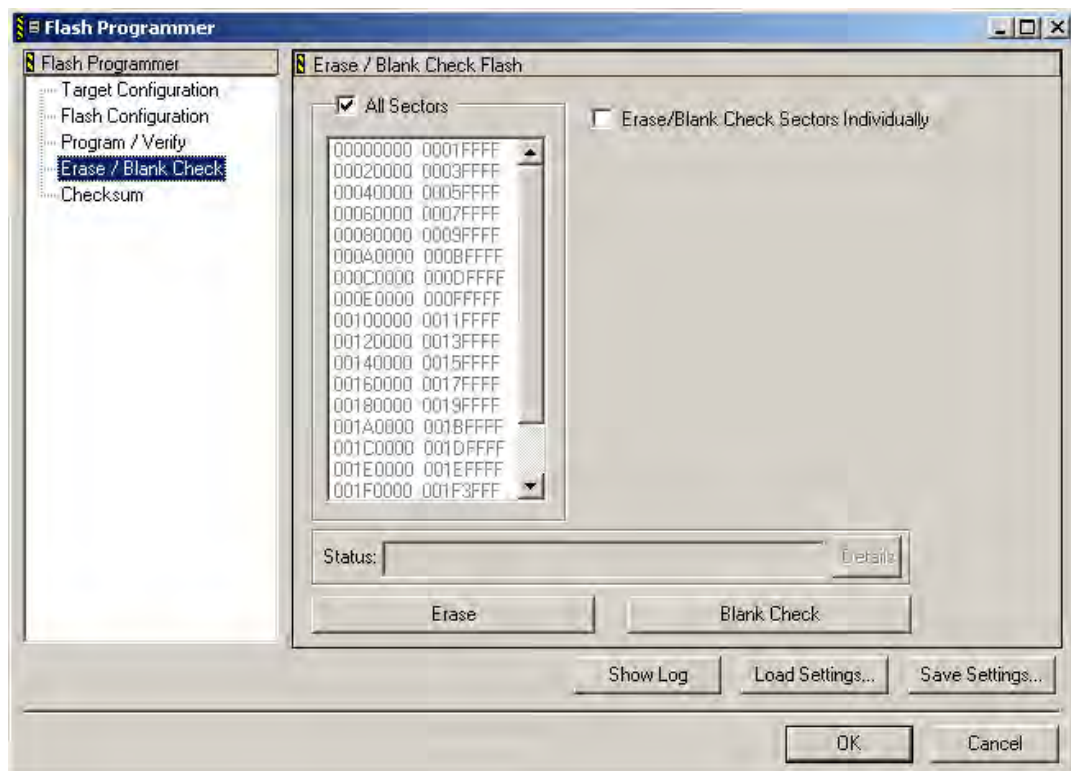
- Click on the *Load Settings...* button
- Browse to the *phyCORE-MPC565_external_standard_flash.xml* file located in the *PHYBasic\pc-MPC565\Demos\CW8.1\Hello_FL\BurnExtRom* folder and open the file.



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

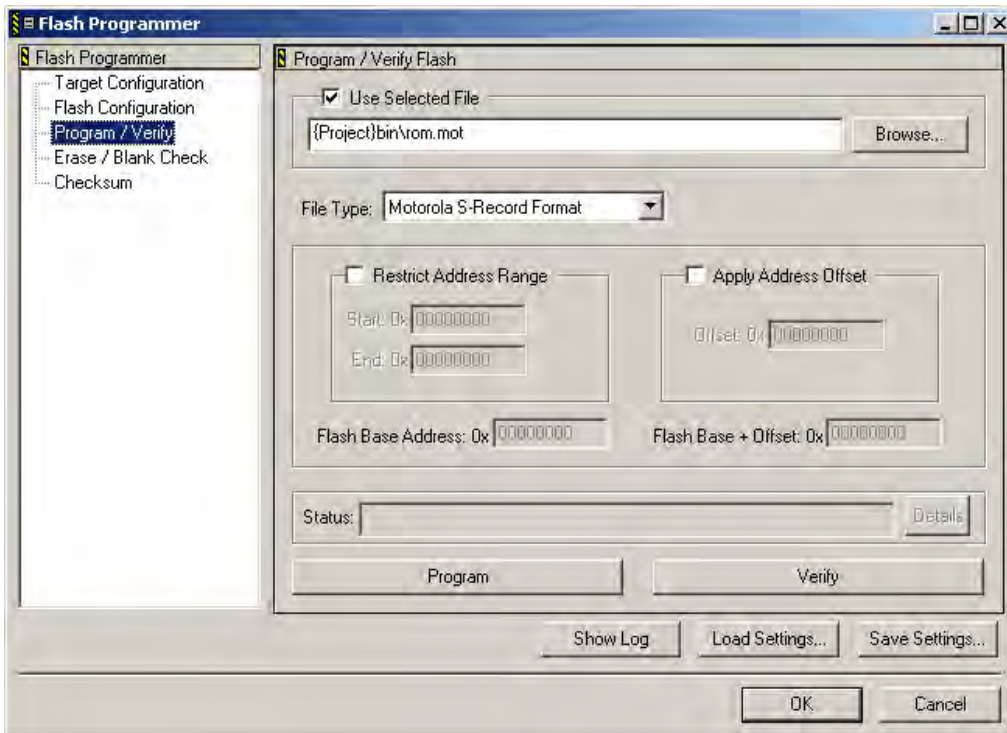


- Click on *Erase / Blank Check*.



- The Flash erase procedure will take up to 10 seconds. The *Status:* line shows the individual steps that are performed during the Flash erasure.
- If you wish you can do a *Blank Check* at the end of the Flash erase sequence to verify the Flash sectors are blank. Simply click on the applicable button.

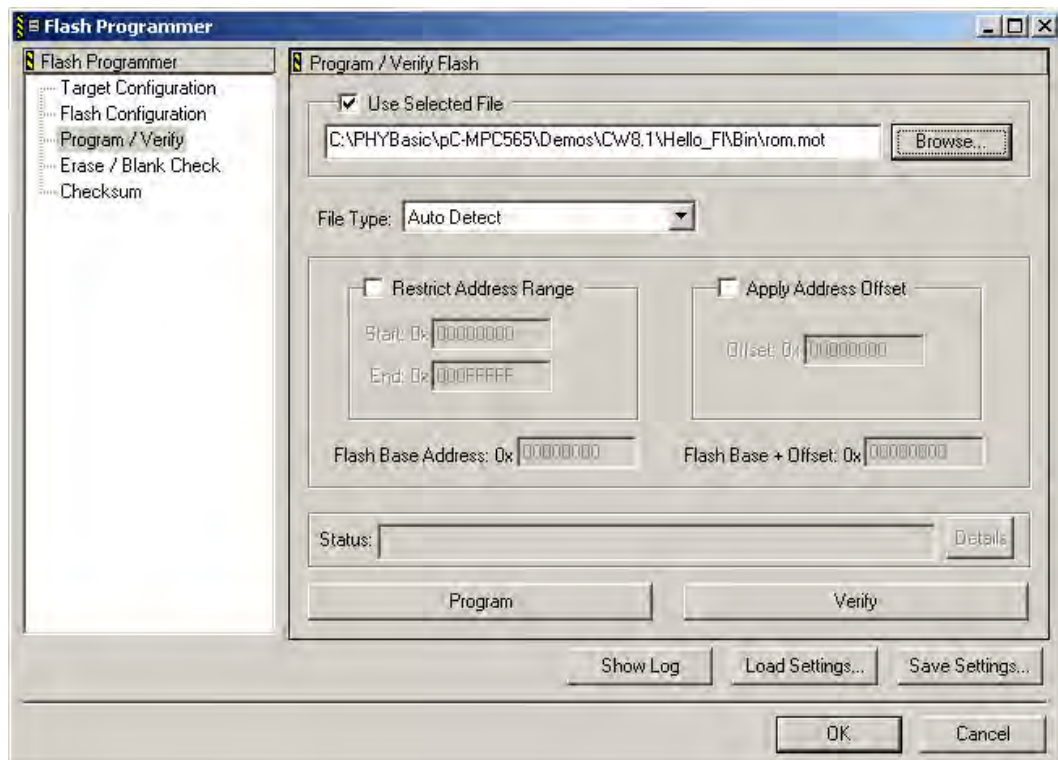
- Now go to the *Program/Verify* section of the *Flash Programmer* window.



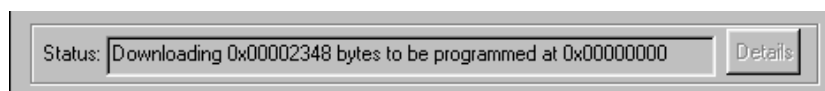
- Select the output file that will be downloaded into the external standard Flash. Click on the *Browse...* button in the upper right-hand corner of this window.
- Navigate to the ***PHYBasic\pc-MPC565\Demos\CW8.1\Hello_Fl\Bin*** folder in the *Select File to Program* window.



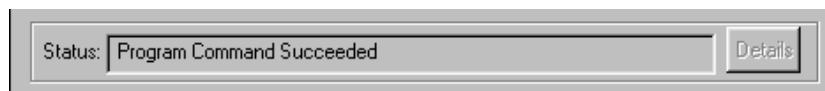
- Select the **rom.mot** file from the list of available files and *Open* it. The *Flash Programmer* window should look as follows:



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

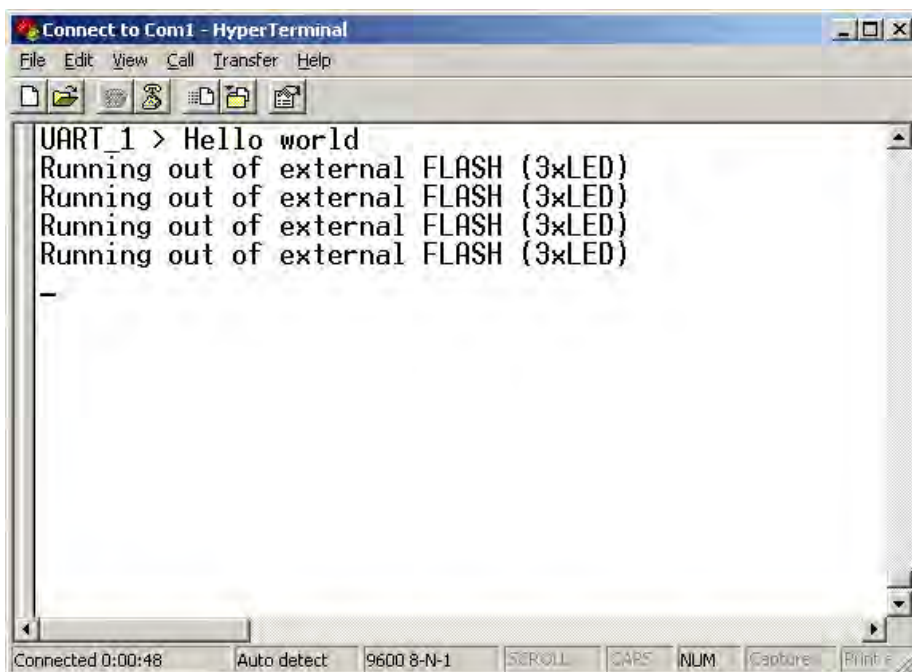
- Click on the OK button to exit the **Flash Programmer** tool.
- Click *No* when you are prompted to save these new settings.

You have now successfully downloaded the **Hello_Fl** example program to the MPC565 external standard Flash.

- Disconnect the power adapter from Development Board.
- Disconnect the DB-25 parallel cable.
- Reconnect the power adapter to the Development Board.
- Start the program execution by simultaneously pressing the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board, first releasing the Reset and then, two or three seconds later, release the Boot button.

The red LED D6 on the Development Board will now flash in a different pattern with three short on/off sequences followed by a longer pause.

- 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 (3x LED)"**



2.5.5 Downloading Hello_Fl to the External Burst Flash (PCM-019-3230 Only)

This example demonstrates burning code to the external burst Flash on the phyCORE-MPC565. The example program will be downloaded to the burst Flash devices at U3 (AM29BDD type).

Note:

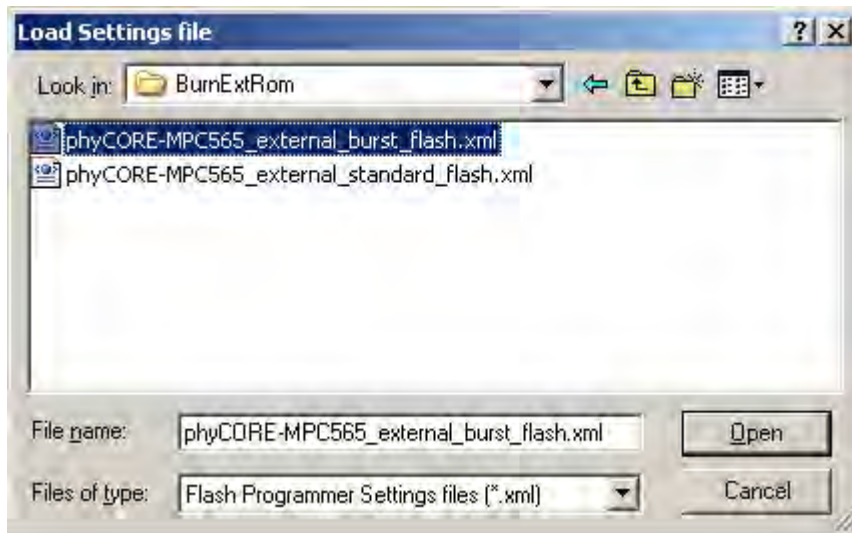
This example only applies to phyCORE-MPC565 modules with part number PCM-019-3230.

- Make sure Jumper JP17 on the Development Board is closed at position 2+3 (refer to Figure 3).
- Open the **Flash Programmer** window by selecting *Tools / FlashProgrammer* from the pull-down menu. The following window will appear:

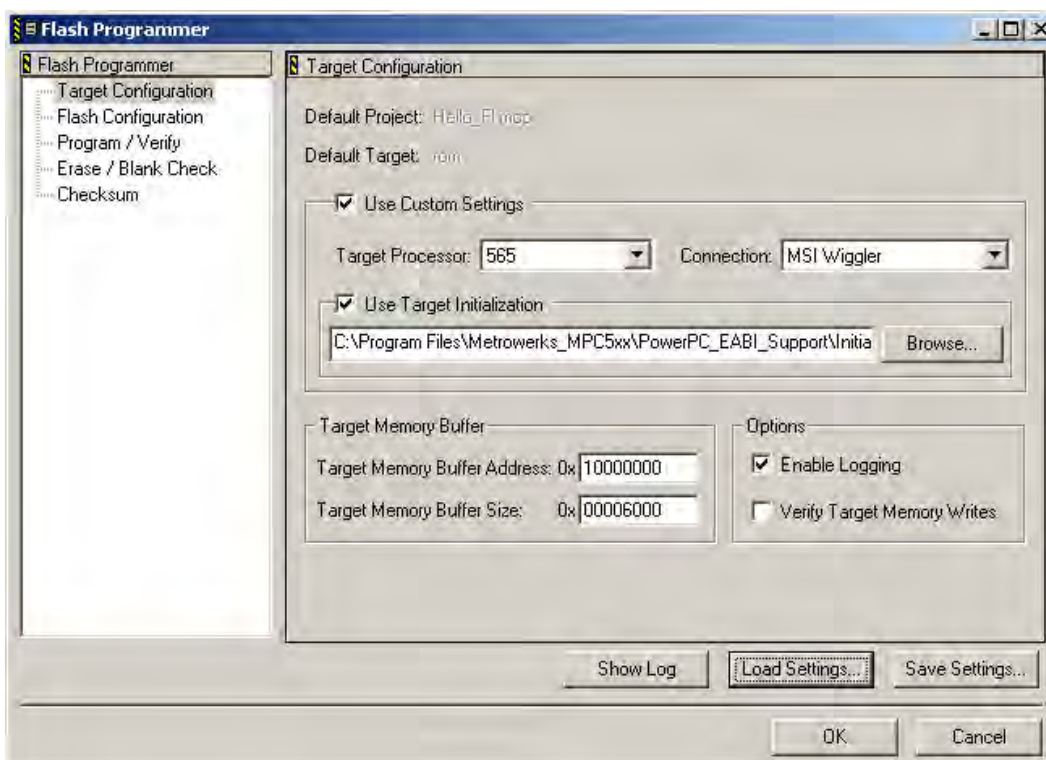


- Click on the *Load Settings...* button.

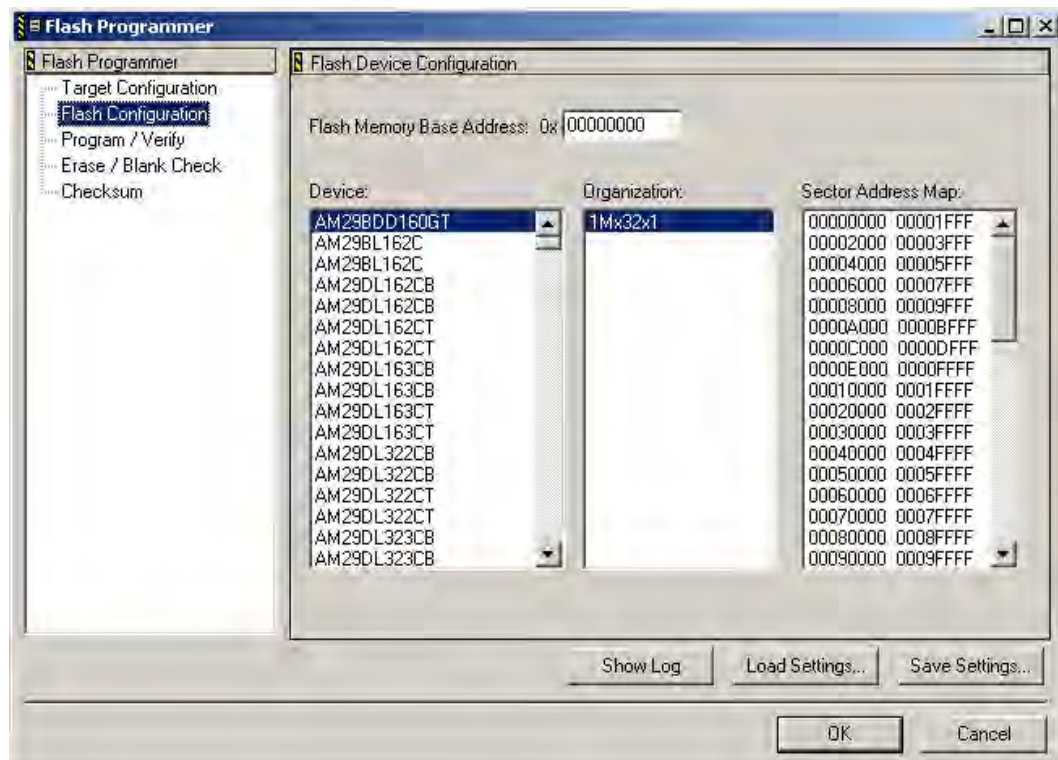
- Browse to the *phyCORE-MPC565_external_burst_flash.xml* file located in the *PHYBasic\pc-MPC565\Demos\CW8.1\Hello_Fl\BurnExtRom* folder and open the file.



- After loading these settings, the *Target Configuration* window should look as follows:

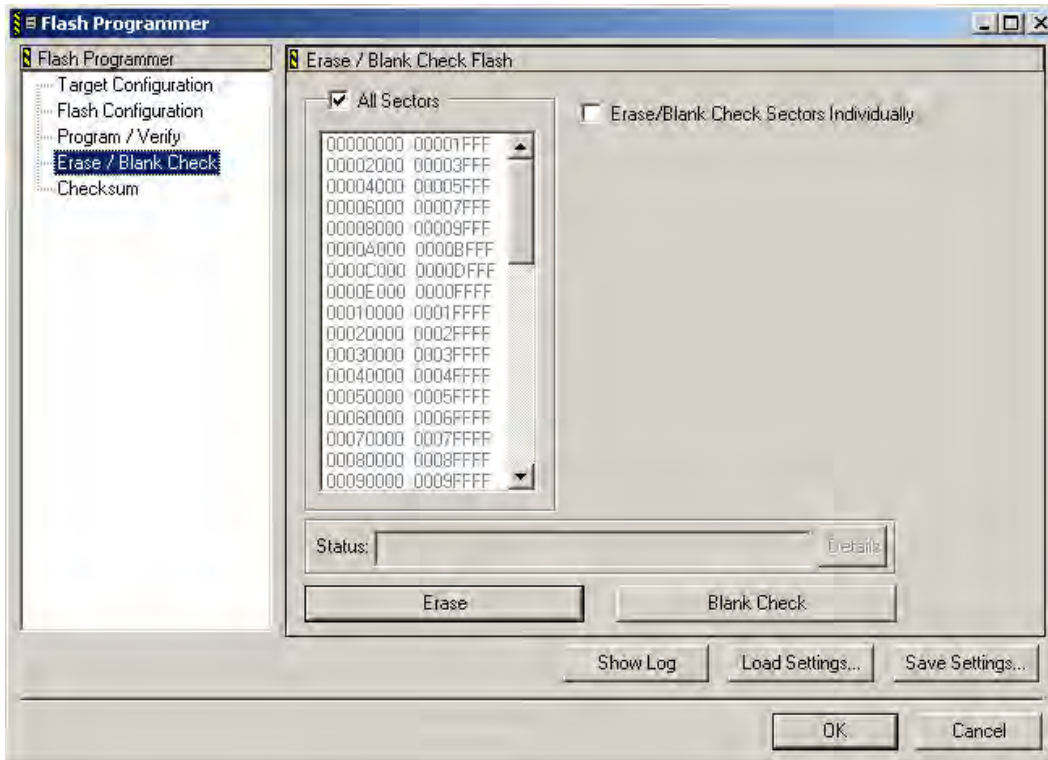


- Click on *Flash Configuration* to check that the correct Flash device is as follows:



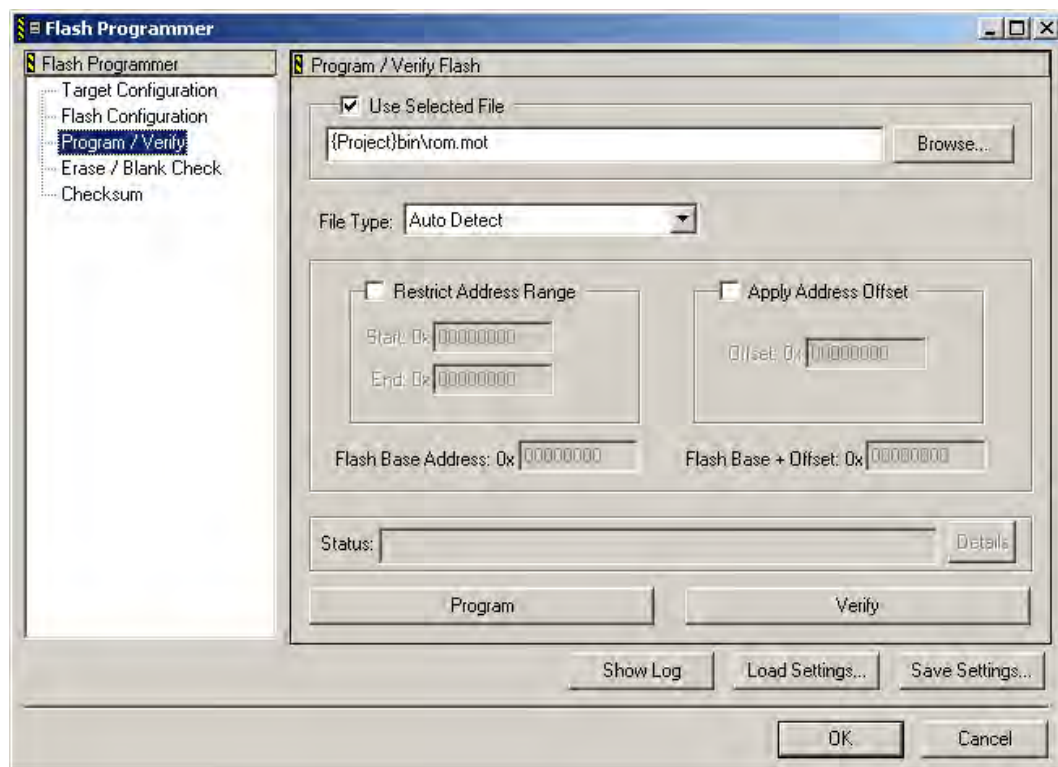
- Select the *Erase/Blank Check* from the *Flash Programmer* window.

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

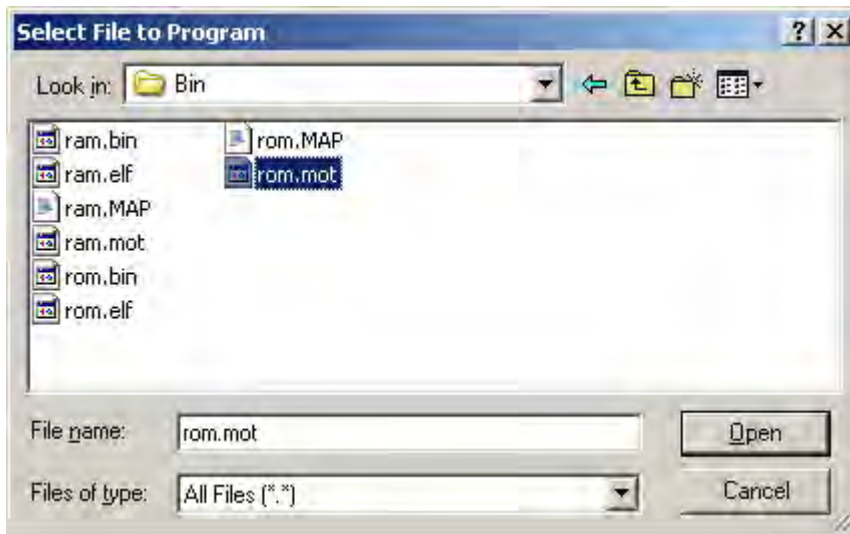


- The Flash erase procedure will take up to 10 seconds. The *Status:* line shows the individual steps that are performed during the Flash erasure.
- If you wish you can do a *Blank Check* at the end of the Flash erase sequence to verify the Flash sectors are blank. Simply click on the applicable button.

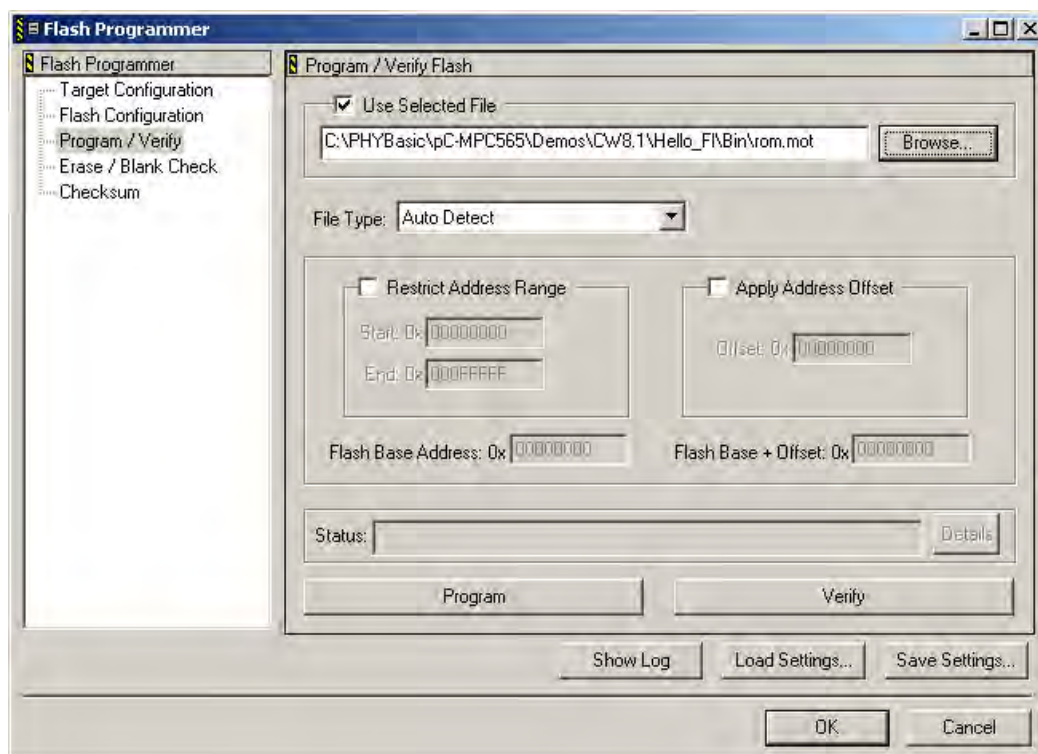
- Now go to the *Program/Verify* section of the **Flash Programmer** window.



- Select the output file that will be downloaded into the external burst Flash. Click on the *Browse...* button in the upper right-hand corner of this window.
- Navigate to the **PHYBasic\pc-MPC565\Demos\CW8.1\Hello_FL\Bin** folder in the *Select File to Program* window.

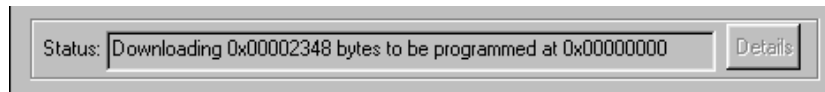


- Select the *rom.mot* file from the list of available files and *Open* it. The *Flash Programmer* window will reappear and should look like this:

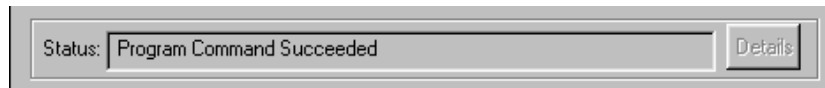


- Now click on the *Program* button to start downloading the *rom.mot* file into the on-chip Flash.

- 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:



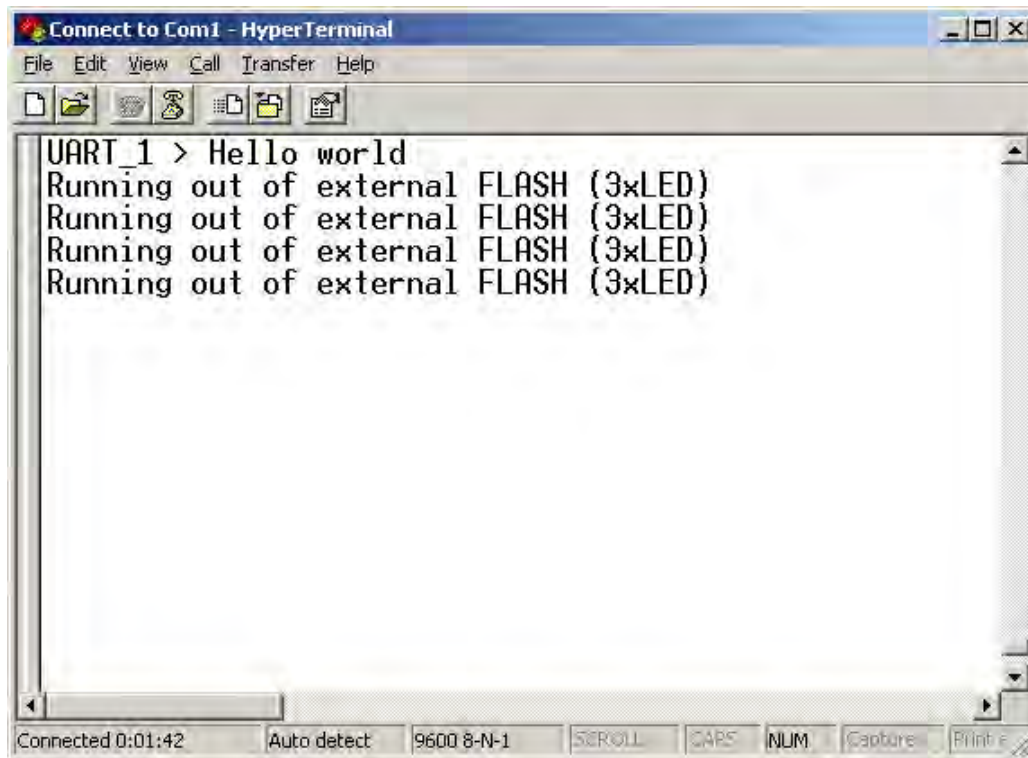
- If you wish to verify the Flash download simply click on the *Verify* button to start this procedure.
- Click on the OK button to exit the *Flash Programmer* tool.
- Click *No* when you are prompted to save these new settings.

You have now successfully downloaded the *Hello_Fl* example program to the external burst Flash on the phyCORE-MPC565.

- Disconnect the power adapter from Development Board.
- Disconnect the DB-25 parallel cable.
- Reconnect the power adapter to the Development Board.
- Start the program execution by simultaneously pressing the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board, first releasing the Reset and then, two or three seconds later, release the Boot button.

The red LED D6 on the Development Board will now flash in a different pattern with three short on/off sequences followed by a longer pause.

- 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 (3x LED)"***



2.6 Creating a New Project

For ease of creating a new project for the phyCORE-MPC565 it is recommended that the given **Hello_FI** project is used as a template.

The **Hello_FI** project already has the target, linker, compiler, debugger, and startup code correctly configured for the phyCORE-MPC565 target.

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

Document: phyCORE-MPC565 QuickStart Instructions
Document number: L-647e_1, March 2005

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) 9221-33

Published by

PHYTEC

© PHYTEC Meßtechnik GmbH 2005

Ordering No. L-647e_1
Printed in Germany