phyCORE-P89C51RD2

QuickStart Instructions

Using PHYTEC FlashToolsOCF for on-chip Flash and the µVision2 Software Evaluation Development Tool Chain

Note: The PHYTEC Spectrum CD includes the electronic version of the phyCORE-P89C51Rx2 English Hardware Manual

Hinweis: Die PHYTEC Spectrum CD beinhaltet die elektronische Version des deutschen phyCORE-P89C51Rx2 Hardware Manuals

Edition: July 2002

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 info@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

6th Edition: July 2002

Table of Figures

1 Introduction to the Rapid Development Kit

This QuickStart provides:

- general information on the PHYTEC phyCORE-P89C51RD2 Single Board Computer (SBC)
- an overview of Keil's µVision2 software evaluation development tool chain, and
- instructions on how to run example programs on the phyCORE-P89C51RD2, mounted on the PHYTEC phyCORE Development Board LD 5V, in conjunction with µVision2 IDE

Please refer to the phyCORE-P89C51Rx2 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-P89C51Rx2 Hardware Manual.

1.1 Rapid Development Kit Documentation

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

- the PHYTEC phyCORE-P89C51Rx2 Hardware Manual and phyCORE Development Board LD 5V Hardware Manual
- controller User's Manuals and Data Sheets
- this QuickStart Instruction with general "Rapid Development Kit" description, software installation hints and three example programs enabling quick out-of-the box start-up of the phyCORE-P89C51RD2 in conjunction with the Keil software development tools

1.2 Overview of this QuickStart Instruction

This QuickStart Instruction gives a general “Rapid Development Kit” description, as well as software installation hints and three example programs enabling quick out-of-the box start-up of the phyCORE-P89C51RD2 in conjunction with the Keil software development tools. It is structured as follows:

1) The “*Getting Started*” section provides the two example programs “Hello” and “Blinky” to demonstrate the download of user code to the on-chip Flash memory using PHYTEC FlashTools for on-chip Flash (OCF).

2) The “*Getting More Involved*” section provides step-by-step instructions on how to modify both examples, create and build new projects and generate and download output files to the phyCORE-P89C51RD2 using the Keil tool chain and FlashTools OCF.

3) The “*Debugging”* section provides a third example program - “Debug” - to demonstrate simple debug functions using the µVision2 simulator environment.

In addition to dedicated data for this Rapid Development Kit, this CD-ROM contains supplemental information on embedded microcontroller design and development.

1.3 System Requirements

Use of this "Rapid Development Kit" requires:

- the phyCORE-P89C51RD2 SBC module
- the phyCORE Development Board LD 5V with included DB-9 serial cable and AC adapter supplying 5 VDC / min. 500 mA
- the PHYTEC Spectrum CD
- an IBM-compatible host-PC (486 or higher running at least Windows95/98)

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

KEIL SOFTWARE

http://www.keil.com
support@keil.com

1.4 The PHYTEC phyCORE-P89C51RD2

The phyCORE-P89C51RD2 represents an affordable, yet highly functional Single Board Computer (SBC) solution in subminiature dimensions (40 x 55 mm). The standard board is populated with a Philips P89C51RD2 controller, featuring on-chip Flash memory and RAM.

All applicable data/address lines and applicable signals extend from the underlying logic devices to standard-width (2.54 mm/ 0.10 in.) pin headers lining the circuit board edges. This enables the phyCORE-P89C51RD2 to be plugged like a “big chip” into target hardware.

The standard memory configuration of the phyCORE-P89C51RD2 features 64 kByte on-chip Flash for code storage, 8 kByte on-chip data RAM and in addition 32 kByte external SRAM. The Philips P89C51Rx2 controller family support both **I**n-**S**ystem **P**rogramming (ISP) and **I**n-**A**pplication **P**rogramming (IAP) that allows direct on-board programming. Three Chip Select signals are available for external I/O connectivity.

The module communicates by means of an RS-232 transceiver, operates within a standard industrial range of 0 to +70 degrees C and requires only a 250 mA power source.

PHYTEC FlashToolsOCF enable easy on-board download of user programs into the on-chip Flash of the microcontroller.

phyCORE-P89C51RD2 Technical Highlights

- SBC in subminiature dimensions (40 x 55 mm) achieved through advanced SMD technology
- populated with a 44-pin packaged (PLCC) Philips 8051-compatible P89C51RD2 controller featuring on-chip Flash and RAM
- 6 clocks per machine cycle operation up to 20 MHz
- 64 kByte on-chip ISP/IAP Flash
- 8 kByte on-chip RAM
- 32 kByte external SRAM
- RS-232 serial interface
- 5-channel on-chip Programmable Counter Array (PCA)
- on-board I²C Real-Time Clock, serial EEPROM (up to 32 kByte) and Watchdog IC
- three Chip Select signals for connection to external peripherals
- requires only a +5 V/ 250 mA power source
- operates in a temperature range of 0...70°C (optional -40... 85°C temperature range available)

The phyCORE Development Board LD 5V, in EURO-card dimensions (160 x 100 mm), is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion, and subsequent programming, of PHYTEC phyCORE series Single Board Computers with standard width (2.54 mm/ 0.10 in.) pin header connectors. Simple jumper configuration readies the Development Board's connection to any phyCORE module (standard header pins), which plug pins-down into the contact strips mounted on the phyCORE Development Board LD 5V.

phyCORE Development Board LD 5V Technical Highlights

- Reset signal controlled by push button or RS-232 control line CTS0
- Boot signal controlled by push button or RS-232 control line DSR0
- low voltage socket for supply with regulated input voltage 5 VDC
- additional supply voltage 3.3 VDC
- two DB-9 connectors (P1A, P1B) configurable as RS-232 interfaces
- two additional DB-9 connectors (P2A, P2B) configurable as CAN interfaces, connector P2B optionally configurable as RS-485 interface
- simple jumper configuration allowing use of the phyCORE Development Board LD 5V with various PHYTEC phyCORE SBC's
- one control LED D3 for quick testing of user software
- 2 x 160-pin Molex connector (X2) enabling easy connectivity to expansion boards (e.g. PHYTEC GPIO Expansion Board)

1.5 The Keil Software Evaluation Development Tool Chain

The Keil software evaluation development tool chain fully supports the entire 8051 and 8051-derivative microcontroller family, including the Philips P89C51Rx2 family. This includes a C compiler, macroassembler, linker/locator and the simulator and target monitor within the µVision2 IDE.

The Keil tool chain consists of the following executables:

- **C Compiler** c51.exe
- **Assembler** a51.exe
- **Linker** bl51.exe
- **Converter** oh51.exe
- **µVision2** Uv2.exe (a Windows-based application)

Once installed, the default destination location for the DOS based files is the ***C:\Keil\C51\Bin*** directory while µVision2 is in ***C:\Keil\Uv2***. Access to these programs from Windows is accomplished with µVision2. The entire tool set can be run from µVision2 or directly from DOS with batch files. The evaluation version of the Keil tool chain is restricted to a manipulable code size of 2 kByte. In addition the code will automatically be located at 0x4000 in order to prevent unauthorized use of this version for programming of common devices with internal code memory less than or equal to 2 kByte. Other than these restrictions, the evaluation tool chain functions exactly as the full version does, enabling full evaluation of the features and functionality of Keil development tools. The full version has no such restrictions, both are fully ANSI compliant.

Note:
The default CODE memory is always linked to address 4000H when using the evaluation version of the Keil software development tool chain. The evaluation version therefor does not support microcontrollers with less than 16 kByte on-chip Flash (e.g. P89C51RB2).

µVision2 IDE

µVision2 is a Windows-based Graphical User Interface for the C compiler and assembler. All compiler, assembler and linker options are set with simple mouse clicks. µVision2 runs under Windows 95/98/ME/NT and 2000. This Integrated Development Environment (IDE) has been expressly designed with the user in mind and includes a fully functional editor.

All IDE commands and functions are accessible via intuitive pull-down menus with prompted selections. An extensive Help utility is included. External executables can be run from within µVision2, including emulator software.

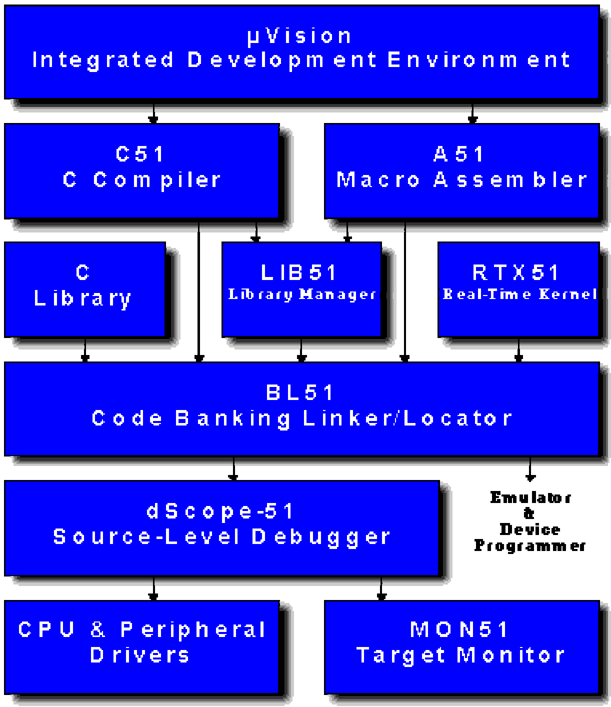


Figure 1: Keil Tool Chain Overview

C51 C Compiler

The C51 ANSI compiler is specifically designed for the 8051 microcontroller family. The compiler can be run either in a DOS box under Windows or directly from µVision2.

A51 Macroassembler

The assembler can be run either in a DOS box under Windows or directly from µVision2.

BL51 Code Banking Linker/Locator

The BL51 linker/locator is used to join re-locatable object modules and library files together and to locate them to fixed memory locations. It supports Code Banking for easy management of multiple code-banks to allow applications with more than 64 kByte of code. The library and object files created with the C51 compiler or the A51 assembler are provided by Keil. This process results in absolute object modules. The BL51 is DOS-based and can be run within a DOS box under Windows or directly from µVision2. A map file (***.m51***) can be produced, giving details of the memory structure. The object file may be specified to contain debugging information as required by simulators, debuggers and emulators. The BL51 also supports an overlay mechanism for variables to optimize the use of the size-restricted internal RAM.

OH51 Object-Hex Converter

The Keil OH51 object-to-hex converter transforms an absolute object file produced by the BL51 Banking Linker into a standard Intel ***.hex*** file. This file is suitable as an input to the PHYTEC FlashTools98 or for programming into an EPROM or an emulator. OH51 can be run in a DOS box under Windows or directly from µVision2.

Debug Environment

µVision2 contains a software simulator supporting debugging either via software on a host-PC or in target hardware. When operated in conjunction with the Keil Monitor resident in target hardware µVision2 enables the following debugging functions:

- run/halt
- set breakpoints
- examine/change memory
- view the stack
- view/set peripheral information
- apply virtual external signals

Note:
The memory model of the phyCORE-P89C51RD2 does not support a Monitor program, hence target hardware debugging is not possible. Application programs can be tested and debugged in the software simulator only.

µVision2 has a performance analysis feature to ensure your code runs efficiently. In addition, µVision2 has a disassembler/assembler that allows the modification of user code without recompiling.

2 Getting Started

What you will learn with this Getting Started example:

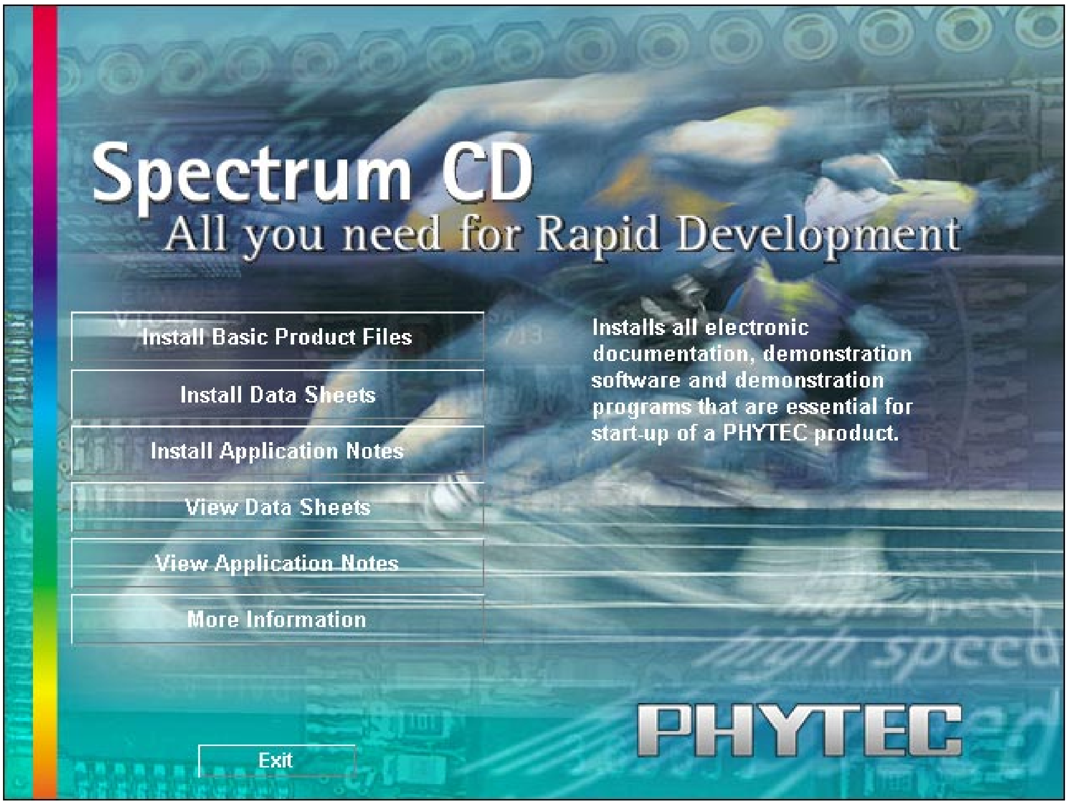
- installing Rapid Development Kit software
- starting PHYTEC FlashToolsOCF for Windows download utility
- interfacing the phyCORE-P89C51RD2, mounted on the phyCORE Development Board LD 5V, to a host-PC
- downloading example user code in Intel hexfile format from a host-PC to the controller’s on-chip Flash memory using PHYTEC FlashToolsOCF

2.1 Installing Rapid Development Kit Software

- Insert the PHYTEC Spectrum CD into the CD-ROM drive of your host-PC

The PHYTEC Spectrum CD should automatically launch a setup program that installs the software required for the Rapid Development Kit as specified by the user. Otherwise the setup program ***start.exe*** can be manually executed from the root directory of the PHYTEC Spectrum CD.

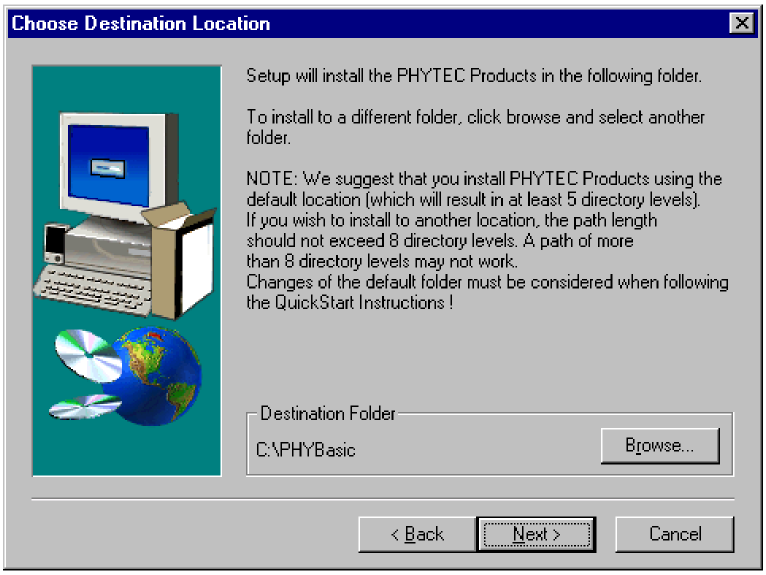
The following window appears:



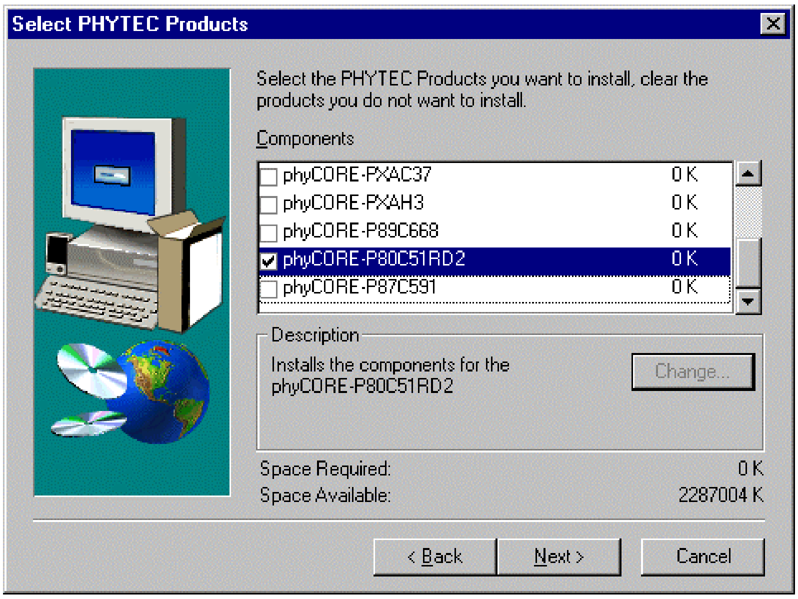
- Choose the *Install Basic Product Files* button.
- After accepting the Welcome window and license agreement, select the destination location for installation of Rapid Development Kit software and documentation.

The default destination location is ***C:\PHYBasic***. 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 decide to individually 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.



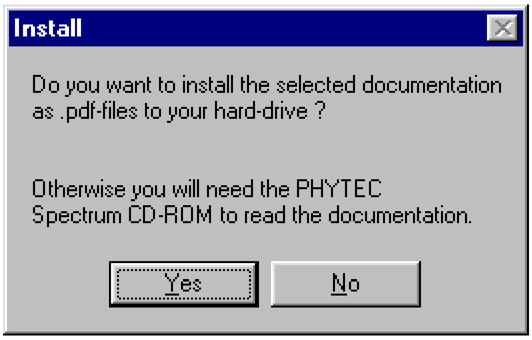
- In the next window, select your Rapid Development Kit of choice from the list of available products. By using the *Change* button, advanced users can select in detail which options should be installed for a specific product.



All Kit-specific content will be installed to a Kit-specific subfolder of the Rapid Development Kit root folder that you have specified at the beginning of the installation process.

All software and tools for this phyCORE-P89C51RD2 RDK will be installed to the ***\PHYBasic*** folder on your hard-drive.

- In the next dialog you must choose whether to copy the selected documentation as **.pdf* files to your hard drive or to install a link to the file on the Spectrum CD.



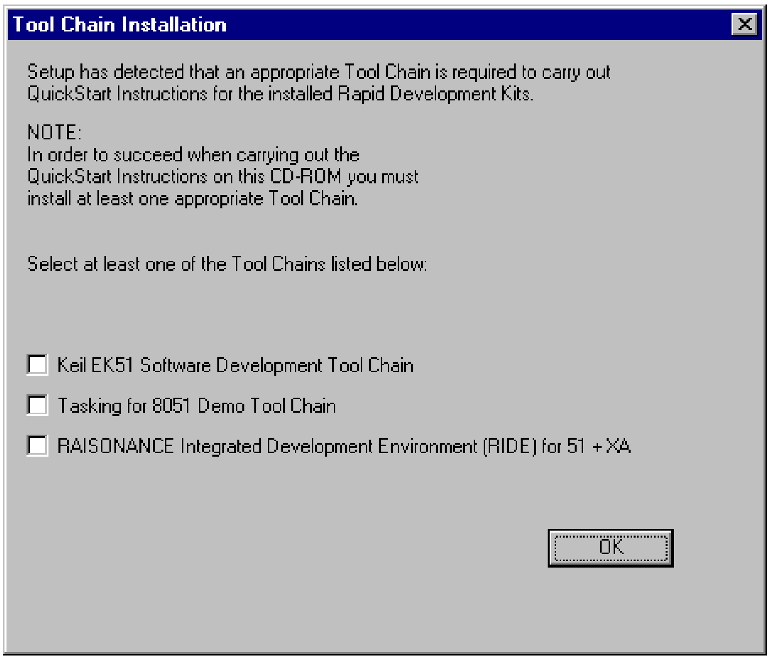
If you decide **not** to copy the documentation to your hard-drive, you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.

If you decide to copy the electronic documentation to your hard drive, the documentation for this phyCORE-P89C51RD2 RDK will also be installed to the kit-specific subfolder. The manuals of the phyCORE Development Board LD 5V are copied to their own specific subfolder (*e.g.* ***\PHYBasic\DevBLD5V***) because each Development Board is suitable for multiple SBC's and is not dedicated to a specific RDK.

Setup will now add program icons to the program folder, named *PHYTEC*.

- Click on *Finish* to complete the installation of PHYTEC products.

- In the next window, you choose the Keil EK51 Software Development tool chain[1].



The applicable Keil 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.

We recommend that you install µVision2 from the Spectrum CD-ROM even if other versions of µVision2 are already installed on your system. These QuickStart Instructions and the demo software included on the CD-ROM have been specifically tailored for use with one another.

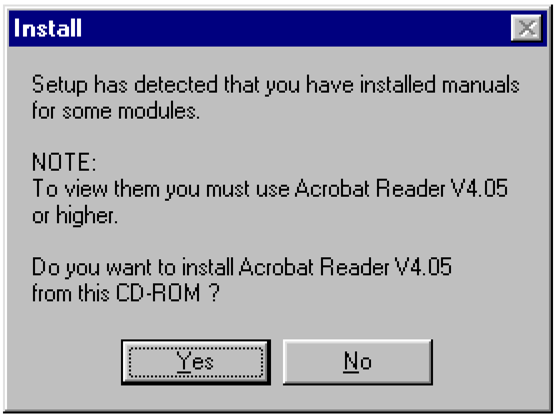
- After accepting the Welcome window and license agreement, select the destination location for installation of the Keil evaluation development tool chain. The default location is ***C:\Keil***.

1: If installing a different Software Development tool chain, *please refer to the applicable version of the QuickStart manual.*

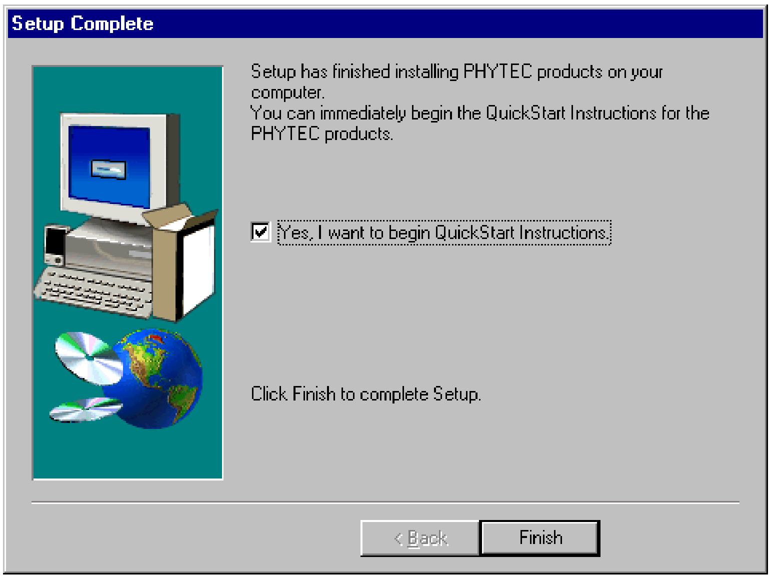
The Keil evaluation development tool chain will be installed to your hard drive. Additional software, such as Adobe Acrobat Reader, will also be offered for installation.

In the following windows, you can install FlashTools for on-chip Flash (OCF) software and the Acrobat Reader.

The applicable FlashTools software 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.



- Decide if you want to begin the QuickStart Instruction immediately by selecting the appropriate checkbox and click on *Finish* to complete the installation.



2.2 Interfacing the phyCORE-P89C51RD2 to a Host-PC

Connecting the phyCORE-P89C51RD2, mounted on the PHYTEC phyCORE Development Board LD 5V, to your computer is simple:

- As shown in the figure below, if the phyCORE module is not already preinstalled, mount it pins-down onto the Development Board's receptacle footprint (X6).
- Ensure that pin 1 of module (denoted by the hash stencil mark on the PCB) matches pin 1 of the receptacle on the phyCORE Development Board LD 5V.
- Ensure that there is a solid connection between the module pins and the phyCORE Development Board LD 5V receptacle.

Caution:
Take precautions not to bend the pins when the phyCORE module is removed from and inserted onto the phyCORE Development Board LD 5V.

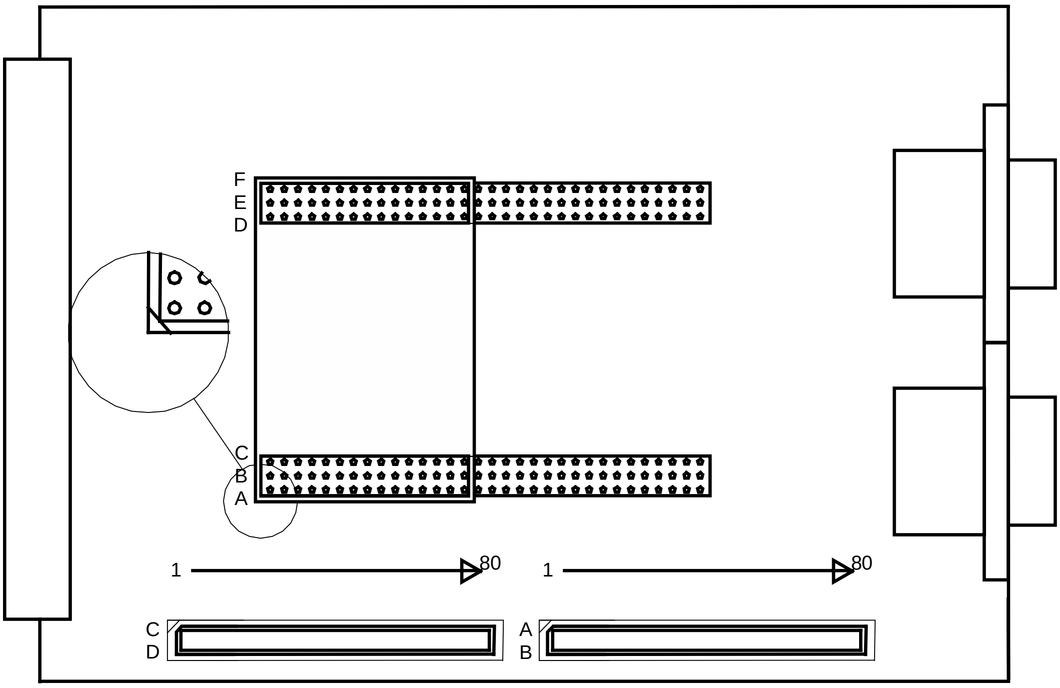


Figure 2: Mounting the phyCORE-P89C51RD2 onto the phyCORE Development Board LD 5V

- Configure the jumpers on the phyCORE Development Board LD 5V as indicated below. This correctly routes the RS-232 signals to the DB-9 connector (P1A = bottom) and connects the Development Board's peripheral devices to the phyCORE module.

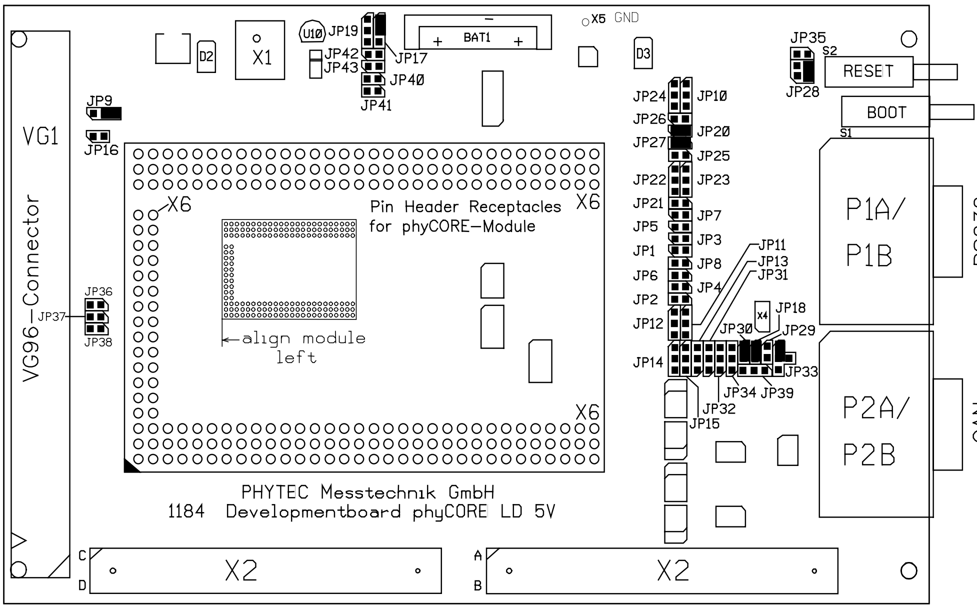


Figure 3: Important Connectors, Buttons and Suitable Jumper Settings on the phyCORE Development Board LD 5V

- Connect the RS-232 interface of your computer to the DB-9 RS-232 interface on the phyCORE Development Board LD 5V (P1A = bottom) using the included serial cable.
- Using the included power adapter, connect the power socket on the board (X1) to a power supply (*refer to Figure 4 for the correct polarity*).

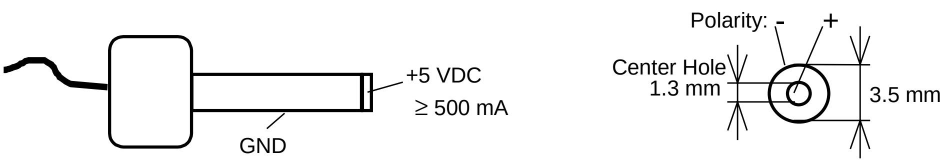


Figure 4: Power Connector

- Simultaneously press the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board LD 5V, first releasing the Reset and then, two or three seconds later, release the Boot button.

This sequence of pressing and releasing the Reset (S2) and Boot (S1) button renders the phyCORE-P89C51RD2 to the Flash programming mode. (FPM). Use of FlashToolsOCF always requires the phyCORE-P89C51RD2 to be in FPM. *See section 2.4, “Downloading Example Code with FlashToolsOCF”* for more details.

The phyCORE-P89C51RD2 should now be properly connected via the phyCORE Development Board LD 5V to a host-PC and power supply. After executing a Reset and rendering the board in Flash programming mode, you are now ready to program the phyCORE-P89C51RD2. This phyCORE module/phyCORE Development Board LD 5V combination is also referred to as “target hardware”.

2.3 Starting PHYTEC FlashToolsOCF for Windows

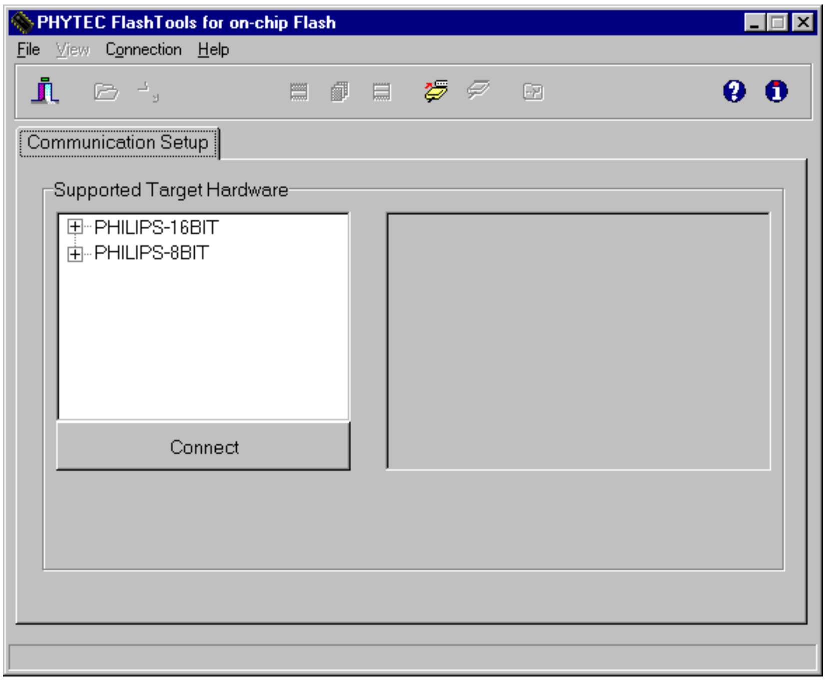
FlashTools for on-chip Flash should have been installed during the initial setup procedure as described in *section 2.1*. If not, you can manually install FlashToolsOCF by executing ***setup.exe*** from within the ***\Software\FTocF*** folder of your PHYTEC Spectrum CD.
FlashToolsOCF for Windows is a utility program that allows download of user code in ****.hex*** file format from a host-PC via an RS-232 connection to the on-chip Flash on the Philips P89C51RD2 controller populating the phyCORE module.
The PC-side software FlashToolsOCF has been designed to work together with the Boot Loader firmware located in an on-chip BOOT-ROM of the P89C51RD2 controller. Proper connection of a PHYTEC phyCORE module to a host-PC enables the software portion of FlashToolsOCF to recognize and communicate to the firmware portion.

- You can start FlashToolsOCF by selecting it from the *Programs* menu using the Windows *Start* button.

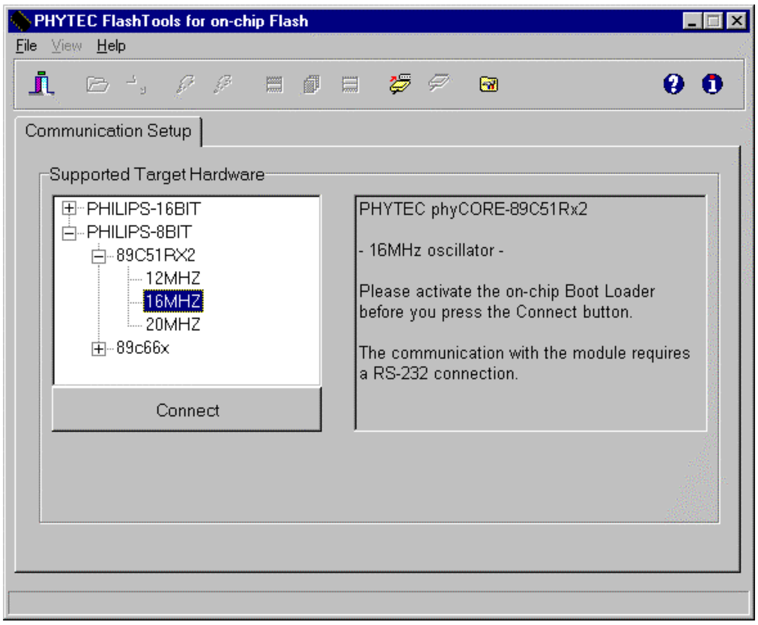
It is recommended that you drag the FlashToolsOCF icon onto the desktop of your PC. This enables easy start of FlashToolsOCF by double-clicking on the icon.

2.4 Downloading Example Code with FlashToolsOCF

- Start FlashToolsOCF by double-clicking on the FlashToolsOCF icon or by selecting *FlashToolsOCF* from within ***Programs|Phytec*** program group.
- The Communication Setup tab of the FlashTools for on-chip Flash tabsheet window will now appear.



- Select the target hardware (*PHILIPS-8BIT*), the controller type (*P89C51Rx2*) and the oscillator frequency. To configure the correct oscillator frequency, identify the value of the oscillator located next to the controller on the phyCORE module. In most cases the value is *16 MHz*. If the oscillator has an uneven value, you can determine the appropriate frequency by rounding up to the next even value (e.g. 11.059 MHz → 12 MHz). The correct frequency must be indicated in order to properly start FlashToolsOCF.



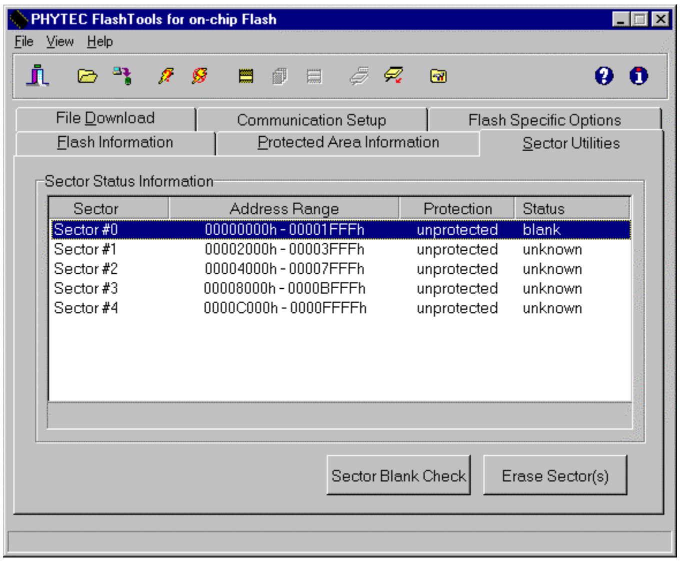
Note:
Always ensure that the phyCORE-P89C51RD2 is in Flash programming mode before pressing the *Connect* button (*refer to section 2.2*).

- Click on the *Connect* button.
- At the *Communication Setup* window of the FlashToolsOFC tabsheet, choose the correct serial port for your host-PC and a 19,200 baud rate.
- Click the *OK* button to establish a connection to the target hardware.

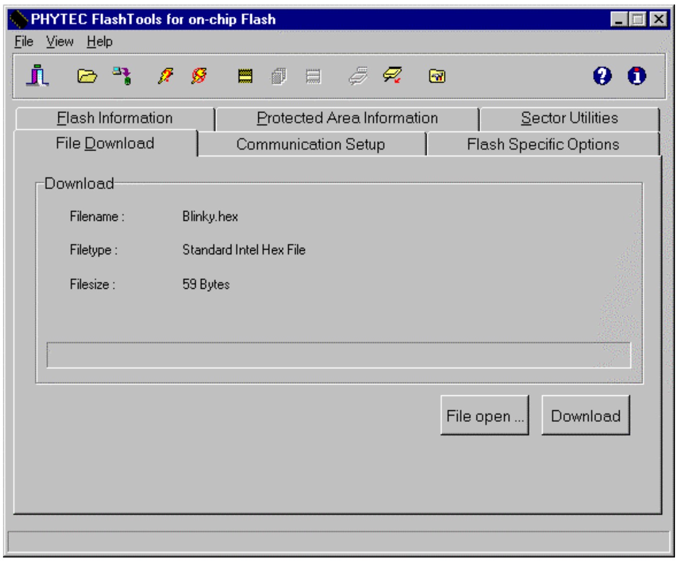
The microcontroller firmware tries to automatically adjust to the baud rate selected within the baud rate tab. However, it may occur that the selected baud rate cannot be attained. This results in a connection error. In this case, try other baud rates to establish a connection.
Before attempting each connection, be sure to reset the target hardware and render it into Flash programming mode (FPM) as described in *section 2.2*

Returning to the FlashToolsOCF tabsheet window, you will see tabs for the following:

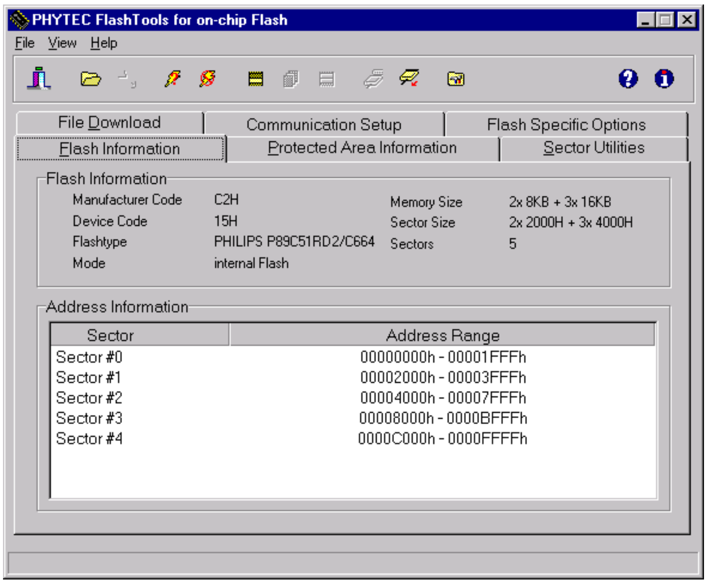
Sector Utilities allows blank check and erasure of individual sectors of Flash memory:



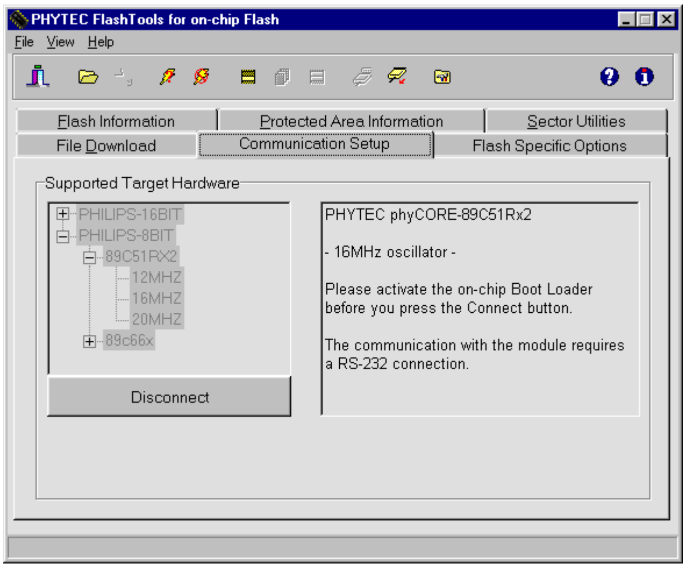
File Download downloads specified hexfiles to the target hardware:



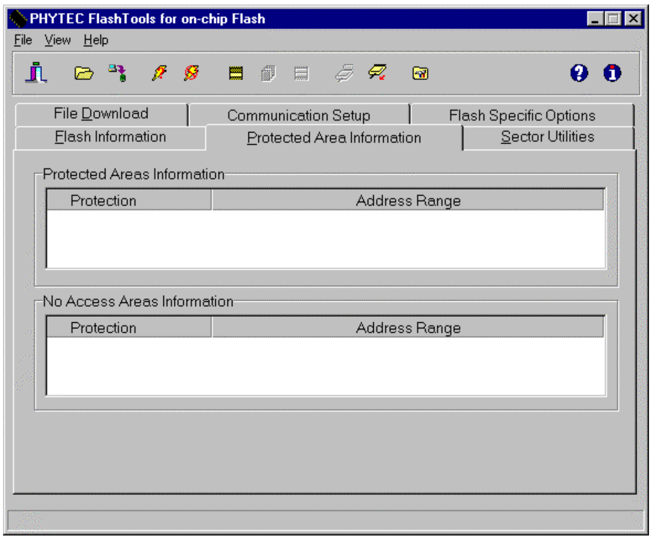
Flash Information shows Flash type, sector and address ranges in Flash memory:



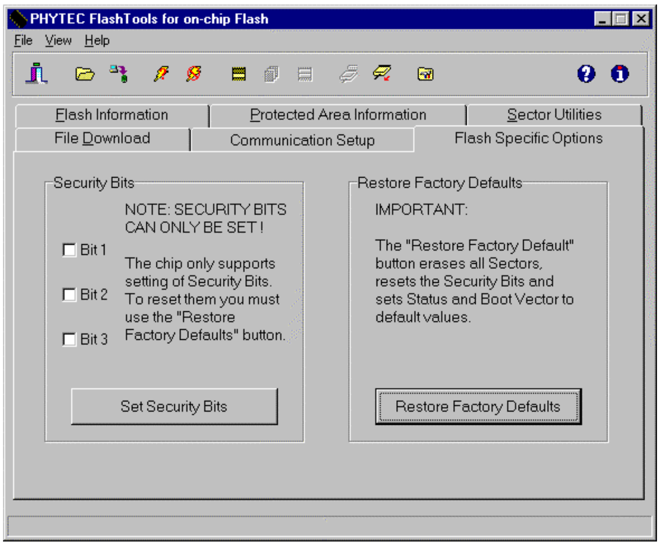
Communication Setup allows connection to and disconnection from the board (this is the same window that was used when you first entered FlashToolsOCF):



Protected Areas Information shows protected areas of Flash memory:



Flash Specific Options enables setting of the controller's security bits.



For the P89C51RD2 controller, the security bit functions are as follows:

Security Bit #1: inhibits writing to Flash
Security Bit #2: inhibits Flash verification
Security Bit #3: disables external memory

The *Restore Factory Defaults* button erases all Flash sectors, resets the security bits and sets status and boot vector to default values, i.e. status vector is set to FFh and boot vector is set to the BOOT-ROM address FCh (P89C51RD2).

2.4.1 "Blinky"

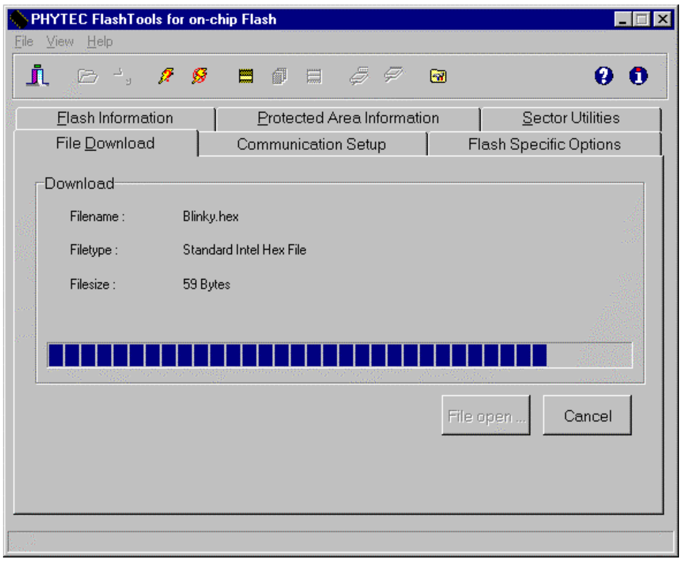
The "Blinky" example downloads a program to the Flash that, when executed, manipulates the LED D3 on the phyCORE Development Board LD 5V that is located above the jumper field (*refer to Figure 3*).

- Returning to the FlashToolsOCF tabsheet, choose the *Sector Utilities* tab, highlight *Sector #0 - #2* and click on the *Erase Sector(s)* button to erase this memory bank.
- Wait until the status check in the lower left corner of the FlashToolsOCF tabsheet finishes, returning the connection properties description to the lower left corner of the window.
- Next choose the *File Download* tab and click on the *File Open* button.

The hexfile has already been installed to your hard drive during the installation procedure.

- Browse to the correct drive and path for the phyCORE-P89C51RD2 Demo folder (default location ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Blinky\Blinky.hex***) and click *Open.*

- Click on the *Download* button. You can watch the status of the download of the ***Blinky.hex*** into on-chip Flash memory in the Download window.



If the selected Flash sector into which you wish to download code is not empty (i.e. erased), a warning dialog box will appear, indicating “Location not empty! Please erase location and try again”. In this event, select the *Sector Utilities* tab from the FlashToolsOCF tabsheet, highlight *Sector #0 - #2* and click on the *Erase Sector(s)* button to erase this memory bank. Then repeat the download procedure.

- At the end of the download, a sector-by-sector status check of the Flash memory can be viewed in the lower left corner of the FlashToolsOCF tabsheet window. Wait until the status check finishes before returning to work with the board. Once the status check is complete, the downloaded code can be executed.
- Returning to the *Communication* tab, click on the *Disconnect* button and exit FlashToolsOCF.
- Press the Reset button (S2) on the phyCORE Development Board LD 5V to reset the target hardware and to start execution of the downloaded software.
- Successful execution of the program will flash the LED D3 with equal on and off durations.

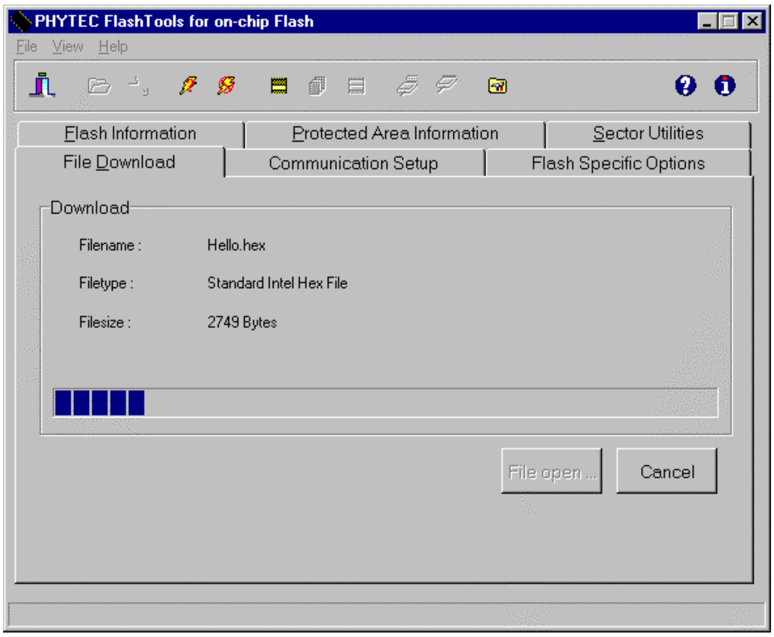
2.4.2 "Hello"

The "Hello" example downloads a program to the Flash that, when executed, performs an automatic baud rate detection and sends a character string from the target hardware back to the host-PC. The character string can be viewed with a terminal emulation program. This example program provides a review of the FlashToolsOCF download procedure. For detailed commentary on each step, described below in concise form, *refer back to sections 2.2 through 2.4.1*.

- Ensure that the target hardware is properly connected to the host-PC and a power supply.
- Reset the target hardware and force it into Flash programming mode by simultaneously pressing the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board LD 5V and then releasing first the Reset and, two or three seconds later, the Boot button.
- Start FlashToolsOCF.
- At the *Communication Setup* tab of the FlashToolsOCF tabsheet select the correct target hardware. In the next window, specify the proper serial port and transmission speed (19,200 Baud) for communication between host-PC and target hardware and click the *Connect* button to establish a connection to the target hardware.
- Returning to the FlashToolsOCF tabsheet, choose the *Sector Utilities* tab, highlight *Sector #0 - #2* and click on the *Erase Sector(s)* button to erase the selected memory sectors.
- Wait until the status check in the lower left corner of the FlashToolsOCF tabsheet finishes, returning the connection properties description to the lower left corner of the window.
- Next choose the *File Download* tab and click on the *File Open* button.

The demo hexfile has already been installed to your hard drive during the installation procedure.

- Browse to the correct drive and path for the phyCORE-P89C51RD2 Demo folder (default location ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Hello\Hello.hex***) and click *Open.*
- Click on the *Download* button. You can watch the status of the download of the ***Hello.hex*** into on-chip Flash memory in the Download window.

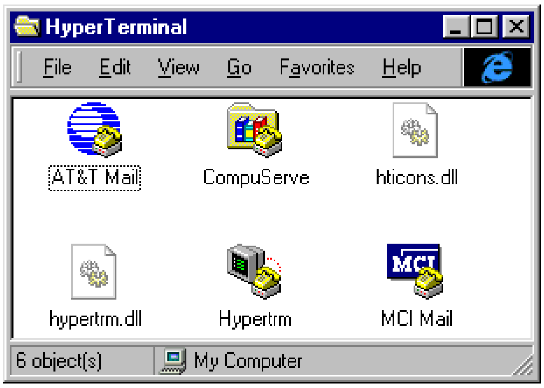


If the selected Flash Sector into which you wish to download code is not empty (i.e. erased), a warning dialog box will appear, indicating "Location not empty! Please erase location and try again". In this event, select the *Sector Utilities* tab from the FlashToolsOCF tabsheet, highlight *Sector #0 - #2* and click on the *Erase Sector(s)* button to erase the selected memory sectors. Then repeat the download procedure.

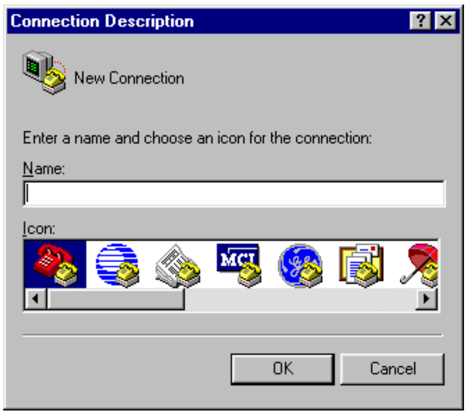
- At the end of the download, a sector-by-sector status check of the Flash memory can be viewed in the lower left corner of the FlashToolsOCF tabsheet window. Wait until the status check finishes before returning to work with the board. Once the status check is complete, the downloaded code can be executed.
- Returning to the *Communication* tab, click on the *Disconnect* button and exit FlashToolsOCF.

Monitoring the execution of the Hello 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[2]:
- Double-click on the HyperTerminal icon *Hypertrm* to create a new HyperTerminal session.



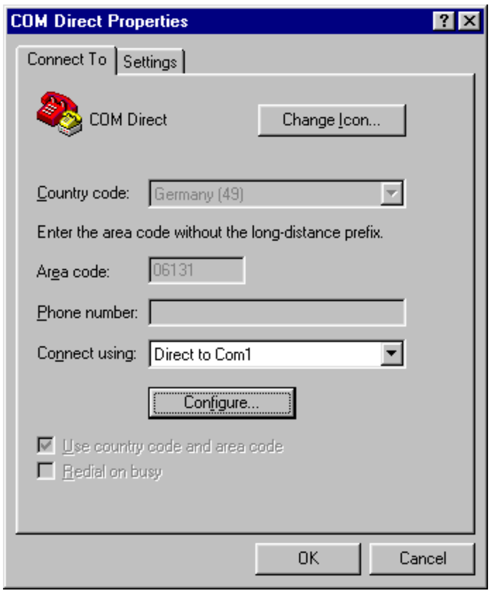
- The Connection Description window will now appear. Enter "COM Direct" in the *Name* text field.



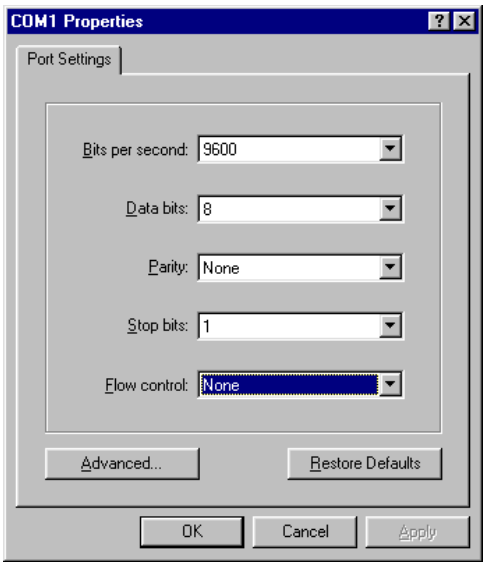
- Next click on *OK*. This creates a new HyperTerminal session named "COM Direct" and advances you to the next HyperTerminal window.

[2] : The HyperTerminal window has a different appearance for different versions of Windows.

- The *COM Direct Properties* window will now appear. Specify *Direct to COM1/COM2* under the *Connect Using* pull-down menu (be sure to indicate the correct COM setting for your system).

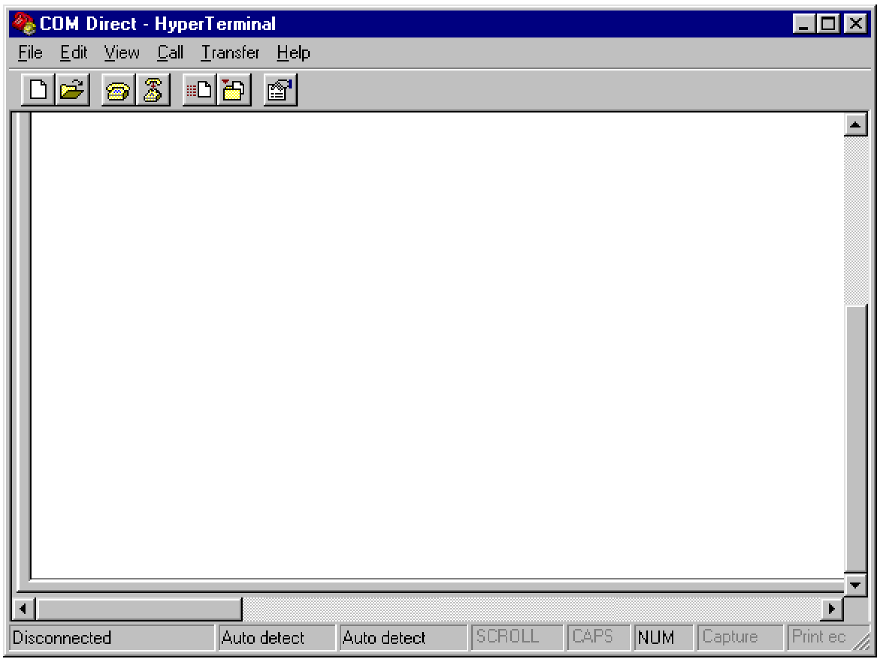


- Click the *Configure* button in the *COM Direct Properties* window to advance to the next window (*COM1/COM2 Properties*).



- Then set the following COM parameters: Bits per second = *9600*; Data bits = *8*; Parity = *None*; Stop Bits = *1*; Flow Control = *None*.

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



- Resetting the phyCORE Development Board LD 5V (at S2) will execute the ***Hello.hex*** file loaded into the Flash.
- Now push the <Space> bar on your keyboard once to start the automatic baud rate detection on phyCORE-P89C51RD2 module.
- Successful execution will send the character string "*Hello World*" from the target hardware to the HyperTerminal window.

Pressing any other key than the <Space> bar leads to an improper baud rate since the automatic baud rate detection is based on the timing measurement during the transmission of a well known character – the <Space> character. As a result you may get incoherent characters in the HyperTerminal window.

- Click the disconnect icon  in HyperTerminal toolbar and exit HyperTerminal.
- If no output appears in the HyperTerminal window check the power supply, the COM parameters and the RS-232 connection.

You have now successfully downloaded and executed two pre-existing example programs in Intel ****.hex*** file format.

3 Getting More Involved

What you will learn with this example:

- how to start the µVision2 tool chain
- how to configure the µVision2 IDE (Integrated Development Environment)
- how to modify the source code from our examples, create a new project and build and download an output ***.hex** file to the target hardware

3.1 Starting the Keil µVision2 Tool Chain

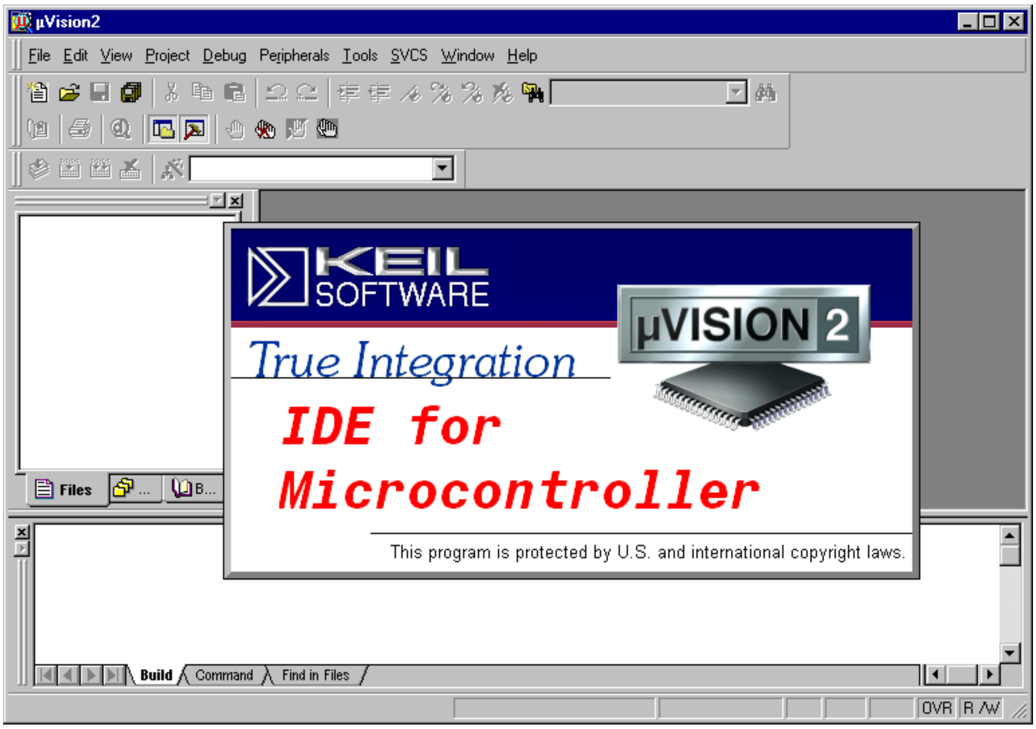
The Keil µVision2 evaluation software development tool chain should have been installed during the install procedure, as described in *section 2.1*.

You can also manually install µVision2 by executing ***install.bat*** from within the ***\Software\Keil\Ek8051*** directory of your PHYTEC Spectrum CD.

Note:
It is necessary to use the Keil tool chain provided on the accompanying Spectrum CD in order to complete these QuickStart Instructions successfully. Use of a different version could lead to possible version conflicts, resulting in functional problems.

- Start the tool chain by selecting *Keil µVision2* from within the *Programs* group.

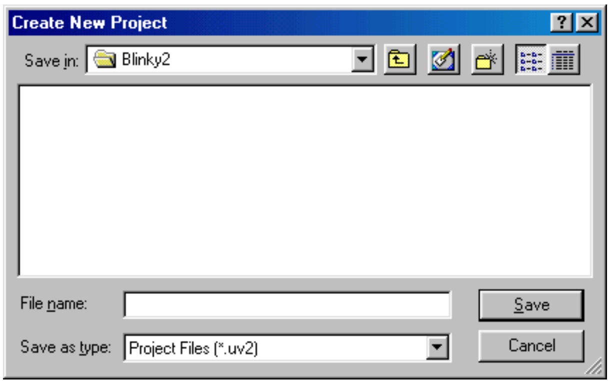
After you start µVision2, the window shown below appears. From this window you can create projects, edit files, configure tools, assemble, link and start the debugger. Other 3rd party tools such as emulators can also be started from here.



3.2 Creating a New Project and Adding an Existing Source File

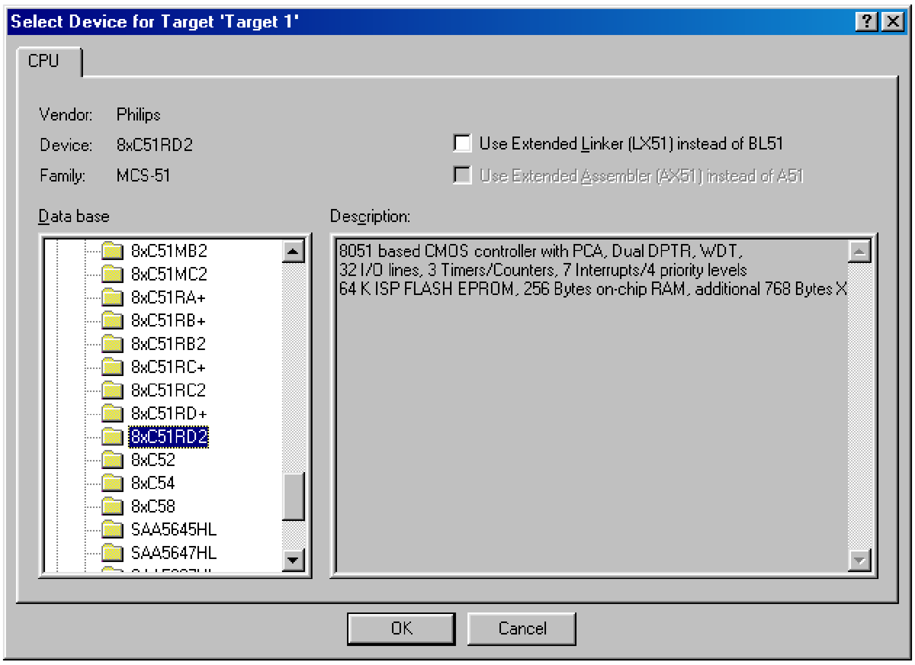
µVision2 automatically loads the most recently opened project. If you find an existing project when starting µVision2, close it by selecting the *Project* menu and *Close* the project.

- To create a new project file select from the µVision2 menu ***Project|New Project…***. This opens a standard Windows dialog box that asks you for the new project file name.
- Change to the project directory created by the installation procedure (default location ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Blinky2***).



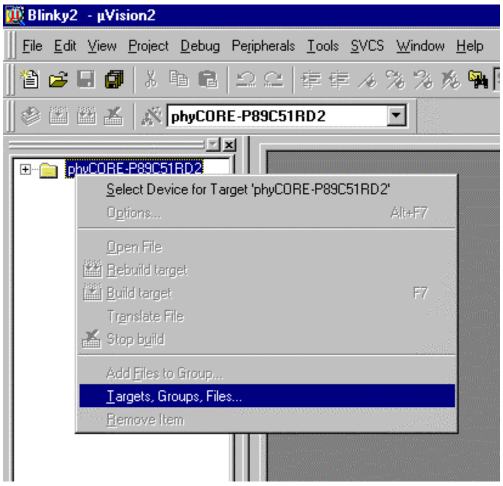
- In the text field '*File name*', enter the file name of the project you are creating. For this example, enter the name ***Blinky2*** and click on *Save*.

- The ***Select Device for Target 'Target 1'*** window will appear. Double-click on *Philips* in the CPU vendor data base list. The phyCORE-P89C51RD2 is equipped with a *Philips 89C51RD2*. Choose this controller type (8xC51RD2) from the list as shown below. This selection sets necessary tool options for the Philips 89C51RD2 device and pre-configures additional settings for the device.

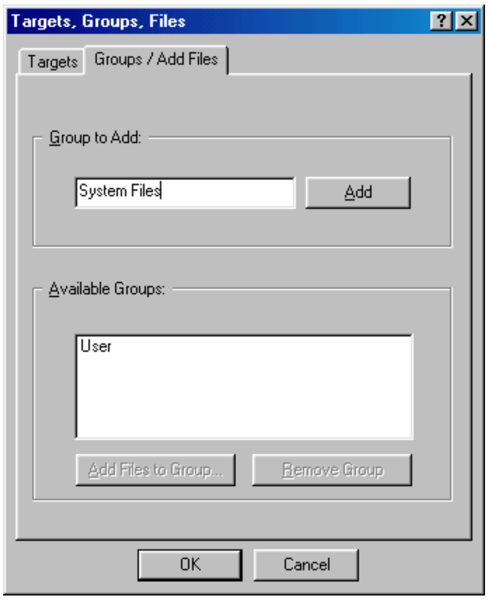


- Click on *OK*.
- Now click on *Target1* within the **Project Window - Files** tab. *Target1* is now highlighted. Click on *Target1* again to enable the edit mode. Change the default name of the target to *phyCORE-P89C51RD2*.

- Select the file group *Source Group 1* in the **Project Window – Files** tab and click on it to change the name into *User*.
- Right-click in the **Project Window – Files** to open a new window. Choose the option ***Targets, Groups, Files...***.

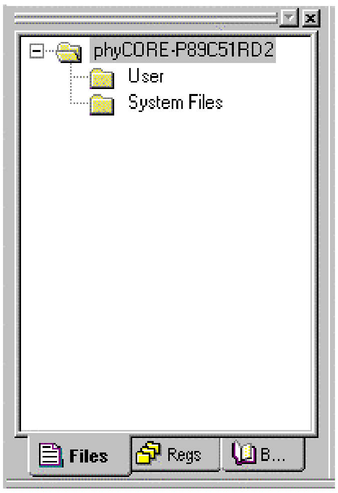


- Select the tab **Groups / Add Files** and type the new group name *System Files* in the ***Group to Add*: ** section.

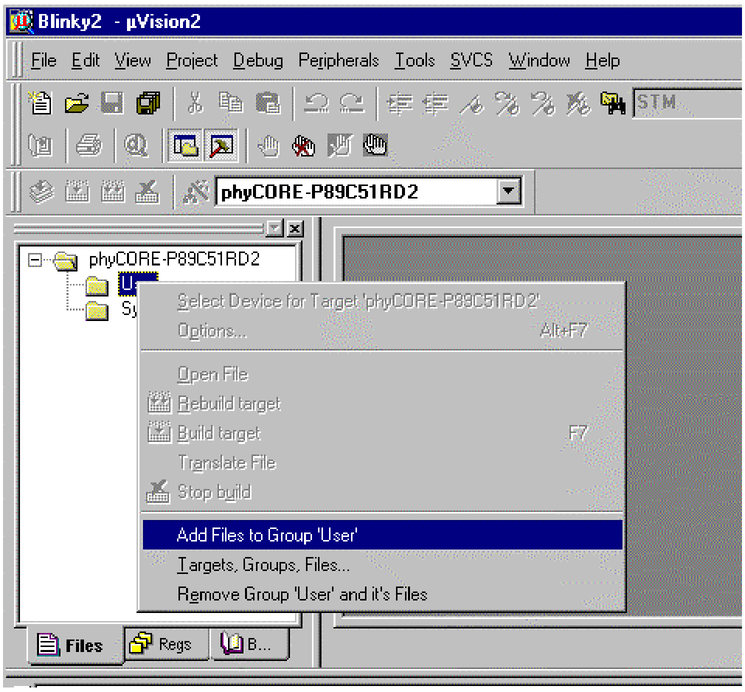


- Click on *Add* and then on *OK.*

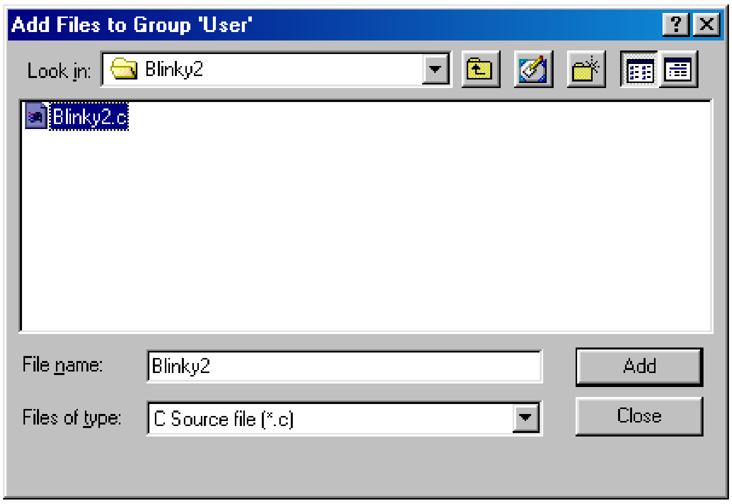
- Your project file structure should now look like this:



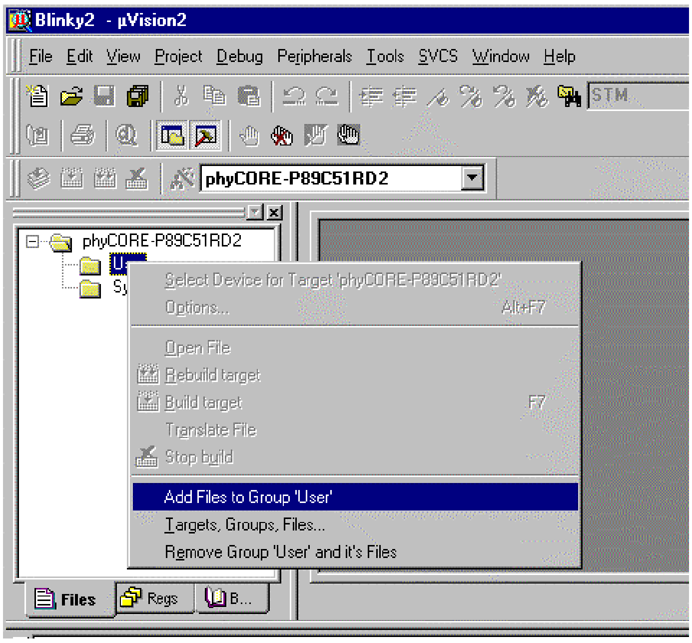
- In order to add ***Blinky2.c*** to our project right-click on the *User* group to open a menu. Select the option ***Add Files to Group 'User'*** to open the standard files dialog.



- Select the file ***Blinky2.c***.

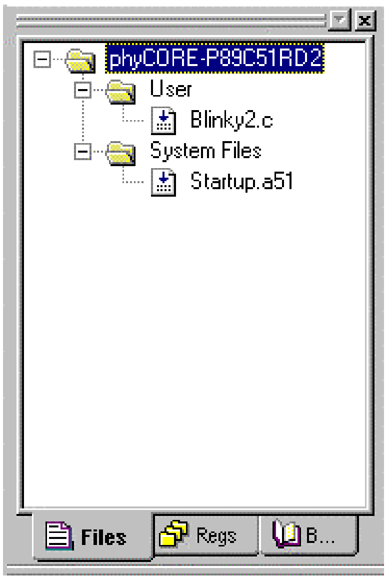


- Click on the *Add* button to add the ***Blinky2.c*** file to your current project window.



- Close the window.
- Now right-click on group *System Files* and add the file ***Startup.a51***. You have to change the file type to "***Asm Source file (*.a, *.src)***" in the *File of types* pull-down menu to see this file.

- Your project window should now look like this:

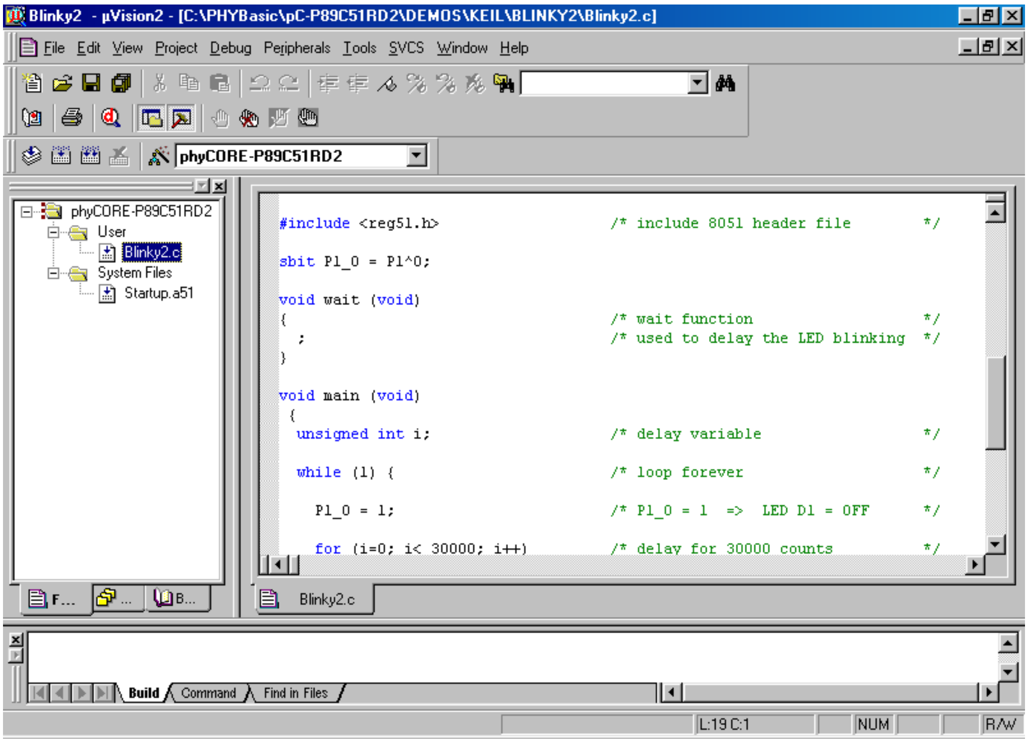


At this point you have created a project called ***Blinky2.uv2*** and added an existing C source file called ***Blinky2.c*** and an existing assembler file called ***Startup.a51***.

The next step is to modify the C source before building your project. This includes compiling, linking, locating and creating the hexfile.

3.3 Modifying the Source Code

- Double click on ***Blinky2.c*** to open it in the source code editor.



- Locate the following code section. Modify the section shown below (the values shown in bold and italic) from the original (30,000) counts to the indicated values:

```
while (1) {                  /* loop forever                */
    *pPD = *pPD & 0xEF;      /* output over PD port to LED D3  */
                             /* bit 1 of port PD = LED D3 = OFF */
    for (i=0; i< 30000; i++) /* delay for 30000 counts      */
    {
      wait ();               /* call wait function          */
    }
    *pPD = *pPD | 0x10;      /* output over PD port to LED D3  */
                             /* bit 1 of port PD = LED D3 = ON  */
    for (i=0; i< 60000; i++) /* delay for 60000 counts      */
    {
      wait();                /* call wait function          */
    }
  }
```

This will change the LED on/off ratio.

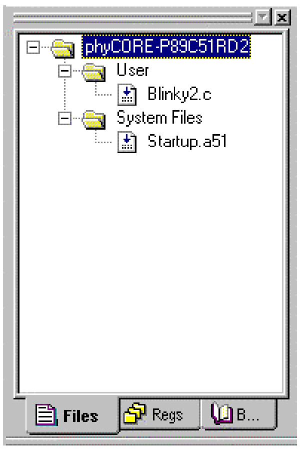
3.4 Saving the Modifications

- Save the modified file by choosing *File|Save* or by clicking the floppy disk icon .

3.5 Setting Tool Chain Options

Keil includes a Make utility that can control compiling and linking source files in several programming languages. Before using the Make utility, macroassembler, C compiler or linker you must configure the corresponding options. Most of the options are set by specifying the device for the project. Only the external memory and output options must be set.

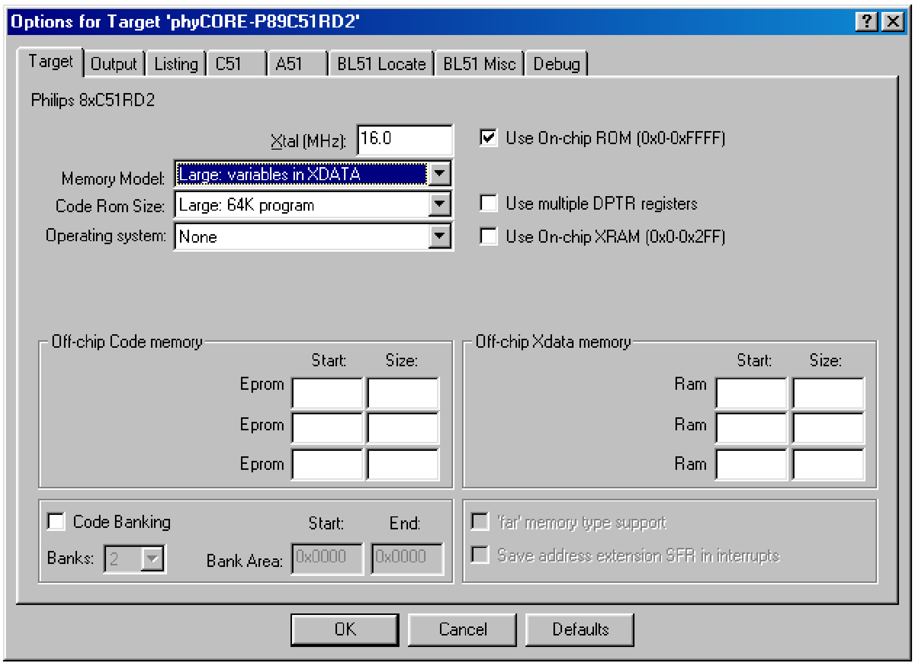
Enter the changes as indicated below and leave all other options set to their default values. µVision2 allows you to set various options with mouse clicks and these are all saved in your project file.



- Select the target *phyCORE-P89C51RD2* within the project window.

To configure the Target:

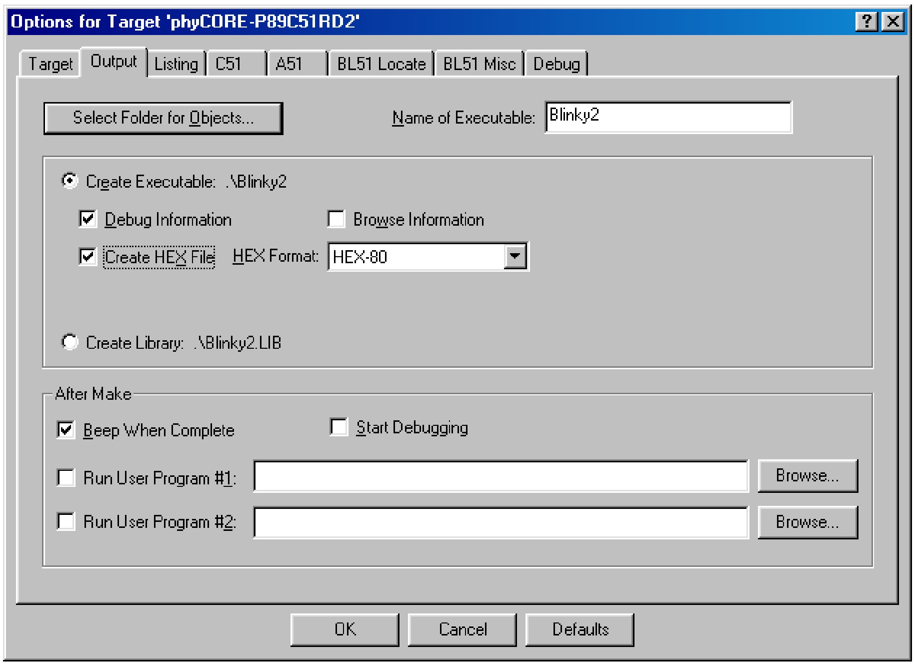
- Open the ***Project|Options for Target 'phyCORE-P89C51RD2'*** menu and change the default settings to the correct values for the phyCORE-P89C51RD2 as shown in the picture below. This includes settings for the clock frequency of your phyCORE module and enabling the *Use on-chip ROM* checkbox.



- The phyCORE-P89C51RD2 is populated with a 32 kByte SRAM device. The ***Memory Model*** should be set to *Large: variables in XDATA.*

To configure the Output options:

- Select the ***Output*** tab and activate the *Create HEX File* checkbox. With this option an Intel ****.hex*** file will be created for download.



- Click on *OK*.

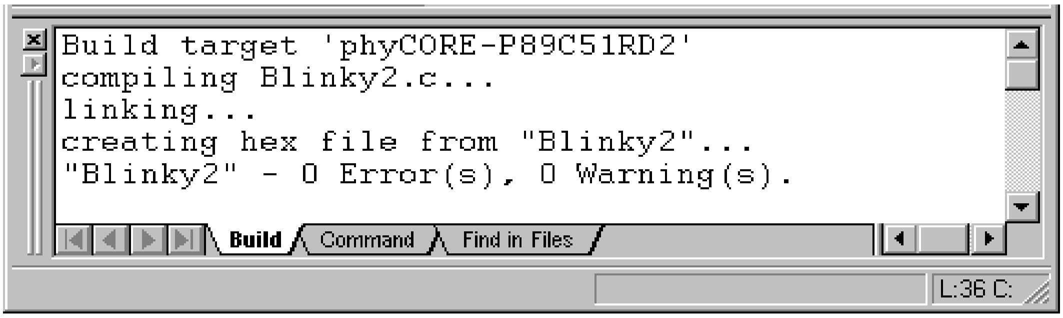
3.6 Building the Project

You are now ready to run the compiler and linker using the Make utility.

- Click on the *Build Target* icon from the µVision2 toolbar or press <F7>.

If the program specified (***Blinky2.c***) contains any errors, they will be shown in an error dialog box on the screen.

If there are no errors, the code is compiled and linked and the executable code is ready to be downloaded to the module. This is shown in the ***Output Window***, which indicates *"Blinky2" - 0 Errors, 0 Warnings*. The created hexfile will have the name of the project with ***.hex*** as the filename extension (in this case ***Blinky2.hex***).



Note:
A machine-readable, executable hexfile has been created. Other files (e.g. list files ****.lst*** and map files ****.map***) are generated to help the debugging or troubleshooting and error searching process.

- If a list of errors appears, use the editor to correct the error(s) in the source code, save the file and (re-)build the project.

3.7 Downloading the Output File

- Exit Keil µVision2.
- Reset the target hardware and force it into Flash programming mode by simultaneously pressing the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board LD 5V and then releasing first the Reset and, two or three seconds later, the Boot button.
- Start FlashTools for on-chip Flash.
- At the *Communication Setup* tab of the FlashToolsOCF tabsheet, specify the desired target hardware and click the *Connect* button. Next select serial port and transmission speed to configure communication between host-PC and target hardware and to establish connection to the target hardware.
- Returning to the FlashToolsOCF tabsheet, choose the *Sector Utilities* tab, highlight *Sector #0 - #2* and click on the *Erase Sector(s)* button to erase the selected Flash sectors.
- Wait until the status check in the lower left corner of the FlashToolsOCF tabsheet finishes, returning the connection properties description to the lower left corner of the window.
- Choose the *File Download* tab and click on the *File Open* button.
- Browse to the correct drive and path for the phyCORE-P89C51RD2 Demo folder (default location ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Blinky2\Blinky2.hex*** and click *Open.*
- Click on the *Download* button and view the download procedure to the board in the status window.
- Returning to the *Communication* tab, click on the *Disconnect* button and exit the FlashToolsOCF.
- Press the Reset button (S2) on the Development Board.

If the modified hexfile properly executes, the LED should now flash in a different mode with different on and off durations.

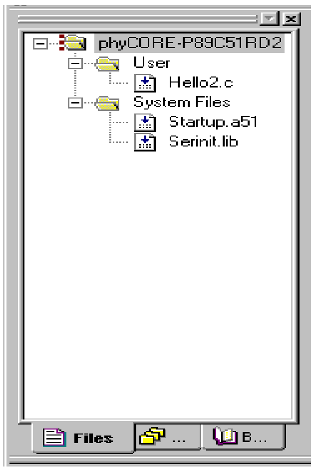
You have now modified source code, recompiled the code, created a modified downloadable hexfile, and successfully executed this modified code.

3.8 "Hello2"

A return to the "Hello" program allows a review of how to modify source code, create and build a new project, and download the resulting output file from the host-PC to the target hardware. For detailed commentary on each step, described below in concise form, *refer back to the "Blinky2" example starting in section 3.1*.

3.8.1 Creating a New Project

- Start Keil µVision2 and close all projects that might be open.
- Open the ***Project*** menu and create a new project called ***Hello2.uv2*** within the existing project folder ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Hello2*** (default location) on your hard-drive. Select the Philips 89C51RD2 in the CPU vendor data base list.
- Add ***Hello2.c***, ***Serinit.lib*** and ***Startup.a51*** from within the project folder to the project ***Hello2.uv2***.
- Your project window should now look like this:



- *Save* the project.

At this point you have created a project called ***Hello2.uv2*** consisting of a C source file called ***Hello2.c***, a library called ***Serinit.lib*** and an assembler file called ***Startup.a51***.

3.8.2 Modifying the Example Source

- Double click the file ***Hello2.c*** from within the project window.
- Use the editor to modify the *printf* command:

```
printf ("\x1AHello World\n")
```

to

```
printf ("\x1APHYTEC... Stick It In!\n")
```

- Save the modified file under the same name ***Hello2.c***.

3.8.3 Setting Tool Chain Options

- Open the ***Project|Options for Target...*** menu and change the default settings to the correct values as shown in *section 3.5*. This includes settings for the oscillator frequency of your phyCORE module in the *Xtal (MHz)* dialog box and activation of the *Use on-chip ROM* checkbox. Make sure that "*Large: variables in XDATA*" is active in the memory model selection.
- Modify the default options for the output file by selecting the *Create HEX File* checkbox in the ***Project|Options for Target....|Output*** tab. This will automatically create a hexfile for download to the phyCORE-P89C51RD2 after compiling.

3.8.4 Building the New Project

- Build the project.
- If any source file in the project contains errors, they will be shown in an error dialog box on the screen. Use the editor to correct the error(s) in the source code and save the file and repeat this section.

If there are no errors, the code is assembled and linked and the executable code is ready to be downloaded to the board.

3.8.5 Downloading the Output File

- Exit Keil µVision2.
- Reset the target hardware and force it into Flash programming mode by simultaneously pressing the Reset (S2) and Boot (S1) buttons on the phyCORE Development Board LD 5V and then releasing first the Reset and, two or three seconds later, the Boot button.
- Start FlashToolsOCF.
- At the *Communication Setup* tab of the FlashToolsOCF tabsheet, specify the desired target hardware and click the *Connect* button. Next specify the proper serial port and transmission speed (19,200 Baud) for communication between host-PC and target hardware and click on *OK* to establish connection to the target hardware.
- Returning to the FlashToolsOCF tabsheet, choose the *Sector Utilities* tab, highlight *Sector #0 - #2* and click on the *Erase Sector(s)* button to erase the selected Flash sectors.
- Wait until the status check in the lower left corner of the FlashToolsOCF tabsheet finishes, returning the connection properties description to the lower left corner of the window.
- Next choose the *File Download* tab and click on the *File Open* button.
- Browse to the correct drive and path for the phyCORE-P89C51RD2 Demo folder (default location ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Hello2\Hello2.hex***) directory.
- Click on the *Download* button and view the download procedure to the board in the status window.
- Returning to the *Communication* tab, click on the *Disconnect* button and exit FlashToolsOCF.

3.8.6 Starting the Terminal Emulation Program

- Start HyperTerminal and connect to the target hardware using the following COM parameters: Bits per second = *9600*; Data bits = *8*; Parity = *None*; Stop Bits = *1*; Flow Control = *None*.
- Resetting the phyCORE Development Board LD 5V (at S2) will execute the ***Hello2.hex*** file loaded into the Flash.
- Now push the <Space> bar on your keyboard once to start the automatic baud rate detection on phyCORE-P89C51RD2 module.
- Successful execution will send the modified character string **"PHYTEC... Stick It In!"** to the HyperTerminal window.
- Click the Disconnect icon .
- Close the Hyper Terminal program.

You have now modified source code, recompiled the code, created a downloadable hexfile, and successfully executed this modified code.

4 Debugging

This Debugging section provides a basic introduction to the debug functions included in the Keil software evaluation development tool chain. Using an existing example, the more important features are described. For a more detailed description of the debugging features, please refer to the appropriate manuals provided by Keil.

The µVision2 Debugger offers two operating modes that can be selected in the ***Project|Options for Target phyCORE-P89C51RD2*** dialog:

- The **Simulator** allows PC-based microcontroller simulation of most features of the 8051 microcontroller family without actually having target hardware. You can test and debug your embedded application before the hardware is ready. µVision2 simulates a wide variety of peripherals, including the serial port, external I/O, and timers. The peripheral set is selected when you select a CPU from the device database for your target.
- Advance GDI drivers, like the **Keil Monitor 51** interface, allow target-based debugging. With the Advanced GDI interface you may connect directly to the target hardware. Debugging on the target hardware also enables testing peripheral components of the application.

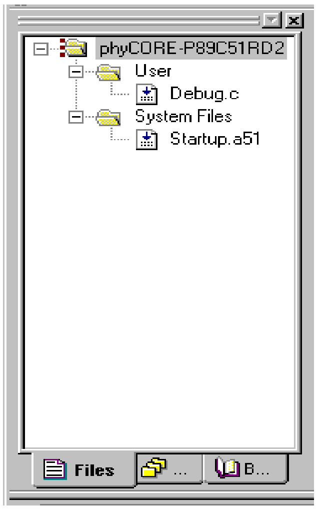
The following examples utilize the Keil Simulator environment.

Note:
Because the P89C51Rx2 controller family features on-chip Flash memory, the phyCORE-P89C51RD2 module does not support flexible memory management like other phyCORE modules populated with external Flash memory. This means that a monitor program cannot be loaded into the Flash, hence debugging with the Keil Monitor 51 interface is not possible. Programs can be evaluated before code is downloaded to the controller using the Keil Simulator.

4.1 Creating a Debug Project and Preparing the Simulator

4.1.1 Creating a New Project

- Start the Keil µVision2 environment and close all projects that might be open.
- Open the Project menu and create a new project called ***Debug.uv2*** within the existing project directory ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Debug*** (default location) on your hard drive.
- Rename the target of your project within the **Project Window – Files** tab into *phyCORE-P89C51RD2***.**
- Rename the file group *Source Group 1* within the **Project Window – Files** tab into *User* and add an additional file group named *System Files.*
- Add ***Debug.c*** to the file group *User* and ***Startup.a51*** to the file group *System Files* from within the project folder.
- Your project should now look like this:

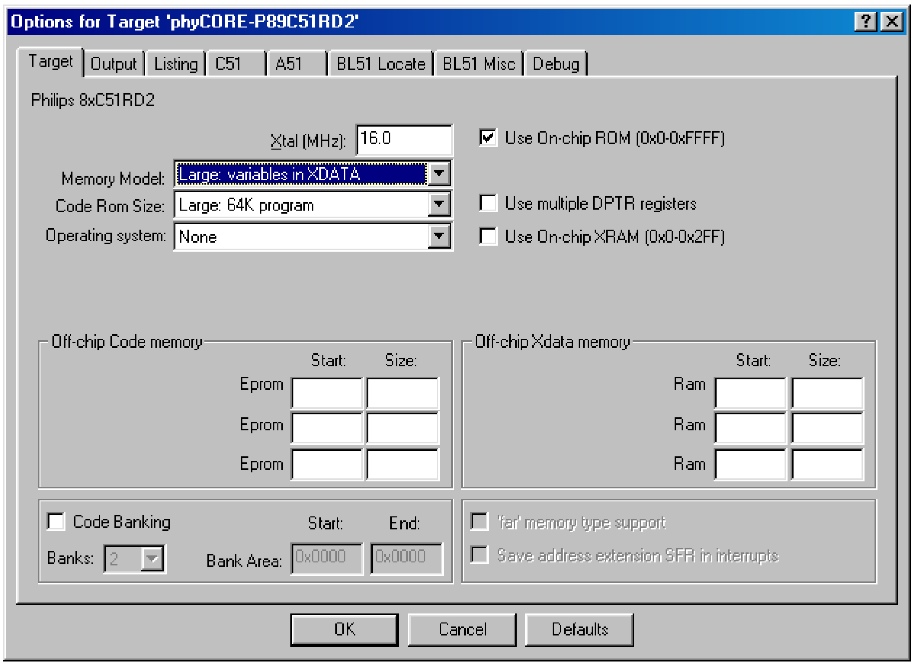


- Save the project.

At this point you have created a project called ***Debug.uv2,*** consisting of a C source file called ***Debug.c*** and the assembler file ***Startup.a51***.

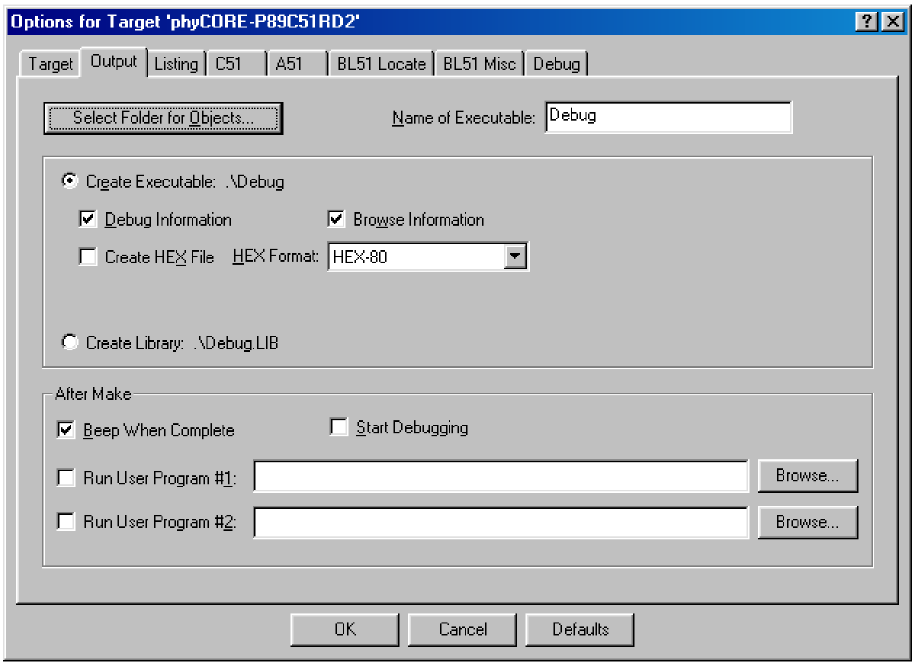
4.1.2 Setting Options for Target

- Open the ***Project|Options for Target 'phyCORE-P89C51RD2'*** menu and and enter the correct oscillator frequency of your phyCORE module in the *Xtal (MHz)* dialog box and select the *Use on-chip ROM* checkbox.



- The phyCORE-P89C51RD2 is populated with a 32 kByte SRAM device. The ***Memory Model*** should be set to *Large: variables in XDATA.*

- Select the ***Output*** tabsheet and enable the *Browse Information* checkbox. This will include additional browser information into the object files that can be viewed with the ***Source Browser*** included in Keil µVision2.

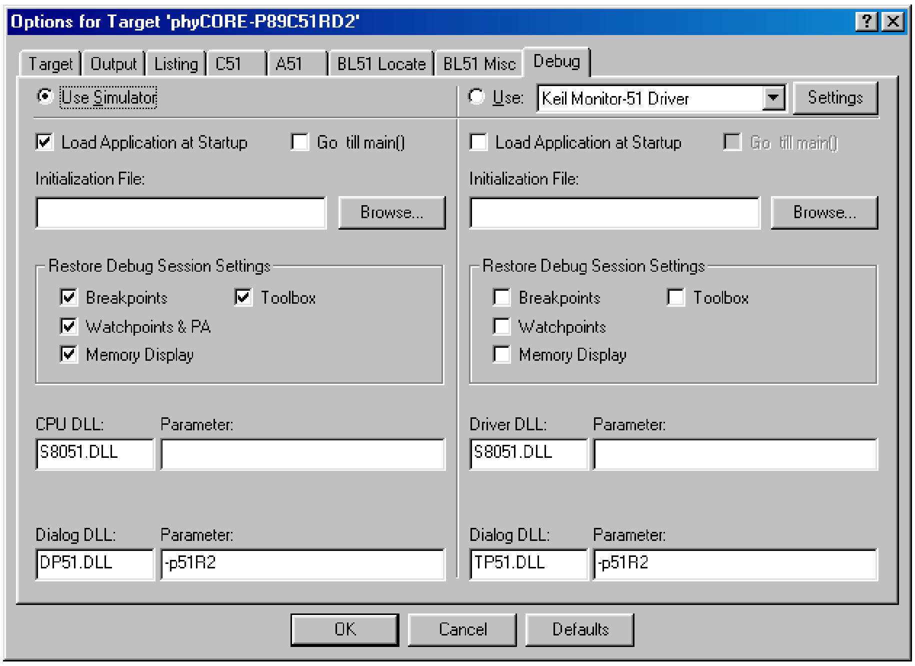


- Ensure that the configurations on the ***Listing***, ***C51***, ***A51***, ***BL51 Locate*** and ***BL51 Misc*** tabsheets are set to their default settings.
- Click the *OK* button to save the settings.
- Click on the **Rebuild all target files**  button to compile and link your project.

4.2 Preparing the Simulator

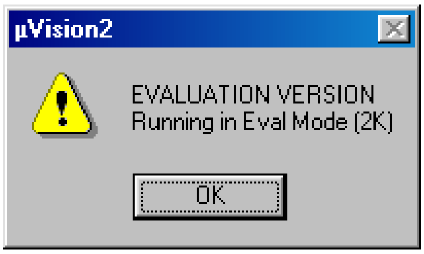
According to the ***Project|Options for Target 'phyCORE-P89C51RD2'*** configuration, µVision2 will load the application program and run the startup code.

- Open the ***Project|Options for Target 'phyCORE-P89C51RD2'*** menu and select the ***Debug*** tabsheet.
- Enable the checkboxes *Use: Simulator* and *Load Application at Startup*. Disable the checkbox *Go till main()*.



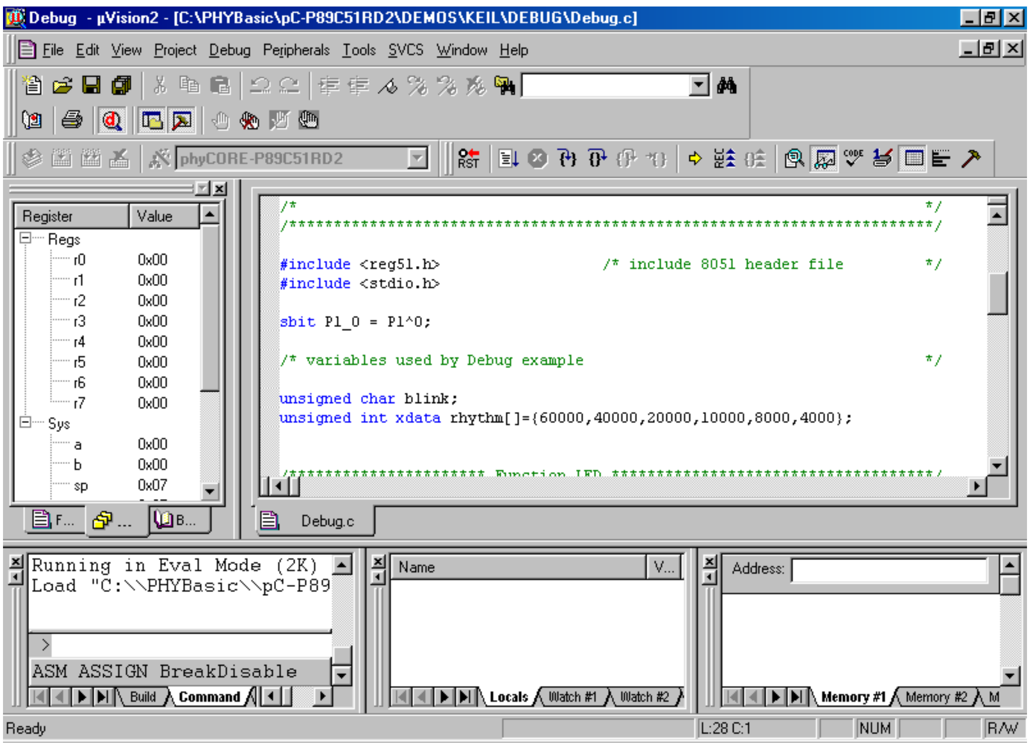
4.3 Starting the Simulator

- To start the µVision2 Simulator, click on the debugger icon  on the µVision2 toolbar. A pop-up window will appear indicating that this is an evaluation version. Click on *OK*.



- You will see a blue status bar from left to right at the bottom of your screen indicating the loading process of the debug program.

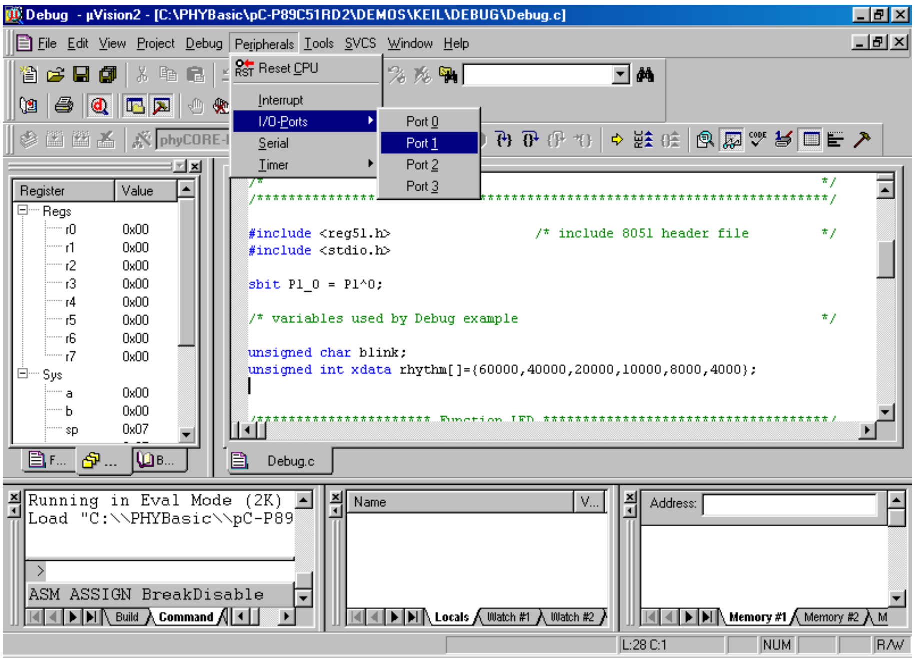
Once loaded, a screen similar to the one shown below will appear. The ***Project*** window is automatically changed to the ***Register*** page, showing the controller's internal registers. The simulator toolbar is also displayed. In the lower part of the debug screen you will see the ***Command***, ***Memory*** and ***Watch/Stack*** windows.



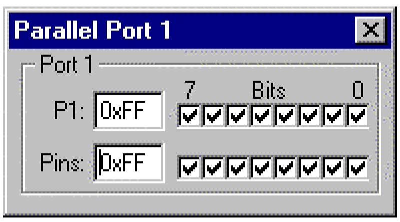
You may need to open, resize and /or move some windows to make your screen look similar to the screen capture. You can open inactive windows by choosing the desired window from the ***View*** pull-down menu. The following screen capture has *Workbook Mode* enabled to allow easy access to various overlapped windows with tabs.

The ***debug.c*** program demonstrates manipulation of an output port as already shown in the Blinky example. Port pin P1.0 of the P89C51RD2 controller controls LED D3 on the phyCORE Development Board LD 5V. The signal level change on port pin P1.0 can be viewed within the simulator environment using the *Port 1* window.

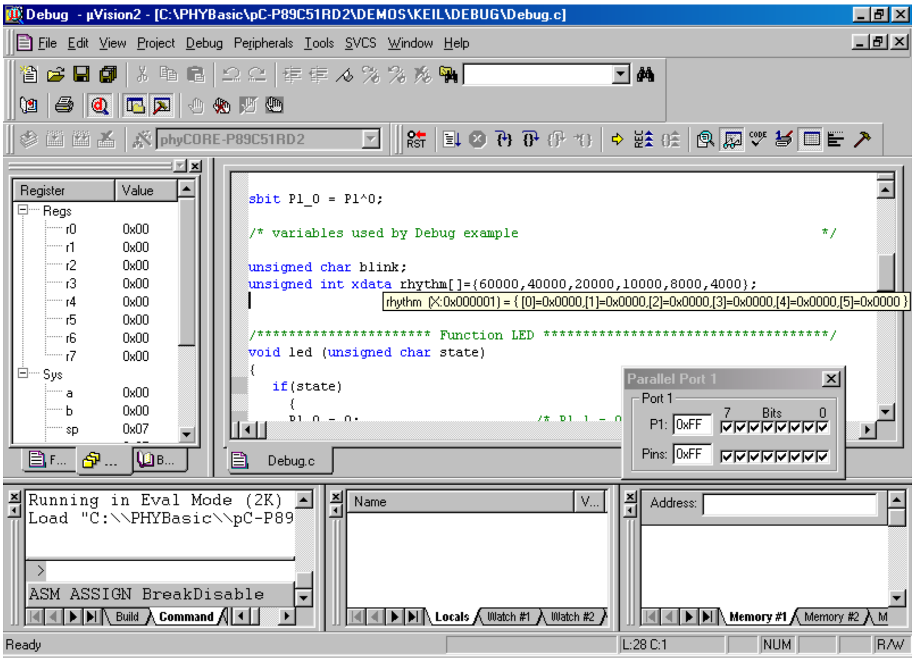
- Open this window by selecting ***Peripherals\I/O-Ports\Port1***.



- The ***Parallel Port 1*** window will now appear. Status of the simulated LED D3 control can be watched at bit position 0 (P1.0 = 0 => LED ON).



- Before starting the program, point the mouse at the *constant* named "**rhythm[]**". A small window appears and displays the address where *rhythm* is stored and its current value is shown. The *constant* is not yet initialized and the value is therefore random.

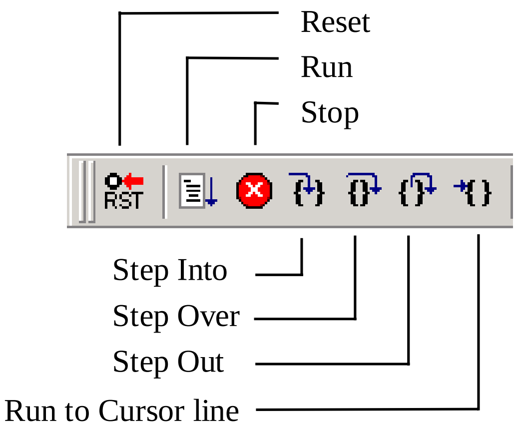


- In the ***Command*** window prompt in the lower left corner, type in ***g,main*** (notice the comma!) and press <Enter> to start the program.

The debugger will run to the '*main*' function and stop automatically. Notice the yellow arrow pointing to the first command in the '*main*' function. Also notice the program counter (**PC $**) within the **Project Window – Register** page showing the start address of the '*main*' function.

4.4 The Keil µVision2 Debug Features

- The ***Simulator*** window toolbar gives access to the following debug commands: *Reset*, *Run*, *Stop*, *Step Into*, *Step Over*, *Step Out* and *Run to Cursor line*.



- The first button on the simulator toolbar is the *Reset*  button.

  The *Reset* command sets the program counter to 0. However, it should be noted that peripherals and SFRs of 8051 devices are not set into reset state. Therefore this command is not identical to a hardware reset of the CPU.

- The button to the right of the *Reset* button starts the *Run*  command.

  Clicking this button runs the program without active debug functions. To stop program execution at a desired point, a breakpoint can be placed before the *Run* button is pushed.

- The next button on the simulator toolbar is the *Stop*  button.

  The *Stop* button interrupts and stops the running program at an undetermined location.

- The first button allowing exact control of the program execution is the *Step Into* button.

  The *Step Into* command performs the execution of the command line to which the *Current-Statement Arrow* points. This can be a C command line or a single assembler line, depending on the current display mode. If the command line is a function call, *Step Into* jumps to the C function or subroutine, enabling you to explore the code contained in the accessed subroutine.

  Note:
  During compiling of the C code, the compiler runs through various optimization steps. This can effect a *Step Into* command by executing a function call in a later command line than expected.

- The *Step Over* button is next on the debugger toolbar.

  The *Step Over* command executes the command line, to which the *Current-Statement Arrow* points. This can be a C command line or a single assembler line, depending on the current display mode. If the command line is a function call, the function will be executed without single stepping into the function.

  Note:
  We recommend not using the *Step Over* function with the simulator. During compiling of the C code, the compiler runs through the optimization steps. This optimization summarizes calls of nested functions and their return jump to the original function in the compiled machine code. This can effect a *Step Over* command by nesting multiple calls for sequential functions. For users with assembler knowledge, the ***View->Disassembly*** window is recommended. Here the C source code contrasts with the machine code and the optimization steps are obvious.

- The next button is the *Step Out button* .

  Step Out is used to exit a function you are currently in. *Step Out* is very useful if you find yourself in a function you are not interested in and need to return quickly to your intended function.

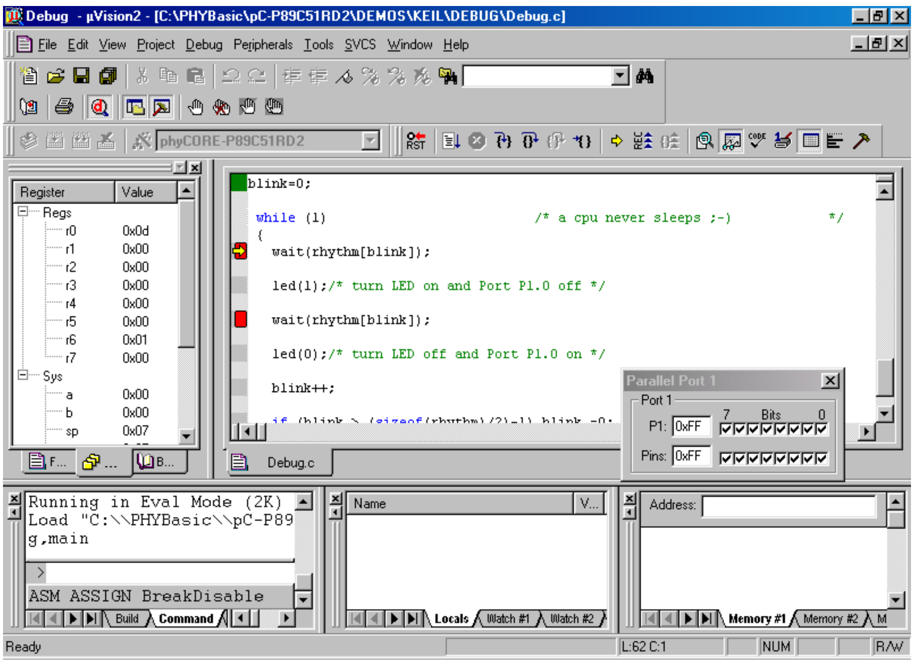
- The last button on the simulator toolbar performs the *Run to Cursor line* command.

  The *Run to Cursor line* command executes the program to the current cursor position within the code window. This allows use of the cursor line as a temporary breakpoint.

4.5 Using the Keil µVision2 Debug Features

4.5.1 Breakpoints

- Click in the line where the *'wait'* function is called for the first time.
- Click on *Insert/Remove Breakpoint* to set a breakpoint here. The red marker on the left-hand side of the selected line indicates the breakpoint.
- Insert a second breakpoint in the line where the *'wait'* function is called for the second time.

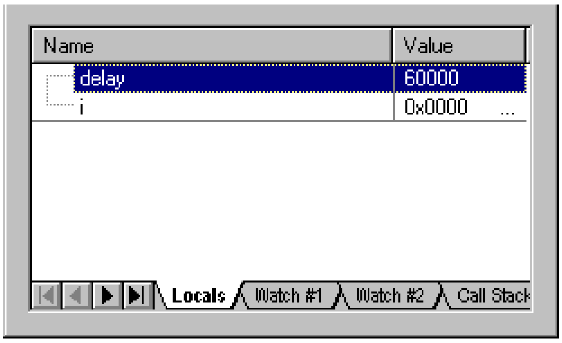


- Click on the *Run* icon and the program will run and stop at the first breakpoint.

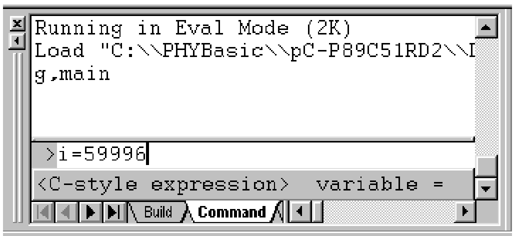
- Take a look at the ***Parallel Port 1*** window. Port pin P1.0 has not yet been modified in the demo program, hence bit position 0 remains on (P1.0 = 1 => LED OFF).
- Click again on the *Run* icon. The program will now stop at the second breakpoint as indicated by the yellow arrow pointing to the second *'wait'* function call.
- Notice the change of bit position ***0*** in the ***Parallel Port 1*** window (P1.0 = 0 => LED ON).
- Click on *Kill all Breakpoints* to remove both breakpoints.

4.5.2 Single Stepping and Watch Window

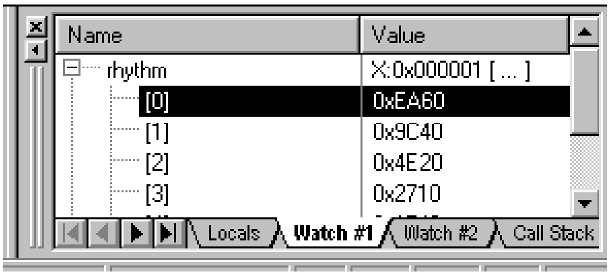
- Click on the ***Step Into*** icon to enter the *'wait'* function.
- The ***Watch*** window automatically shows the value of the two local variables; *delay* and *i*. In order to change the number base from hexadecimal to decimal, right-click on the variable you want to change.



- Click ***Step Into*** several times and watch the value of *i* count up.
- As you can see in the source code, the for{} loop will end if *i* becomes equal to *delay*. To leave the *wait* function, change the value of *i* by typing ***i=59996*** in the command line and pressing <Enter>. Now repeat clicking on **Step *Into*** until you leave the wait function.



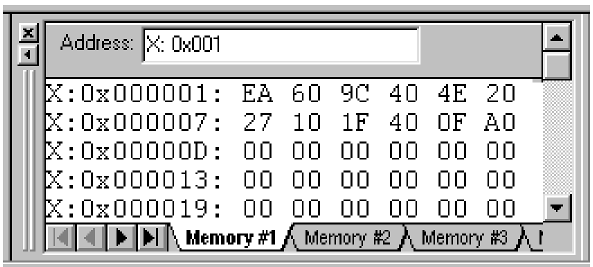
- Click in the source code line *blink++* and choose *Run to Cursor line* from the simulator toolbar. Your program will be executed until it reaches this line.
- Notice that bit position ***0*** in the ***Parallel Port 1*** window is set to "1" (P1.0 = 1 => LED OFF).
- As a last example, the constant "**rhythm[]**" will be evaluated. Go to the source code line where the constant "**rhythm[]**" is declared. Right-click on "**rhythm[]**" and choose the **"ADD rhythm to watch window" -> #1** option. Select the "**Watch #1**" tabsheet at the bottom of the watch window. The constant is shown with its address and a small ⊞ sign in front which indicates that "**rhythm[]**" is an array with a group of array elements. Click the ⊞ sign to expand the view and to see all array elements of "**rhythm[]**".



Using the memory address of *rhythm*, which is shown in the watch window, it is also possible to view the values of the array elements in the ***Memory*** window**.**

- Type X: 0x001 in the *Address*: line of the ***Memory*** window and press <Enter>.

A memory area starting at address 0x001 is now displayed in the ***Memory*** window**.**



- The passing of the array element to the wait function can be viewed whenever you step into the wait function. After stepping into the wait function you will find one of the elements in registers R6 and R7.

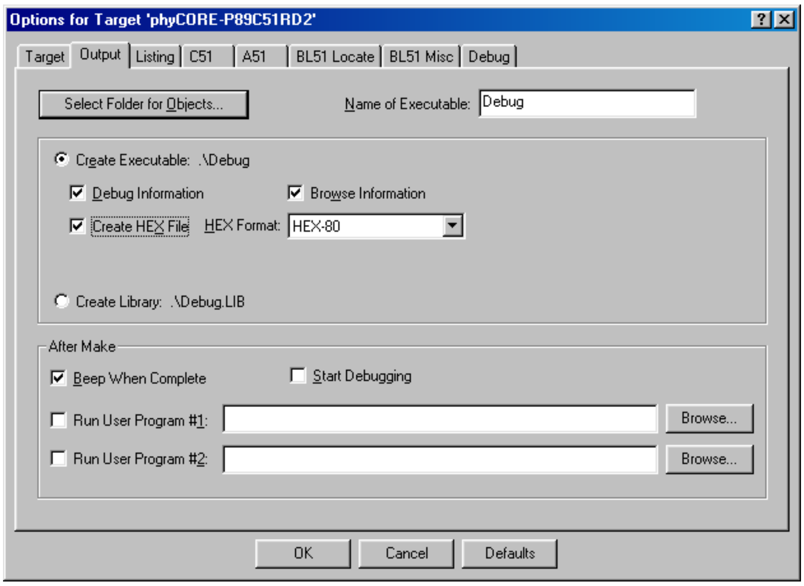
4.5.3 Running, Stopping and Resetting

- To run your program without stopping at any time, delete all breakpoints by clicking on the button.
- Click the *Run* button .
- Using the *Stop* button allows the user to stop execution of the running program at any time within the simulator environment.
- The Reset button can be used to reset the application program to its initial state. This renders the program into the same state as when starting the simulator (*refer to section 4.3*.)

4.6 Changing Target Settings for the "Final Version"

After successfully debugging the program, next change the target settings in order to create an Intel hexfile. This can then be downloaded to the on-chip Flash of the phyCORE-P89C51RD2.

- Select the ***Output*** tabsheet and enable the checkbox *Create HEX File*.



- Click on the *OK* button to save the settings.
- Click on the *Rebuild all target files*  icon to compile and link your project.
- Download the created ***Debug.hex*** file (located in ***C:\PHYBasic\pC-P89C51RD2\Demos\Keil\Debug***) to the on-chip Flash memory. *For general download procedure information refer to sections 2.2 through 2.4.*
- Press the Reset button S2 on the Development Board and watch your final debug example execute.

5 Advanced User Information

This section provides advanced information for successful operation of the phyCORE-P89C51RD2 in conjunction with the Keil tool chain.

5.1 FlashTools for On-Chip Flash

Flash is a highly functional means of storing non-volatile data. One of its advantages, among many others, is the possibility of on-board progamming. Programming tools for the on-chip Flash memory are included with the phyCORE-P89C51RD2. This includes the on-chip Flash with a Boot Loader and a counterpart PC-based software serving as the user interface. Once the Boot Loader communicates with the PC-based software, FlashTools for on-chip Flash allows download of user code from a host-PC into the on-chip Flash. Additionally, the re-programmable Flash memory on the phyCORE-P89C51RD2 allows you to easily update your own code and the target application in which the phyCORE-P89C51RD2 has been implemented.

For more information on how to render the phyCORE-P89C51RD2 into Flash programming mode *refer to the applicable section of the phyCORE-P89C51Rx2 Hardware Manual*.

5.2 STARTUP.A51

The code within the assembly file ***Startup.a51*** initializes the target hardware for your C project. This includes setting of the system stack, initialization of variables and clearing of memory areas.

The ***Startup.a51*** routines always start at code memory address 0x0000. This is where the reset vector with the jump instruction to the actual ***Startup.a51*** initialization functions is located. Following a hardware or software reset, the microcontroller starts execution at address 0x0000 and then jumps to the start-up routines location configured by the reset vector. After performing the initialization steps, your individual *main()* function is called by the startup code.

To configure the start-up code to fit the needs of your application, copy it from the ***Lib*** folder of the Keil tool chain to your project folder. You can then edit, modify and compile it using the Keil macroassembler.

You must explicitly instruct the linker to take into account your start-up object file. If not, the start-up code in the default runtime libraries will be used. To do this, we recommend adding your modified ***Startup.a51*** file to your project. Ensure that this file is always included in the Link/Lib process of your project (see options within the ***Project*** window of the Keil tool chain).

We recommend that you add ***Startup.a51*** or ***Startup.obj*** (depending on the kind of file you want to add to the project) within the project tree.

5.3 Linking and Location

The linker must combine several relocateable object modules contained in object files and/or libraries to generate a single absolute object.

In addition, the linker must locate several segments of code and data to fixed address locations within the address range in regards to the memory types of the microcontroller. XDATA segments always must be located to Random Access Memory (e.g. RAM), CODE segments should be located in non-volatile memory (e.g. Flash). The 8051 family supports a Harvard memory architecture that distinguishes between non-volatile and randomly accessible memory and has two physically different signals for separate fetching of data and code.

The Keil tool chain distinguishes the following segment types:

- CODE: code
- XDATA: external data (max. 64 kByte)
- DATA: direct addressable on-chip data (max. 128 Byte)
- IDATA: indirect addressable on-chip data (max. 256 Byte)
- BIT: bit-addressable on-chip data (max. 128-bits)

The segment types DATA, IDATA and BIT always reside in the on-chip RAM of the controller. CODE resides in the on-chip Flash of the P89C51RD2 controller as no external Flash is available on the phyCORE module.

The segment type XDATA will usually reside in the external RAM device.

To ensure proper execution of your application it is required that all XDATA segments are located to the external RAM of the phyCORE-P89C51RD2 and that all CODE segments are located to the on-chip Flash memory of the phyCORE-P89C51RD2.

The standard configuration of the phyCORE-P89C51RD2 is not populated with external Flash and instead offers 64 kByte of internal Flash on the P89C51RD2 controller.

The RAM will be addressable at 0x0000 to 0x06FF (internal XRAM) or 0x0000 to 0x7FFF (external RAM). This default runtime memory model actually requires no additional linker settings because both RAM and Flash start at 0x0000. This is also the default start address of the linkers segment types.

Since you cannot define any end address you should always ensure that the size of the segments fits within the available size of the mounted memory devices. For instance all XDATA segments should end below 0x06FF (or 0x7FFF if a 32 kByte external RAM is available). We recommend generation of a ***.m51** map file for your project and inspection of the memory map information within this file.

Document: **phyCORE-P89C51RD2 QuickStart Instructions**
Document number: L-465e_6, July 2002

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

Ordering No. L-465e_6
Printed in Germany