PHYTEC

QuickStart Instructions

PowerPC Kit

phyCORE-MPC5554

Using iSYSTEM winIDEA for PowerPC
Development Tool Chain

Note: The PHYTEC Tool-CD includes the electronic version of the phyCORE-MPC5554 English Hardware Manual

Edition: October 2009

A product of a PHYTEC Technology Holding company

	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: October 2009

Index of Figures

1 Introduction to the phyCORE-MPC5554 Rapid Development Kit

This QuickStart provides:

- general information on the PHYTEC phyCORE-MPC5554 Single Board Computer
- an overview of iSYSTEM AG winIDEA GNU development tool chain
- instructions on how to run example programs on the phyCORE-MPC5554, mounted on the PHYTEC Development Board, in conjunction with iSYSTEM winIDEA tools.

Please refer to the phyCORE-MPC5554 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-MPC5554 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-MPC5554 Hardware Manual
- PHYTEC phyCORE-MPC5554 QuickStart Instructions
- MPC5554 controller User's Manuals and Data Sheets

1.2 Overview of this QuickStart Instruction

This QuickStart Instruction provides a general "Rapid Development Kit" description, as well as software installation hints and example programs enabling quick out-of-the box start-up of the phyCORE-MPC5554 in conjunction with iSYSTEM winIDEA tools. It is structured as follows:

1) The *"Getting Started"* section uses the example program *"Hello"* to demonstrate the download of user code to the phyCORE-MPC5554 external RAM and Flash memory using iSYSTEM' winIDEA Debugger.

1.3 System Requirements

Use of this "QuickStart" requires:

- the PHYTEC phyCORE-MPC5554
- the PHYTEC Development Board with the included DB-9 serial cable and AC adapter supplying 5 VDC/min. 1000 mA
- the iSYSTEM iONE-E Emulator unit (or other iSYSTEM iC3000 / iONE based Emulator unit you own). iONE-E is the hardware interface between a standard PC USB 1.1 port and the standard (2.54 mm) 14-pin MPC55xx JTAG/ONCE/Nexus header connector
- the PHYTEC phyCORE-MPC5554 Tool CD-ROM
- the iSYSTEM winIDEA standard CD-ROM with lot of information about winIDEA products generally
- computer with 800 MHz Pentium compatible microprocessor, 512 MB RAM, CD-ROM drive, USB port, serial, and Ethernet ports, running Windows XP/2000, 350 MB free 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.isystem.com
sales@isystem.com
support@isystem.com

http://www.freescale.com

1.4 The PHYTEC phyCORE-MPC5554

The phyCORE-MPC5554 module integrates the 32-bit Freescale MPC5554 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 memory (2 MByte high speed Flash, 64 kByte SRAM, 32 kByte Cache), the phyCORE-MPC5554 can be populated with 2 MByte to 8 MByte Standatd Flash memory (29LV800..320) and 1 MByte to 16 MByte of Sync. BurstSRAM. The external Flash supports direct on-board programming without additional programming voltages. A serial EEPROM, 4 Kbyte (up to 32 KByte), is available for storing operating parameters. I^2C Real-Time Clock with a calendar and alarm function serves as a real-time reference.

The phyCORE-MPC5554 is especially suited for applications requiring processing of calculation-intensive algorithms in addition for handling of numerous complex and time critical external signals.

The MPC5554's integrated Signal Processing Extension (SPE) provides DSP-like hardware-level execution of elementary operations with single precision floating point numbers (FPU) and supports Multiply and Accumulate operations with its MAC unit. The two Enhanced Time Processing Units (eTPU) with its 32 channels (signals) each provides flexible co-processing to meet hard real-time requirements. In addition of the processors processing units, the SBC integrates a very high density FPGA device, that enables a very flexible way to generate the application specific interface needs. Interfaces like PCI bus, normal address/data bus, GPIO, LVDS ports etc. can be implemented by software. This FPGA can accommodate additional co-processing and periphery units like I2C-Master Controller, 1-Wire-Bus Controller etc.

Other controller features supported by the SBC module include three on-chip CAN 2.0B controllers, two UARTs; SPI Interface, PWM Ports and 40 channel Dual-ADC.

phyCORE-MPC5554 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
- processor: Freescale embedded PowerPC MPC5554 (up to 132 MHz clock)

Internal Features of the MPC5554:

- 32-bit PowerPC core, up to 132 MHz CPU speed
- 32 kByte Cache Memory
- SPE Signal Processing Extention (FPU, MAC Unit)
- MMU Memory Management Unit
- DMA Direct Memroy Access Controller
- Interrupt Latency <70ns @132MHz
- 64 kByte SRAM; 32kByte capable of battery buffering
- 2 MByte Flash (read while write functionality)
- two UART’s (eSCI) LIN support
- four SPI interfaces (DSPI)
- three CAN 2.0B interfaces
- two TPU Time Processing Units with 32 channels (pins) each
- 24 channels (pins) Timer system (eMIOS) for PWM etc.
- dual 12-bit ADC with 40 (65) channels (ext. MUX)
- multi-purpose I/O signals
- JTAG/ONCE/NEXUS/Nexus test/debug port

Memory Configuration[1]:

- 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 MByte to 8 MByte asynchronous standard Flash-EEPROM, 32-bit access
- I^2C Memory: 4 kByte EEPROM (up to 32 kByte, alternatively I^2C FRAM, I^2C SRAM)

Other Board-Level Features:

- UART: two RS-232 transceivers for channel A and B (RxD/TxD); also configurable as TTL
- CAN: two CAN transceivers 82C250-compatible for channels A and B; also configurable as TTL
- Ethernet: 10/100 Mbit/s LAN91C111
- FPGA: Lattice XP FPGA XP6/10/15 or XP20 device for
  IP cores: e.g. I2C-Master, 1-Wire-Master, UART, SPI etc.
  Programmable bus bridge (simple address-/data bus, PCI-Bus, DDR-RAM etc.)
  84 external GPIO with programmable characteristics (TTL, CMOS, differential logic, LVDS etc.)
  Application specific control logic and clock generation (PLL)
  Embedded memory: Single-/Dualport SRAM, FIFO etc.
  In-system programmable over JTAG-Emulation
- I^2C Real-Time Clock with calendar and alarm function
- JTAG/JTAG/ONCE/NEXUS/Nexus test/debug port
- industrial temperature range (-40…+85°C)

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

phyCORE-MPC5554 Development Board

The phyCORE-MPC5554 Development Board is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion and subsequent programming of the phyCORE-MPC5554 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 the Ethernet signals extend to a RJ-45 connector.
Two debugging connectors are provided. A reduced 14-pin JTAG/OnCE and a full 38-pin JTAG/OnCE/Nexus connector.

Development Board Technical Highlights

- Development Board (160 x 100 mm)
- high-density (0.635 pitch) Molex connector for speedy and secure insertion, and subsequent programming, of the phyCORE-MPC5554 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 interfaces
- 10-pin JTAG header to the FPGA device
- 14-pin OnCE/JTAG/Nexus interface
- 38-pin OnCE/JTAG/Nexus interface
- two power LEDs for monitoring the supply voltages
- RESET and IRQ push buttons
- Expansion Bus: all signals routed from implemented phyCORE module to 2 x 200-pin Molex connectors, enabling connectivity to PHYTEC **Add-On hardware**

1.5 The iSYSTEM winIDEA Development Tool Chain

The iSYSTEM winIDEA itself is a highly functional Integrated Development Environment (IDE) tool for developers to create, compile, assemble and link optimized embedded systems code for the PowerPC and many other architectures. The winIDEA works together with included or your own GNU CC package or any common third party Compiler (like i.E. DIAB-Data, Cosmic, Freescale etc.) and other useful tools (like i.E. EasyCode, Tessy, LDRA Suite, National Instruments LabView etc.). There are interfaces via "isystem.connect" to any Windows based application includes your own PC applications, Excel and others. After building the code winIDEA has a powerful debug control center with target download and flash programming possibilities together with the iSYSTEM Emulator hardware. For On-Chip debugging this could be a unlimited iONE, iC3000 universal Base unit or a limited iONE-E connected.

The iSYSTEM winIDEA Integrated Development Environment (IDE) is quite similar to the PC market leader "Microsoft Visual Studio" with extensions and changes for embedded developing and debugging. That IDE is intuitive, easy to use and work in a cohesive manner regardless of compiler or target architecture. This means you can reach more markets and solve more problems faster without invest time and money for learning how new different tools work or work together. iSYSTEM winIDEA IDE graphical interface includes an editor, project manager, class browser, universal command line based compiler interface, source-level and assembly-level debugger.

In addition to keeping all your development tools a mouse-click away, the IDE stores compiler, project, window and debug settings and tracks all the dependencies for your project, simplifying even the most complex development build. The software debugging in winIDEA are state-of-the art and features syntax highlighting of sources and the power to evaluate structures, complex expressions and such things in the debugger. You may correct/change/add source code within the build-in Editor while debugging your application. With iC3000 based systems your are able to perform Multi-Core (i.E. eTPU C-Source

Level) debugging and trace and you can adapt other controllers easy through changing the iCard's.

Now you get a brief overview, please refer the details in the following documents - stored in the winIDEA program directory - after installing winIDEA. Of course they are available as on-line help too.

Please use at first the Demo Workspaces to get familiar with the system! You shouldn't use the "workspace new" key (it would generate a complete blank project without any necessary settings!). Better way is always copying the essential files from one directory to the a new directory. Essential files are the extensions: JRF, QRF, TRD, INI, IND (or whatever for the linker control template). IND and INI Files are simple text format, the other are binaries. You can see/export/import your project settings as XML Files too (this are the "xjrf" and "xqrf" file extensions, that winIDEA write in parallel).

The important documents for you regarding winIDEA are:

- **winIDEA.pdf**
  This is the universal reference and user manual
- **OCD MPC5500.pdf**
  This is the special manual for MPC55xx [**]
- **IONE123_10.pdf**
  This is the description of your iONE [**]

[**] this should be attached to you iONE-E in printed form

iSYSTEM winIDEA consists of the IDE with interfaces to other tools and the Debugger, which are structured as follows:

IDE:

Windows-based Integrated Development Environment housing the following tools in one IDE:

- **Project &Build Manager:** it shows the collection of groups, its files and support files necessary and used to build a target loadable output file, invokes the assembler, compiler and linker, pass error messages to the build-in Editor. It interfaces to external make utilities too with lot of possibilities.
- **Editor:** double-clicking on a file in the Project Manager window opens that file in the Editor.
- **C compiler:** GNU Standard compiler package included (*)
- **Debugger:** Fully active source-level and assembly-level Windows-based debugger for iSYSTEM Debug hardware

(*) You can adapt own GNU or other common compilers. You may edit your sources with your favorite editor outside of winIDEA too - winIDEA then automatically reload the changed sources.

Upon installation of winIDEA - please select "full" installation - , the executables are located in the ***C:\winIDEA\2010*** folder (selectable).

The GNU CC tools are then located at ..\gcc\ppc\. There you will find the GNU Documentation too (..\doc\). All tool commands are easily accessible via intuitive pull-down menus with prompted selections.

The winIDEA handling itself has mostly self-describing user Interface. You have three ways for dealing with it:

- Hot-Keys (re-attachable under "Tools => Customize")
- Icons (Buttons for mouse - simply place mouse over it and wait for function explanation)
- Use "right mouse key" for context sensitive local menus. It makes a difference whether you do it in the middle of a window or at the margin for example, simply try it out!

As depicted on *Figure 1*, winIDEA's editor, Project/Build manager with connected C/C++ compiler and linker and winIDEA debugger with flash programming possibilities comprise the complete tool chain. You have - configurable - windows and window panes for your project, internal editor, disassembly and CPU main registers, global watches and real-time refreshed watches, local variables and context, build manager and search output, memory contents and special-function registers and toolbars for the main functionality.

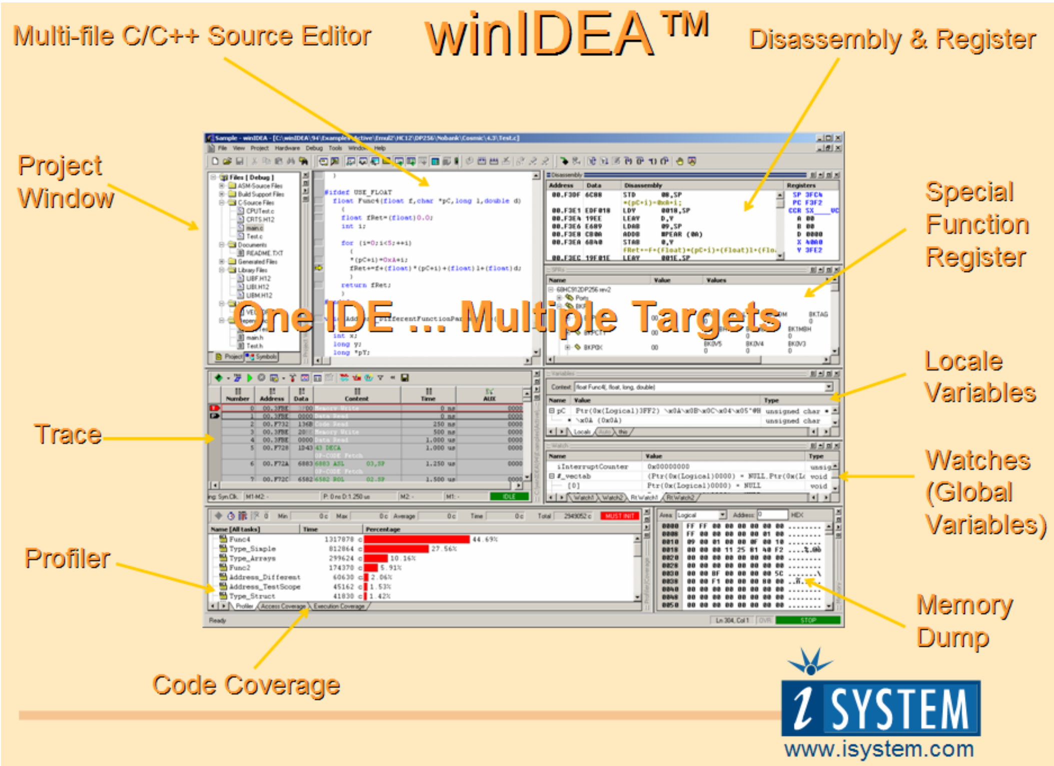


Figure 1: *iSYSTEM winIDEA IDE Architecture*

As depicted on *Figure 1*, winIDEA's editor, Project/Build manager with connected C/C++ compiler and linker and winIDEA debugger with flash programming possibilities comprise the complete tool chain. You have - configurable - windows and window panes for your project, internal editor, disassembly and CPU main registers, global watches and real-time refreshed watches, local variables and context, build manager and search output, memory contents and special-function registers and toolbars for the main functionality.

All IDE commands and functions are accessible via pull-down menus, most of it with short cuts too.

The integrated **Editor** is a powerful tool to compose and later debug the source code. It includes a context-sensitive coloring of keywords for easy navigation and recognition. All colors and custom key sets (for the whole winIDEA of course) 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. The right mouse key at different locations give you different context-sensitive selection possibilities. The search engine enables location of a specified text string within file or within project (scope selectable) and implements find-and-replace operations. At the left selection margin you will find rectangle blocks after download your application code into the target via winIDEA debug hardware. If the compiler has build MCU executable code from that line, there exist the rectangle there. So you easy can control your compiler and source constructs and partly the optimizing of your compiler. But you can do even more. You can select (right mouse key in editor, options…) which function your left mouse key has, when it is above selection margin (Line select, set/reset break or run until). In addition the margin display you Break status (active/ not active) in the source and - with the iSYSTEM Trace tools or Emulators - you get Execution coverage Info's there. Of course it is a multi-file, multi window editor and every window within winIDEA can arranged as "docked" or "MDE" or "Mini Frame". With that we support i. E. a very useful second screen on your windows PC. Of course there is a interface to common Source Control Systems. Anyway: Please refer the manuals for details! If you prefer a other Editor (i.E. "Context" as powerful freeware), winIDEA Editor will load and synchronize automatically the results of your work.

The **Build System** controls the Project dependencies automatically and invokes the different C/C++ compilers, assemblers, linkers, code converters and many other tools for "perfect" build. After download your results to the target winIDEA automatically recognize any of your changes in a project file (source, headers, …) and will ask you whether you want to "make" new version or continue (i. E. if only comment changed) the debugging. If you agree, winIDEA can do all the necessary things (compile, link, download/flash to the target, run until…) automatically WITHOUT closing the debug session. This depends from your settings.

For syntax check, winIDEA provides pre-processing of single source files or for the complete project. It will update dependencies too.

The compiler and linker support ***.elf*** and - via converter or direct - Freescale S-record output file formats. The winIDEA can download and handle all usual Output formats. You need normally neither different files for debug and flashing nor other tools for flashing - it is all in one! For PPC is this the preset (compare the examples and GNU Documentation) ELF Format.

The **WinIDEA Debugger** provides C/C++ (and ADA for some CPU's) 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 usual HLL constructs (i.E. arrays, enums, pointers, structs …) can be displayed in different kinds, monitored and contents and kind of presentation manually modified. There exist different global watches, global real-time watches and local variables display windows including context (= Call-Stack). For internal flash it provides unlimited flash breaks, but be careful, this can worn your number of possible flashing for that Chip (check Freescale manual for information). A Flash-Break flashes up to 3 times the smallest possible flash-sector for every break (worst case; if active, reached and then continued execution).

The iSYSTEM winIDEA iONE-E supports the following debug options of the phyCORE-MPC5554 target.

- There exists no iSYSTEM MCU Simulator for that architecture at all, so you need always your PHYTEC or other target for on-chip debugging

- A hardware-level debugging is provided via a USB 1.1 PC-to-iONE-E and from that to target connection using the MPC5554 JTAG/OnCE port (OnCE: on-chip emulation). The OnCE itself is a kind of hardware debug module integrated on the processor from Freescale.

To the iSYSTEM winIDEA the following debug parts could be connected:

iONE-E This is the small blue box you got with the package. It is a medium level debug tool with following characteristics:
Run up to 90 days AFTER FIRST DOWNLOAD without restriction with any common compiler output. So you are able to try different commercial compilers - mostly 30 day versions are usual - if you want compare it the delivered standard GNU solution results against it.
The Nexus Trace Port feature and Multi-Core-Debugging; i.e. eTPU debugging isn't supported with iONE-E solution.
iONE-E has a USB 1.1 connection and the device is powered over the USB line from the Host-PC.
After that 90 day trial period, it is download code-size limited to 32 Kbyte

It is license-upgradeable for a fee (ask iSYSTEM AG please for current pricing) to a unlimited iONE, but you should think about the faster solutions after that trial-period.

The support for this tool is limited to usual warranty and technical function together with the package. In case of questions beside that please ask for, we will find a agreement.

For the Demo-workspaces PHYTEC is your primary support partner

iONE This is the same as iONE-E, but without any time or download size limit and full iSYSTEM support including free software update for one year (later extendable for a fee).

C3000HS Base Unit This is the latest base unit for unlimited high-speed and high-end debugging. You can connect different Emulators, iCard's for on-Chip Debugging and iTRACE solutions for On-Chip Trace Ports. It contain a powerful Freescale MPC CPU and connects to the PC via very fast (up to ten times faster that iONE's) USB 2.0 and/or Ethernet 100 line. In case of trouble there exists still a serial connection, but no one use/need that today.

The following units can be connected to that iC3000HS base for the 55xx family:

iCard 55xx This small PCMCIA format card fit in the iC3000 and is connected to the JTAG/ONCE Connector like iONE's, but it can debug the eTPU's (multi-core debug possible).

iTRACE PRO This is a second blue box is connected at one side with a interface iCARD to the iC3000 and with a flex cable to a small NEXUS Probe. This probe fit into the NEXUS Port of the PHYTEC board. Than you can do the same as with iCard 55xx, but in addition you get a real-time trace with triggering and filtering for code and data information's, time stamp, profiler and code execution coverage. It have megabytes or in latest version gigabyte of memory for storing the traceport information's.

2 Getting Started

What you will learn with this Getting Started example:

- installing iSYSTEM winIDEA too chain
- installing the Rapid Development Kit software
- interfacing the phyCORE-MPC5554, mounted on the Development Board, to a host-PC
- downloading example user code to the phyCORE-MPC5554 external memory
- programming example user code to the MPC5554 On-Chip Flash memory, external SRAM or external Flash memory

2.1 Installing the iSYSTEM winIDEA Tool Chain

When you insert the iSYSTEM winIDEA CD into the CD-ROM drive of your host-PC, the iSYSTEM WinIDEA CD should automatically launch a setup HTML screen. Than you can install the required software. Otherwise the setup program ***setup*.exe*** can be manually executed from the \install folder of the Standard iSYSTEM WinIDEA CD.

The applicable WinIDEA 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.

- Install the iSYSTEM winIDEA 2010 tool chain for PowerPC from the enclosed iSYSTEM CD, following the steps indicated in the install procedure. We explicitly encourage you to choose winIDEA 2010 even in case you will be prompted to prefer winIDEA 2009!

The default destination location is
C:\WinIDEA\2010
All path and file statements within this QuickStart Instruction are based on the assumption that you accept the default install paths and drives. If you choose different paths and/or drives you must consider this for all further file and path statements. We recommend that you accept the default destination location. For installing you need Administrator rights.

2.2 Installing Rapid Development Kit Software

- Insert the PHYTEC phyCORE-MPC5554 Tool-CD into the CD-ROM drive of your host-PC. The CD should automatically launch a setup program that installs the required demos, documentation and other support documents. Otherwise the setup program ***setup.exe*** can be manually executed from the root folder of the PHYTEC Tool-CD.

The default destination location is ***C:\PHYTEC***. All path and file statements within this QuickStart Instruction are based on the assumption that you choose the default install paths and drives. If you decide to choose different paths and/or drives you must consider this for all further file and path statements. We recommend that you accept the default destination location.

- Follow the instructions in the setup window.

2.3 Interfacing the phyCORE-MPC5554 to a Host-PC

Connecting the phyCORE-MPC5554 (part # PCM-028-xxxx, mounted on the PHYTEC Development Board (part # PCM-979), to your computer is simple:

- If the phyCORE module is not already pre-installed, mount it pins-down onto the Development Board's receptacle footprint (X1) as shown in the *Figure 2* 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.

- Configure the jumpers on the phyCORE Development Board as indicated in *Figure 2*. This correctly routes the CAN interface signals. *Please see the phyCORE-MPC5554 Hardware Manual for further information on jumper settings*.

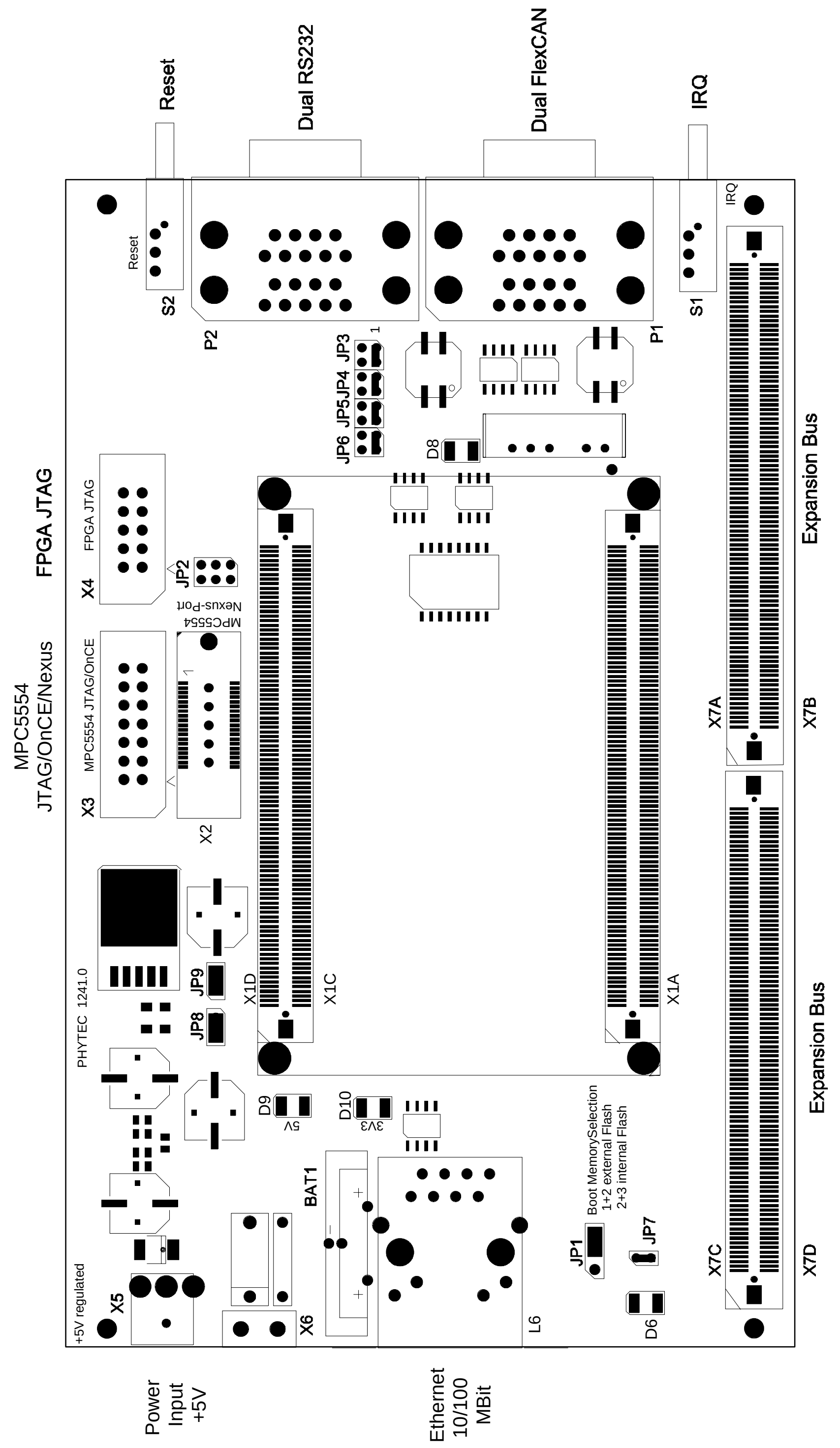


Figure 2: View of Development Board and Jumper Settings

- Connect the included iONE-E to any USB Port of your host-PC. Windows will recognize it and ask for Drivers. Don't try the Windows search, use individual search. If you installed winIDEA first (as recommended), drivers are located in directory "c:\winIDEA\2010\USBDrv\", in addition in the root directory of the winIDEA CD. Of course the drivers are NOT Microsoft verified, but it work and are compatible to any actual updated Windows 2000 and XP version (in case Windows ask for, you should answer "use it anyway" or so). Please note, that you have unfortunately repeat this procedure for every USB port connection of your PC, if you connect iONE-E (or any other iONE, iC3000, etc.) first to that port. This is a Windows request and we cannot change that behavior.
- Connect the iONE-E JTAG/OnCE interface cable to the 14-pin JTAG/OnCE header on the Development Board at X3. This connection is used for the communication between the iONE-E and the phyCORE-MPC5554 target hardware.
- Connect the included serial cable to the lower socket P2A of the double DB-9 connector on the Development Board and to a free serial port of your host-PC. This will enable you to monitor board-host communication via the terminal emulation window included in the winIDA surface or a terminal emulation program, such as Windows HyperTerminal.
- Using the included 5VDC power adapter to connect the power socket X5 on the board (*refer to Figure 3 for the correct polarity*). The phyCORE module/Development Board combination requires a 5 VDC@1A ±5 % regulated supply.

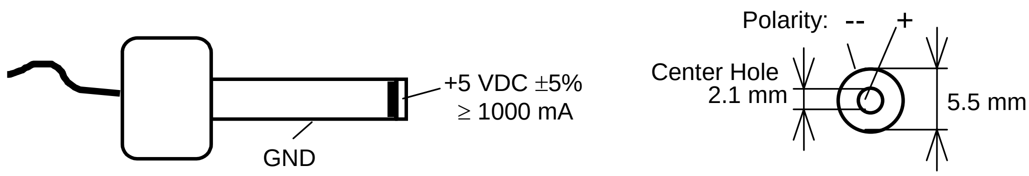


Figure 3: Connecting the Supply Voltage at X5

The LED’s D9 and D10 (two green LED’s) should light, indicating that all voltages are supplied to the phyCORE module. D9 indicates the +5V supplied by the attached power supply and D10 monitors the on-board regulated 3V3.

The phyCORE-MPC5554 should now be properly connected via the Development Board to a host-PC and power supply and you are now ready to use the iSYSTEM winIDEA tool chain to establish communication. This phyCORE module/Development Board combination shall also be referred to as "target hardware".

2.4 Downloading Example Code with winIDEA IDE

The ***Hello*** example sends a program to the target hardware that, when executed, sends a character string from the RDK back to the host-PC. The character string can be viewed with the terminal emulation window in the winIDEA or with a terminal emulation program. The program also controls the user LED D6 (red LED) with equal on and off ratio.

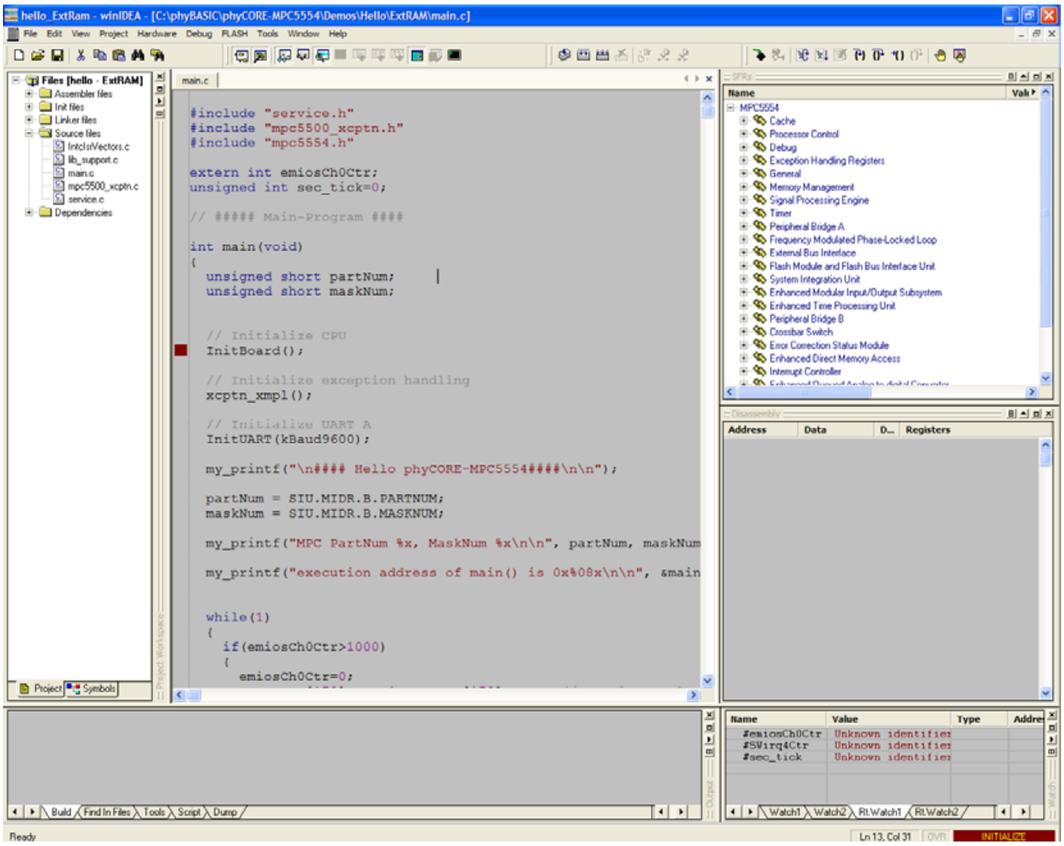
To show how to load and run software, the Hello Demo exists in three different versions:

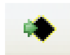
- ***Hello_ExtRAM.jrf***
  Running the Hello Demo Project in external RAM
- ***Hello_IntFLASH.jrf***
  Running the Hello Demo Project in MPC5554 on-chip Flash
- ***Hello_ExtFLASH.jrf***
  Running the Hello Demo Project in external Flash memory

2.4.1 Running the Hello Demo Project in external RAM

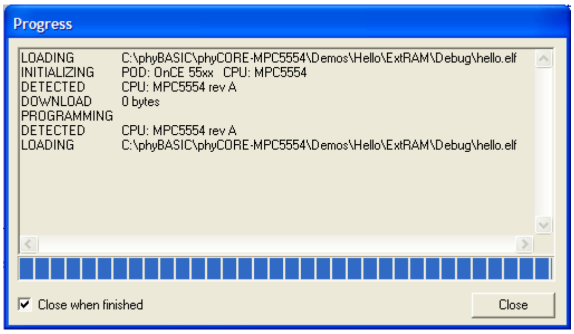
In this section you will download the Hello Demo to the external Sync. Burst Mode SRAM.

- Launch the iSYSTEM *winIDEA* by double-clicking on the *winIDEA* icon or by selecting *winIDEA IDE* from within the *winIDEA* program group
- Open the ***Hello_ExtRAM.jrf*** project file by selecting *File | Workspace* from the *winIDEA* menu bar and navigating to ***C:\PHYTEC\PCM-028 phyCORE-MPC5554\PowerPC-Kit iSystem\Quickstart\Demos\Hello\ExtRAM***.
- The ***Hello_ExtRAM.jrf*** project will open with a predefined window arrangement.

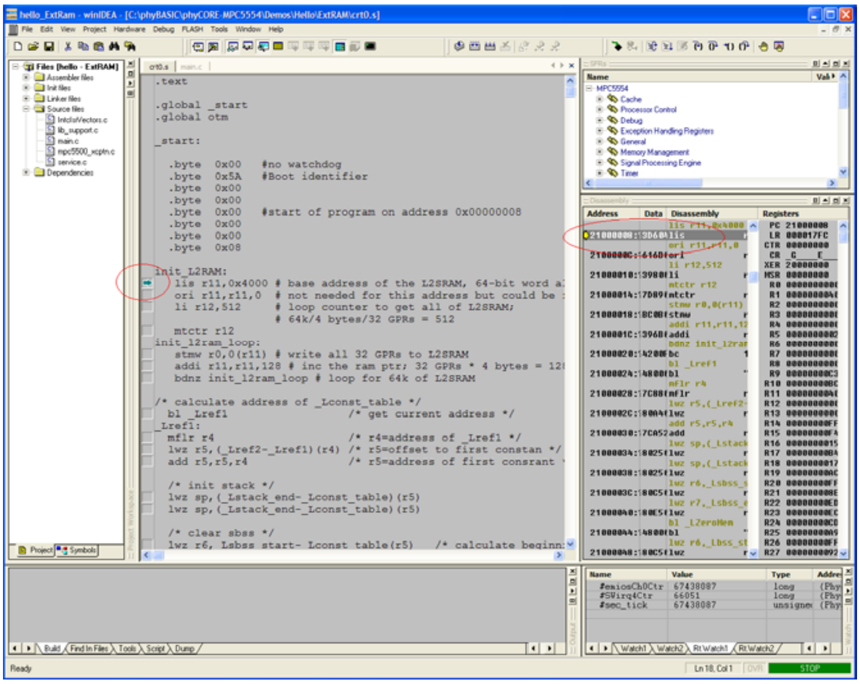


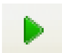
- Alternatively you can open the demo by browsing to the Hello_ExtRAM.jrf file with the Windows Explorer or an other File Manager tool and double-click to invoke the winIDEA with the demo project.
- Perform the build process by clicking the *Make*  icon at the tool bar or select *Project | Make* from the winIDEA pull-down menu (or press F7 key in standard-setting).
- The Make process generates - if no error occurs - an actual output binary (***hello.elf***) in subdirectory /DEBUG/ of the project workspace directory. The "OUTPUT" build/search status window will appear within winIDEA while the project is being compiled and linked. You should/may close it if no error occurs, otherwise you can navigate (right mouse key => "Error" in local menu or simply press function key "F4") to your source file errors. Note that linker errors and any build message appears in the output window pane *Tools*.
- Start the download process by clicking on the download icon  at the tool bar (or press key combination Ctrl+F3). If this will rise an error "Error 40: Communication port not found", then please select *Hardware \ Hardware…* from the *winIDEA* menu bar, navigate to *Communication* and clear content of field *Device*. Reason is, that the blue iONE-box will hard connect to its USB-port. If you change the USB-port later, you will get this error. Clicking on Button *Test* should show a working USB-connection now. Click *close* and *ok* to return to *winIDEA* main window and click the download icon again. It should work now.

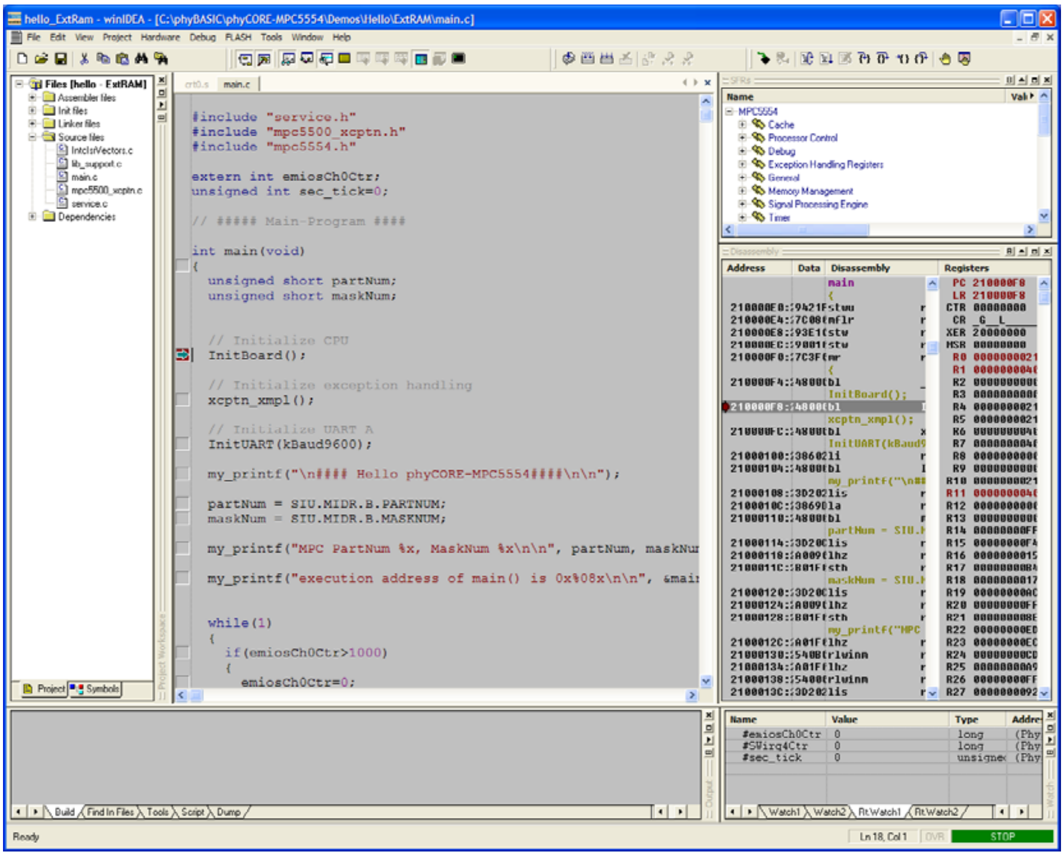
  In this example the download is performed to the external RAM. The appropriated target initialization for the debugger to reach the RAM are in the *.ini file contained in the project folder. The following status window will appear during the download.



- Once download is complete, the source code window will show the actual location of the program counter from where to start the software execution. In the high level language window this location is marked with a green arrow and in the Disassembly Window the code line is highlighted. The high level language window shows at the moment also assembler code due to the entry point of the project is located in a startup file that is written in assembler code. The next step will run the code to the main() function.

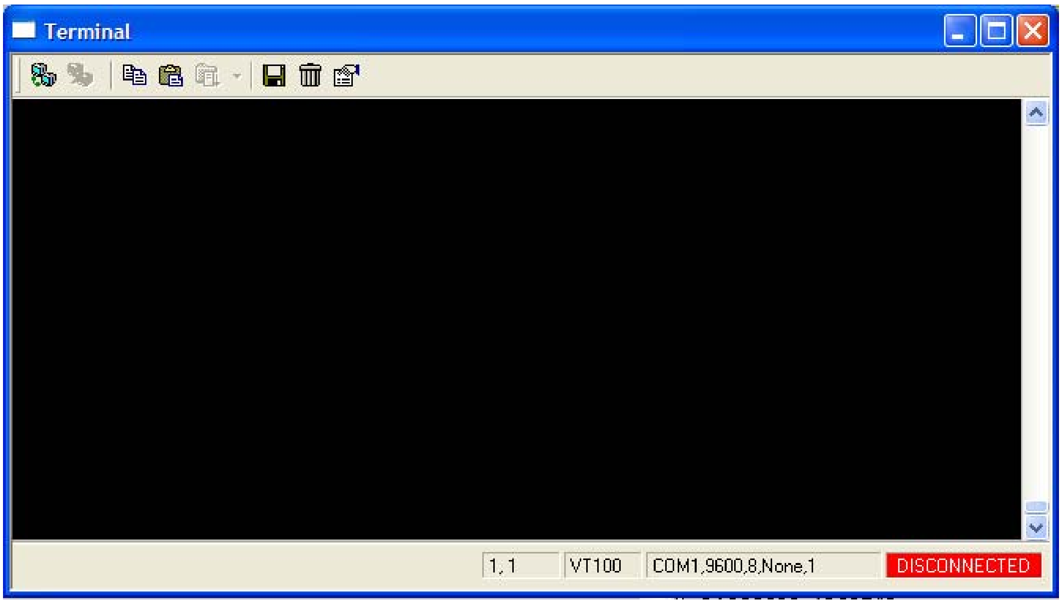


- Run the code by clicking on the  icon or press F5. This will execute the startup code and the processor is running until the first breakpoint is reached.

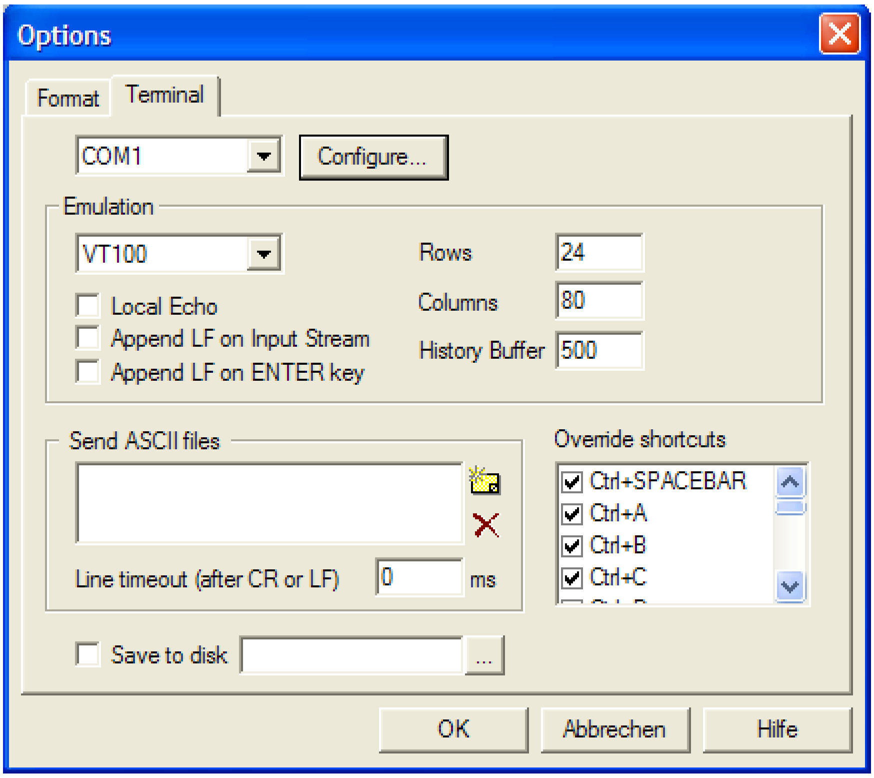


Now the processor has stopped at the breakpoint before the InitBoard() function. Breakpoints can be placed by setting the focus to a code line and pressing F9. The grey shaded area left the code line will then filled brown. Pressing F9 again remove the breakpoint.

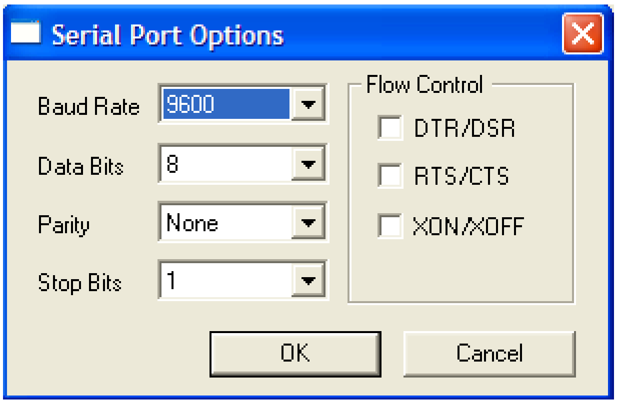
- Before running the complete demo, we need a terminal window to show the output that will send over the serial port A (connector P2A at the development board; refer to section ***2.3***).
  Open and configure the Terminal Window of the winIDEA by selecting *View|Terminal* over the menu bar or click to the  icon.



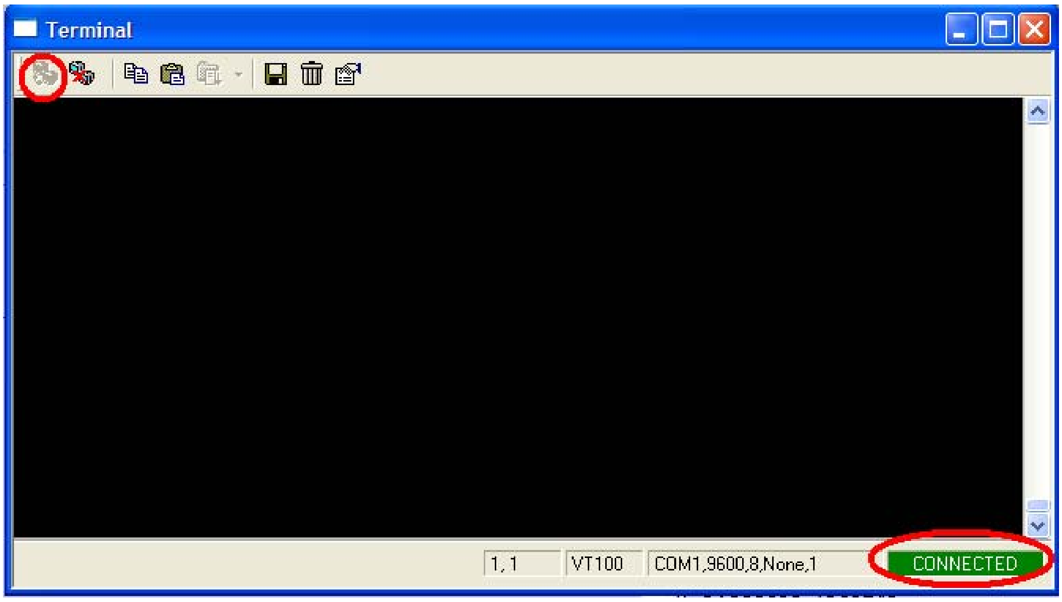
For configuration open the Options Icon  within the terminal window and select the corresponding Host PC serial port number.

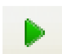


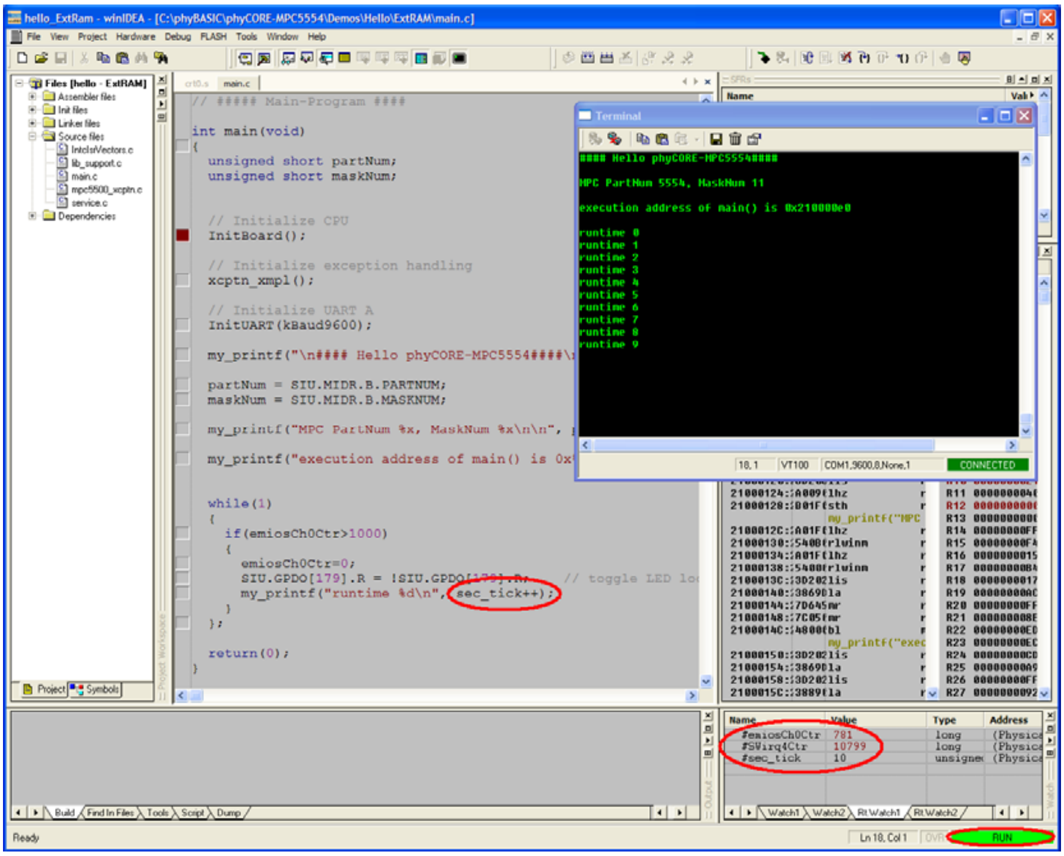
Press the ***Configure*** button adjust the Baud Rate to 9600.



Close the Configure window and Options window by confirming with the OK button.
To establish the serial port communication press the ***Connect*** button  within the Terminal window icon bar.
With a successful opened port the Connect Icon disappears and the status in the right corner changes to ***CONNECTED***.



- Run the code by clicking on the  icon or press F5. This will re-start the code execution beginning with the InitBoard() function with full processor speed. The status of the processor is shown in the lower right corner of the winIDEA window. It is now changed from ***STOP*** to ***RUN***.



In the Terminal window you can see some status messages and the output of a second counter. Also see the User LED D6 located on the baseboard between the Ethernet plug and the Expansion Bus Connector. The LED changes once a second.

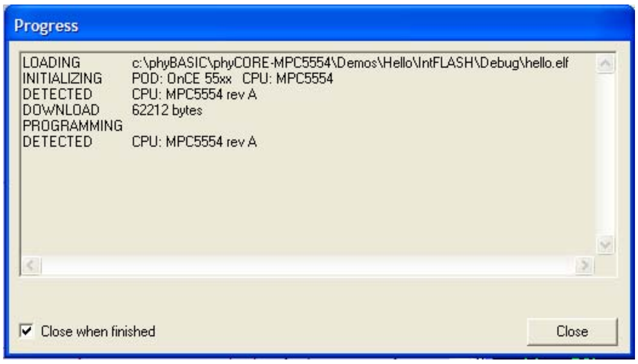
The watch window shows some pre-defined Real-time Watches. The debugger communicates via the JTAG/OnCE module of the Processor and reads the content of these variables in back ground without stopping the processor.

See the ***sec_tick*** variable. This variable is incremented in the demo once a second dependent on the ***emiosCh0Ctr*** counter variable that is incremented by an interrupt service routine once a millisecond.

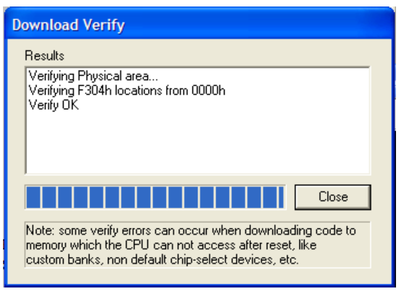
2.4.2 Running the Hello Demo Project in MPC5554 On-Chip Flash

In this section you will download and program the Hello Demo to the MPC5554 on-chip Flash memory.

- Make sure Jumper JP1 on the phyCORE-MPC5554 Development Board is closed at position 2+3 (*refer to Figure 2*).
- Open the ***Hello_IntFLASH.jrf*** and build the output file ***hello.elf*** as described in section ***2.4.1***.
- To program the ***hello.elf*** to the internal processor Flash perform a normal download by selecting the icon or press Ctrl+F3.

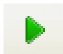


winIDEA automatically first erase and then burns the binary to the internal CPU Flash during download process.
Verify is performed on-the-fly while flashing, but you may use a extra verify procedure by selecting ***Debug | Verify Download*** from the main menu bar. This feature of winIDEA compares the Download File with its target address area in the memory.



It is independent of the memory type and will work for ROM or RAM.

You have now successfully downloaded and burned the ***Hello*** example program to the MPC5554 on-chip Flash memory.
The next steps are to start the ***Hello*** application with the debugger or use it stand-alone.

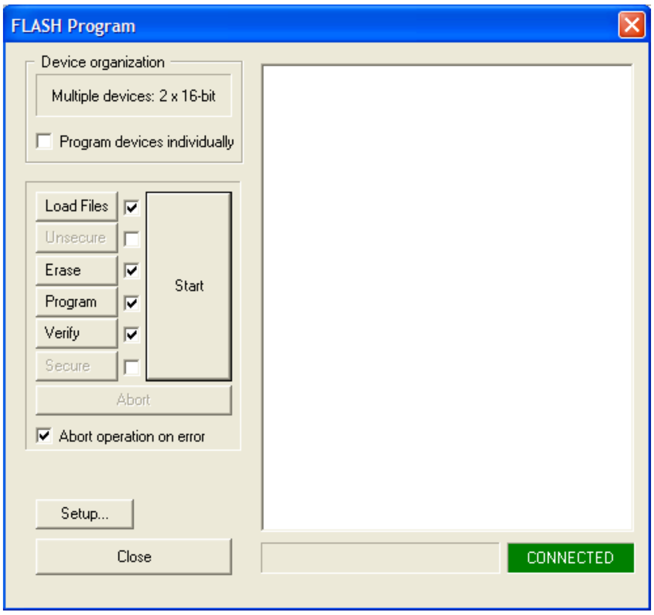
- Running the Hello with the Debugger
  After download/burn procedure the processor is stopped at the first code instruction. Prior executing the hello demo code the processor runs its internal BAM (Boot Assist Module) code. This code starts at address 0xFFFFFFFC, pre-initializes the processor and reads the first 32-bit word (see crt0.s file) from flash address 0x00000000. This 32-bit word code contains a boot identifier and the start address (0x00000008) of the hello demo. After finishing the BAM processor branches to that address and run the hello demo code until the first breakpoint. To start the execution click on the  icon or press F5. This is the same procedure as in the previous section.

- Stand-alone running
  Check that jumper JP1 on the baseboard is configured to 2+3 (*refer to Figure 2*) and press the reset button S2 near the double DB9 serial connector. The system will re-start out of the on-chip Flash memory.
  Or re-connect the power supply cable to X5. The target performs automatically an power-on reset and starts the hello application out of the on-chip internal Flash.

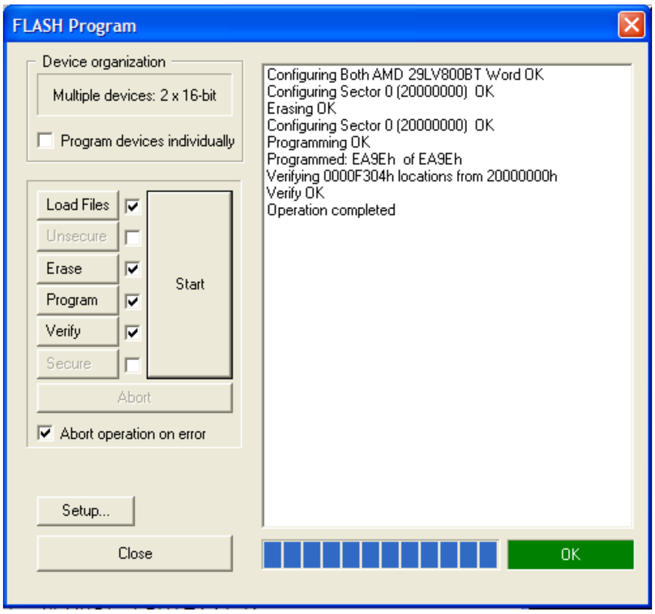
2.4.3 Running the Hello Demo Project in external Flash

In this section you will download program the Hello Demo to external Flash memory.

- Make sure Jumper JP2 on the phyCORE-MPC5554 Development Board is closed at position 1+2 (*refer to Figure 2*).
- Open the ***Hello_ExtFLASH.jrf*** and build the output file ***hello.elf*** as described in section ***2.4.1***.
- To program the ***hello.elf*** to the external Flash open the Flash Programmer window by selecting ***Flash | Program*** from the main menu bar of the winIDEA.



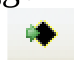
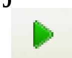
- To perform Flash programming click to the Start button. This will do the selected operations Load Files, Erase, Program and Verify.



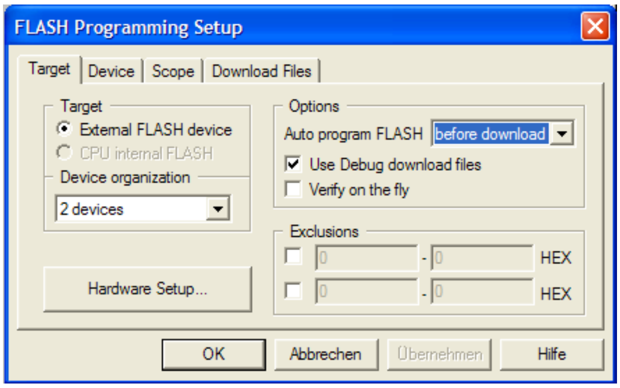
- Close the Flash Program window.

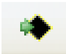
You have now successfully downloaded and burned the ***Hello*** example program to the external Flash memory.
The next steps are to start the ***Hello*** application with the debugger or use it stand-alone.

- Running the Hello with the Debugger

  First click to the Download Icon  to initialize the project for debugging. Then run the code by clicking on the Run Icon  or press F5. This will execute the startup code (crt0.s) and the processor is running until the first breakpoint is reached near the InitBoard() function in main.c. Press again F5 to go ahead with code execution. Now you can see the output in the terminal window and the blinking LED as in the previous sections.

  For development process it is more easy to invoke the flash programming prior the debugger Initialize/Download procedure automatically. Open ***Flash | Setup*** from the main menu bar.



Change under ***Option - Auto program FLASH*** from ***never*** to ***before download*** and confirm with ***OK***. Now try again the download function by clicking Download Icon  or pressing F5.

- Stand-alone running
  Check that jumper JP1 on the baseboard is configured to 1+2 (*refer to Figure 2*) and press the reset button S2 near the double DB9 serial connector. The system will re-start out of the external Flash memory.
  Or re-connect the power supply cable to X5. The target performs automatically an power-on reset and starts the hello application out of the external Flash

Document:	**phyCORE-MPC5554 QuickStart Instructions**
Document number:	**L-695e_1, October 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