phyCORE-P89C51RD2

QuickStart Instructions

Using PHYTEC FlashToolsOCF for on-chip Flash and the Tasking 8051 Embedded Development Environment (EDE) Evaluation Version

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

	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

Index 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 the Tasking 8051 Embedded Development Environment (EDE) evaluation version, and

- instructions on how to run example programs on the phyCORE-P89C51RD2, mounted on the PHYTEC phyCORE Development Board LD 5V, in conjunction with the Tasking EDE

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 four example programs enabling quick out-of-the box start-up of the phyCORE-P89C51RD2 in conjunction with the Tasking 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 four example programs enabling quick out-of-the box start-up of the phyCORE-P89C51RD2 in conjunction with the Tasking 8051 software development tools. It is structured as follows:

1) The "*Getting Started*" section uses two example programs:

- *"Blinky"* and *"Hello"* to demonstrate the download of user code to the Flash device 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 Tasking tool chain and FlashTools OCF.

3) The *"Debugging"* section provides a third example program - "Debug" - to demonstrate simple debug functions using the Tasking 8051 EDE debug 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 the 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

TASKING
Embedded software development from *Altium*

http://www.tasking.com
tasking.support.na@altium.com (North America)
tasking.support.de@altium.com (Germany)

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, 1 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

- 1 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 sockets (P1A, P1B) configurable as RS-232 interfaces

- two additional DB-9 plugs (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 Tasking 8051 Embedded Development Environment (EDE)

The Tasking tool chain fully supports the entire 8051-derivative microcontroller family, including the Philips P89C51Rx2 family. It includes a C compiler, macroassembler, linker/locator and the CrossView Pro simulator and ROM monitor debugger within the EDE development environment.

Note:
The Tasking tool chain supports all in-circuit emulators that adhere to the IEEE-695 debugging specification or support the Tasking a.out format. The IHEX51 object-to-hex converter converts an absolute object file into an Intel hexfile that is suitable for downloading into the on-chip Flash on the PHYTEC phyCORE-P89C51RD2 target board.

The Tasking tool chain consists of the following executables:

- **C Compiler**   cc51.exe
- **Macro Preprocessor**   mpp51.exe
- **Assembler**   asm51.exe
- **Linker/Locator**   link51.exe
- **HEX converter**   ihex51.exe
- **Make Utility**   mk51.exe
- **IEEE-695 converter**   ieee51.exe
- **SREC converter**   srec51.exe
- **Object dumper**   dmp51.exe   (not in eval version)
- **Library archiver**   ar51.exe   (not in eval version)
- **CrossView Pro**   xfw51.exe   (Windows-based)
- **EDE**   ede.exe   (Windows-based)

Once installed, the default destination location for these files is the *C:\DCC51\bin* folder for the evaluation version. If using the professional (i.e. full) version of the Tasking tool chain, the default destination location for these files is *C:\CC51\bin*. The executables are 32-bit Windows programs that run in command line mode except for CrossView Pro and EDE which are Windows-based applications. Access to these programs from Windows is accomplished with EDE. The entire tool set can be run from EDE or directly from command line using batch or make files.

The Tasking tools are also available for UNIX.

Embedded Development Environment (EDE)

EDE 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. EDE runs under Windows 95/98/2000 and NT. This Embedded Development Environment has been expressly designed with the user in mind and includes a fully functional editor based on the popular Codewright editor from Starbase.

All EDE commands and functions are accessible via intuitive pull-down menus with prompted selections. An extensive Help utility and the complete set of online manuals is included. External executables can be run from within EDE, including emulator software.

8051 C Compiler

The 8051 ANSI C compiler and ASM51 assembler are designed specifically for 8051 controllers. All common derivatives from Philips, Infineon, Dallas, Intel and Atmel can be selected from a list in the EDE Processor Options menu.

The Tasking 8051 compiler provides the fastest and smallest code using industry benchmarks.

MPP51 Macro Preprocessor/ASM51 Assembler

The macro preprocessor and the assembler are started subsequently by the EDE environment. It is also possible to pass a file directly to the assembler when it does not need any macro expansion.

Debug Environment

CrossView Pro is a software simulator/ROM monitor debugger that supports debugging either via software on a host-PC or in target hardware. CrossView Pro enables the following debugging functions:

- run/halt, instruction/source stepping
- set (complex) breakpoints
- examine/change memory
- view the stack
- simulated I/O
- extended logging and scripting language
- powerful C like command language
- communication via RS-232

The CrossView Pro Simulator supports code coverage and profiling to ensure your code runs efficiently, and high level language trace.

The restrictions of the evaluation version of the Tasking 8051 tool set are described in '*C:\dcc51\doc\limit51.hlp*'. Other than these restrictions, the evaluation tool chain functions exactly as the full version. The evaluation version does not have a starting address restriction and produces useful object code. This allows you to fully evaluate the features of Tasking products in conjunction with the PHYTEC Single Board Computer module. The full version has no restrictions and is fully ANSI compliant.

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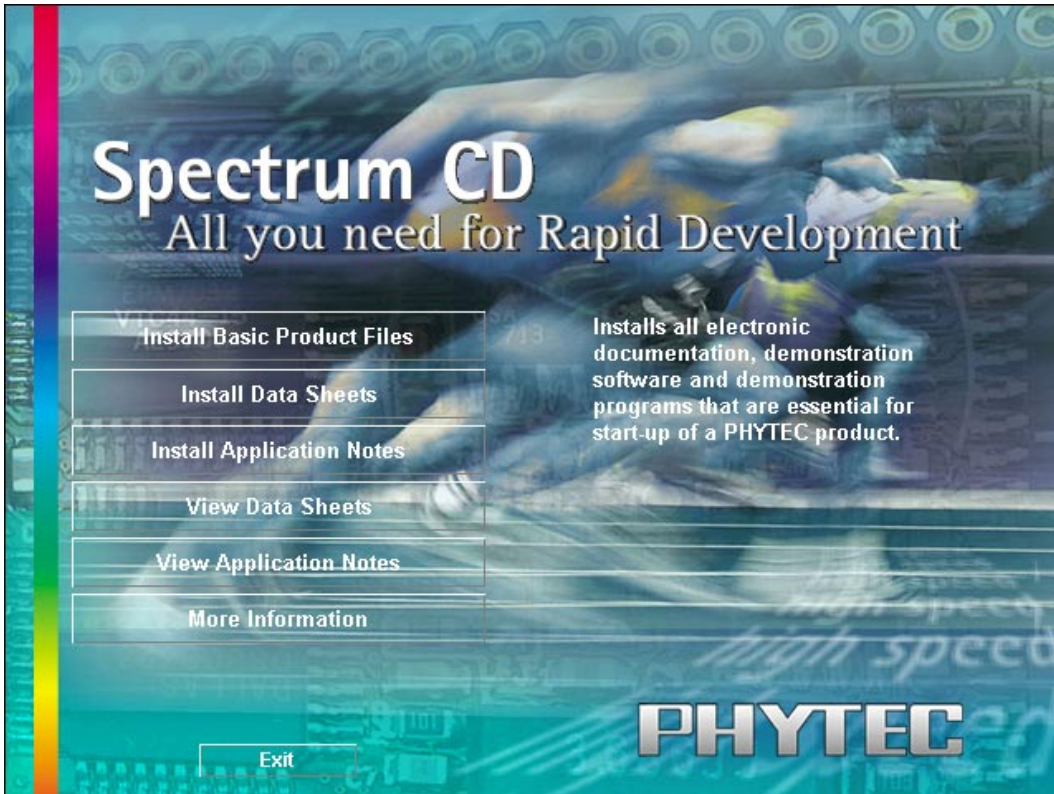
- Choose the *Install Basic Product Files* button.
- After accepting the Welcome window and license agreement, select the destination location for installation of Rapid Development Kit software and documentation.

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.
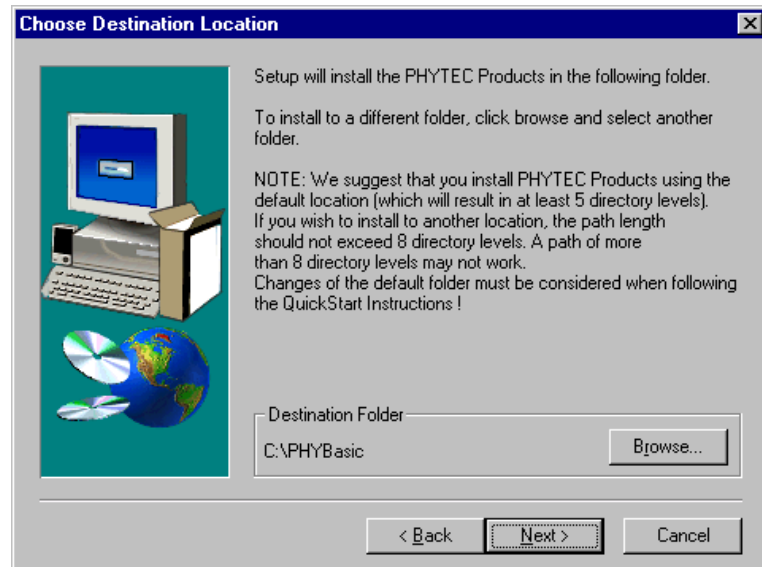
- In the next window, select your Rapid Development Kit of choice from the list of available products. By using the *Change* button, advanced users can select in detail which options should be installed for a specific product.
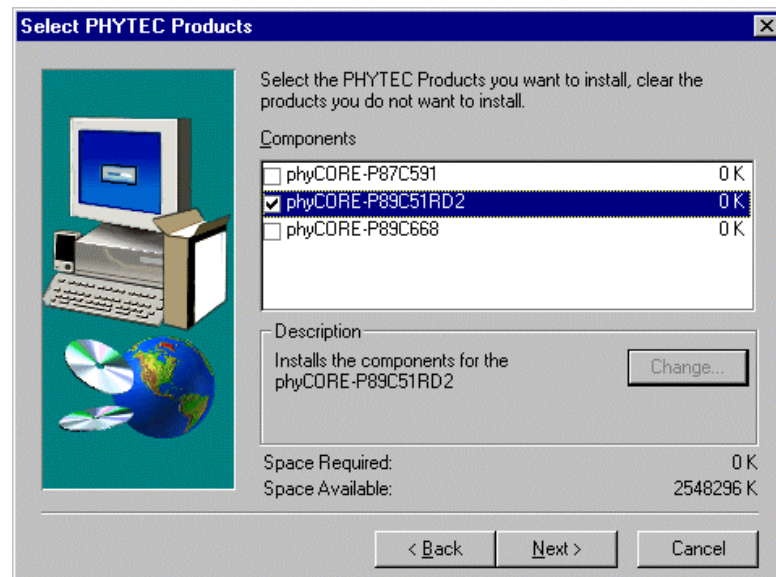
All Kit-specific content will be installed to a Kit-specific subfolder of the Rapid Development Kit root folder that you have specified at the beginning of the installation process.

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.
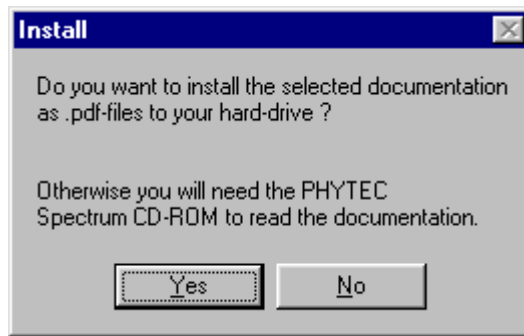

If you decide **not** to copy the documentation to your hard drive, you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.

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*.

- Press *Finish* to complete the installation of PHYTEC products.

In the next window, choose the Tasking for 8051 Software Development Tool Chain for installation[1].
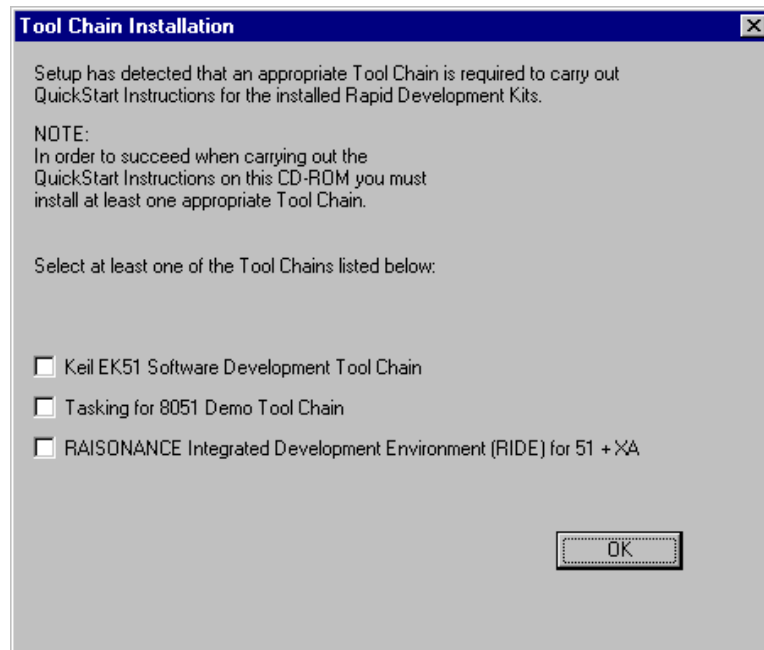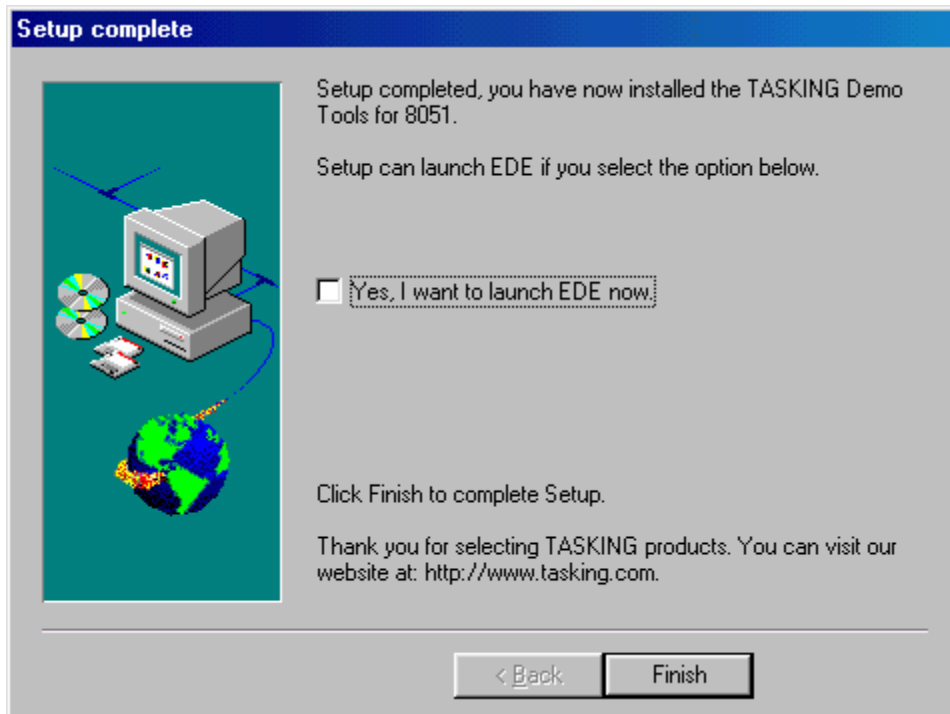

The applicable Tasking tool chain must be installed to ensure successful completion of this QuickStart Instruction. Failure to install the proper software could lead to possible version conflicts, resulting in functional problems.

We recommend that you install Tasking for 8051 from the Spectrum CD-ROM even if other versions of the Tasking tool chain 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 Tasking for 8051 demo tool chain. The default location is *C:\Dcc51*.

---

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

- Disable the check box *'Yes, I want to launch EDE now'*, the Tasking tool chain is not required for the first examples. Click on *Finish* to exit this installation step.

In the following windows you can decide to install FlashToolsOCF 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.
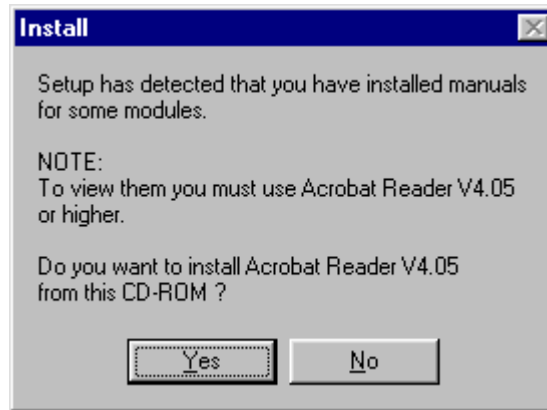
- Decide if you want to begin the QuickStart Instruction immediately by selecting the appropriate checkbox and press *Finish* to complete the installation.
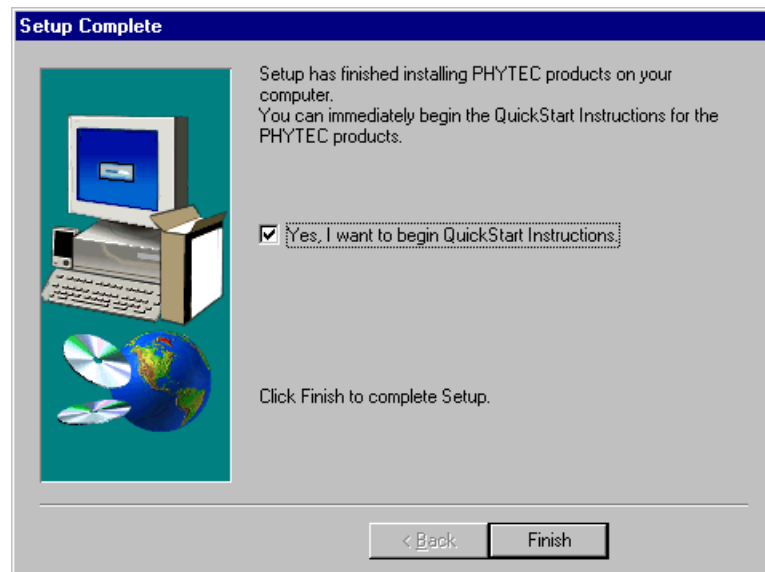
2.2 Interfacing the phyCORE-P89C51RD2 to a Host-PC

Connecting the phyCORE-P89C51RD2, mounted on the 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.
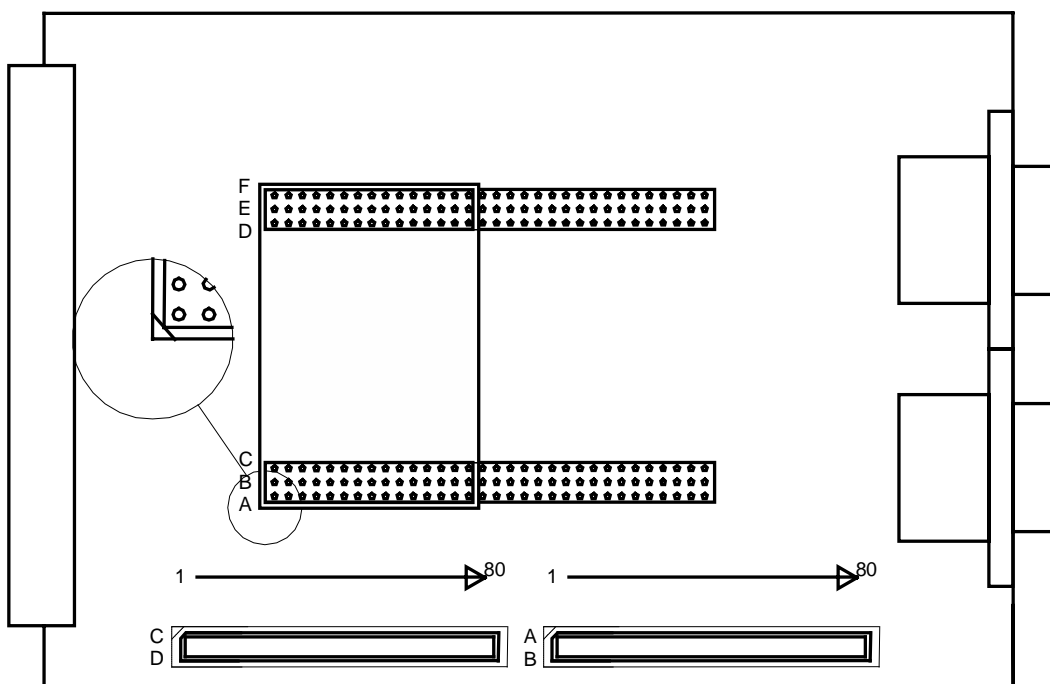

Figure 1:   Mounting the phyCORE-P89C51RD2 onto the phyCORE Development Board LD 5V

- 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.
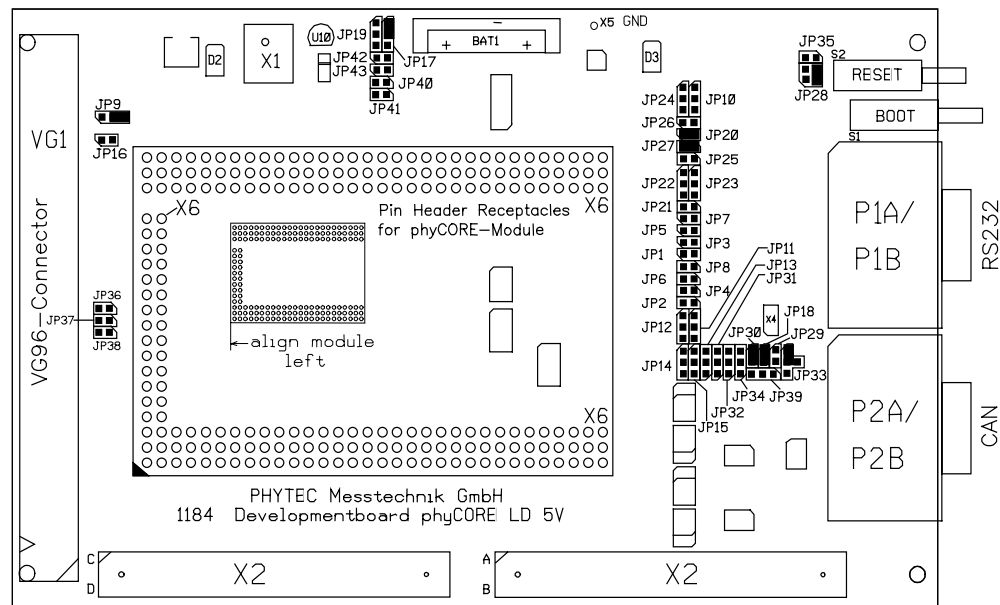

Figure 2:   Important Connectors, Buttons and Suitable Jumper Settings on the phyCORE Development Board LD 5V

- 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 3 for the correct polarity*).
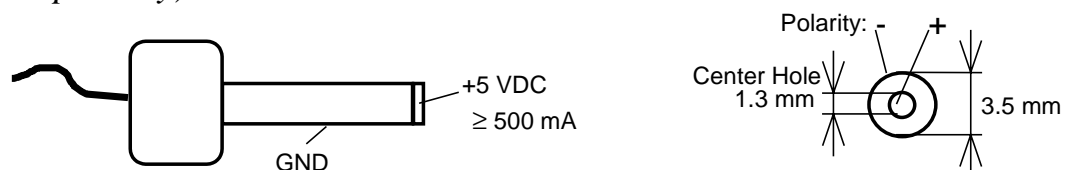

Figure 3:   Power Connector

- 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 into 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 FlashTools"* for more details.

The phyCORE module 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 (OCF) should have been installed during the initial setup procedure as described in *section 2.1*. If not, you can manually install it using the ***setup.exe*** file located in the folder ***\Software\FTocF***.

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 the **Programs/Phytec** program group.

- The Communication Setup tab of the FlashTools for on-chip Flash tabsheet window will now appear.
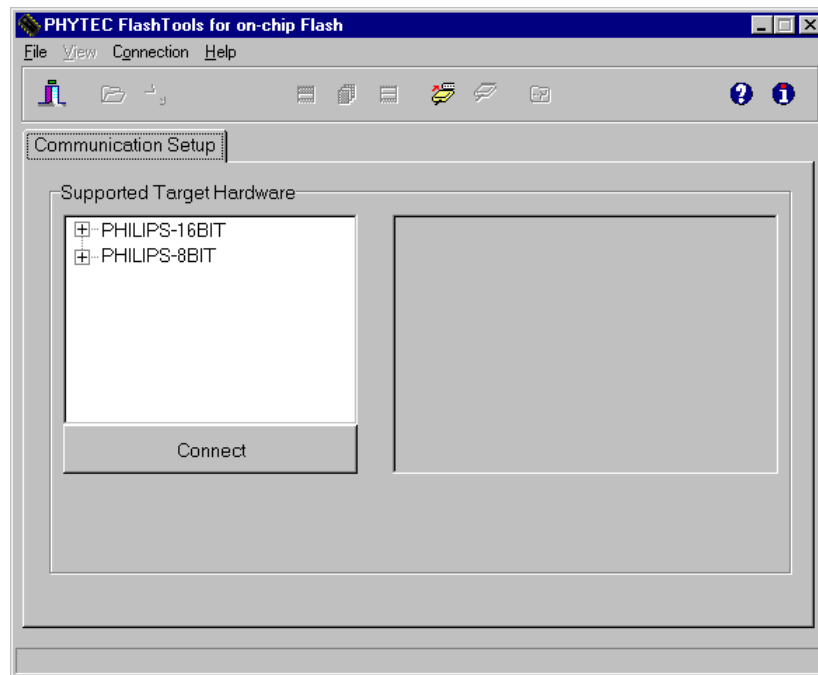
- Select the target hardware (*PHILIPS-8BIT*), the controller type (*P89CRx2*) and the oscillator frequency. To configure the correct oscillator frequency, identify the value of the oscillator located next to the controller on the phyCORE module. In most cases the value is *16 MHz*. If the oscillator has an uneven value, you can determine the appropriate frequency by rounding up to the next even value (e.g. 11.059 MHz $\rightarrow$ 12 MHz). The correct frequency must be indicated in order to properly start FlashToolsOCF.
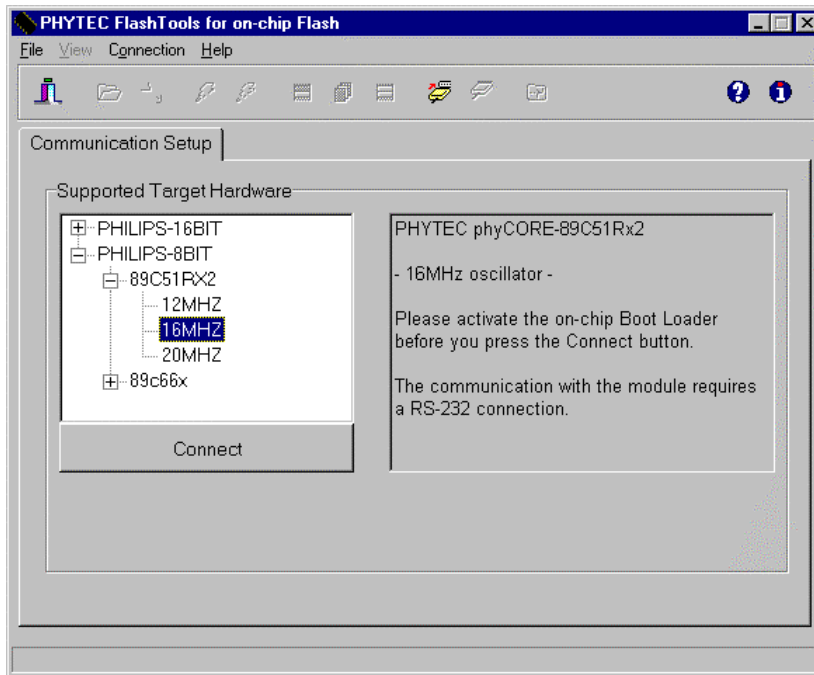
Note:
Always ensure that the phyCORE-P89C51RD2 is in Flash programming mode before pressing the *Connect* button (*refer to section 2.2*).

- 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:
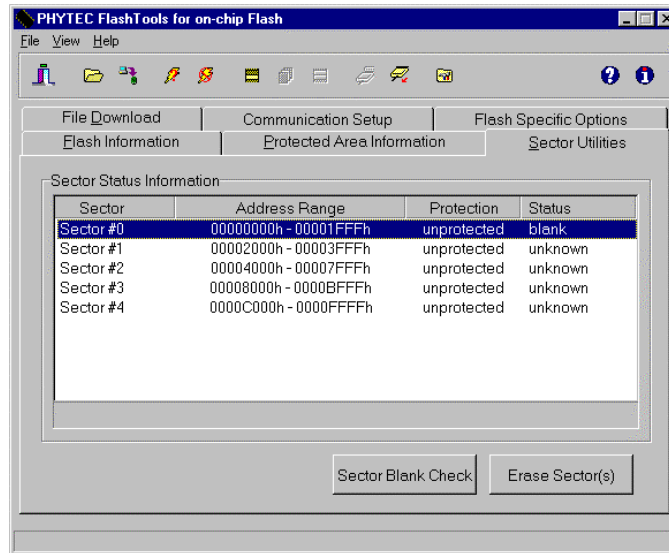
File Download downloads specified hexfiles to the target hardware:
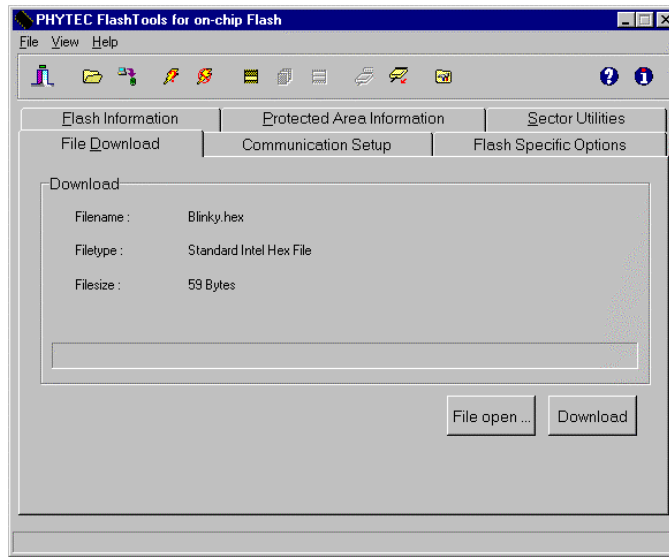
Flash Information shows Flash type, sector and address ranges in Flash memory:
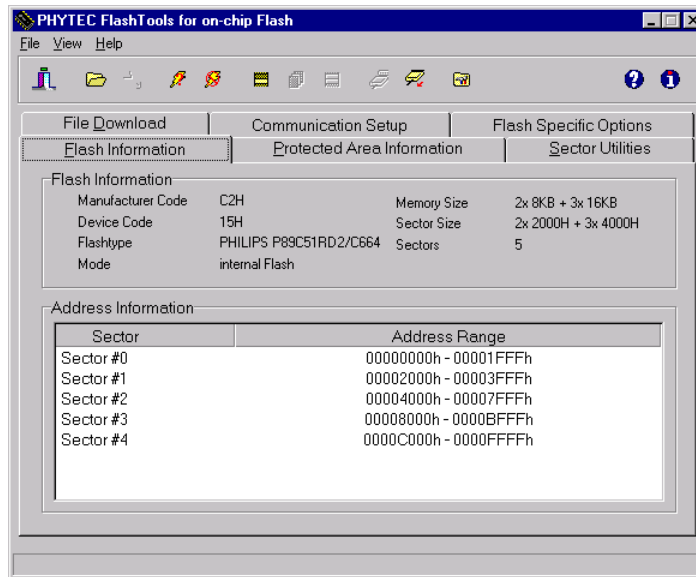
Communication Setup allows connection to and disconnection from the board (this is the same window that was used when you first entered FlashToolsOCF):

Protected Areas Information shows protected areas of Flash memory:

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

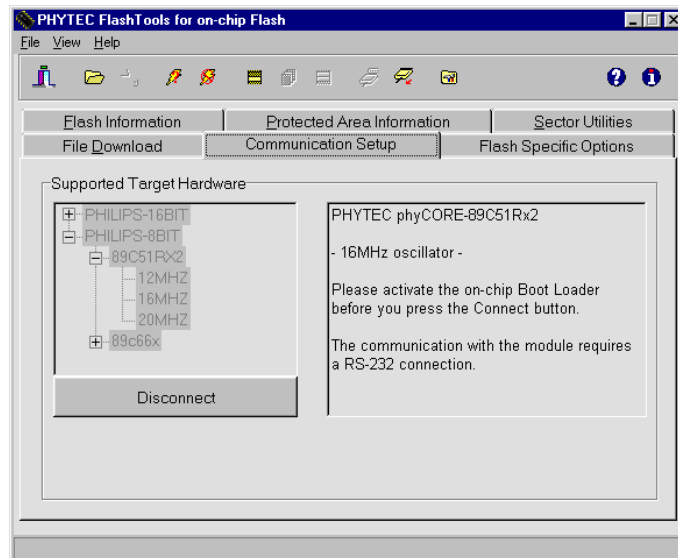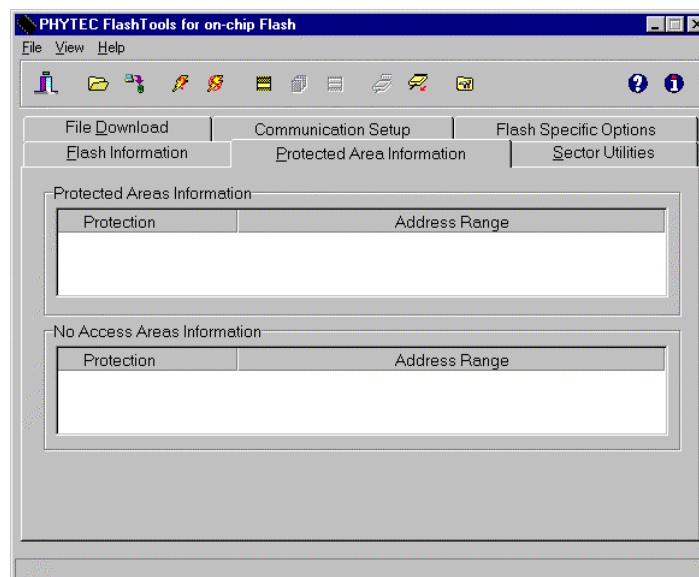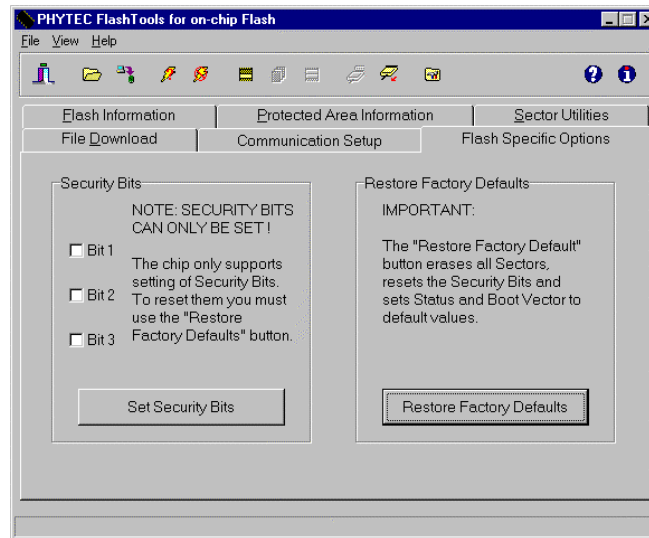

For the P89C51RD2 controller, the security bit functions are as follows:

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 2*):

- 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\Tasking\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.
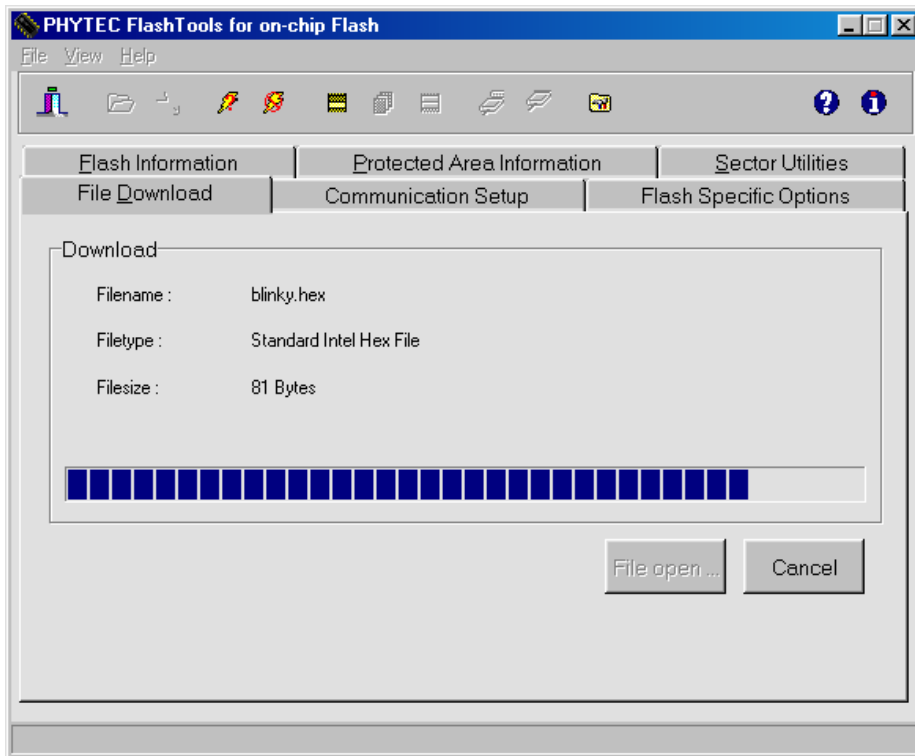
If the selected Flash sector into which you wish to download code is not empty (i.e. erased), a warning dialog box will appear, indicating "Location not empty! Please erase location and try again". In this event, select the *Sector Utilities* tab from the FlashToolsOCF tabsheet, highlight *Sector #0 - #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\Tasking\Hello\Hello.hex*) and click *Open.*
- Click on the *Download* button. You can watch the status of the download of **Hello.hex** into the on-chip Flash memory in the Download window.
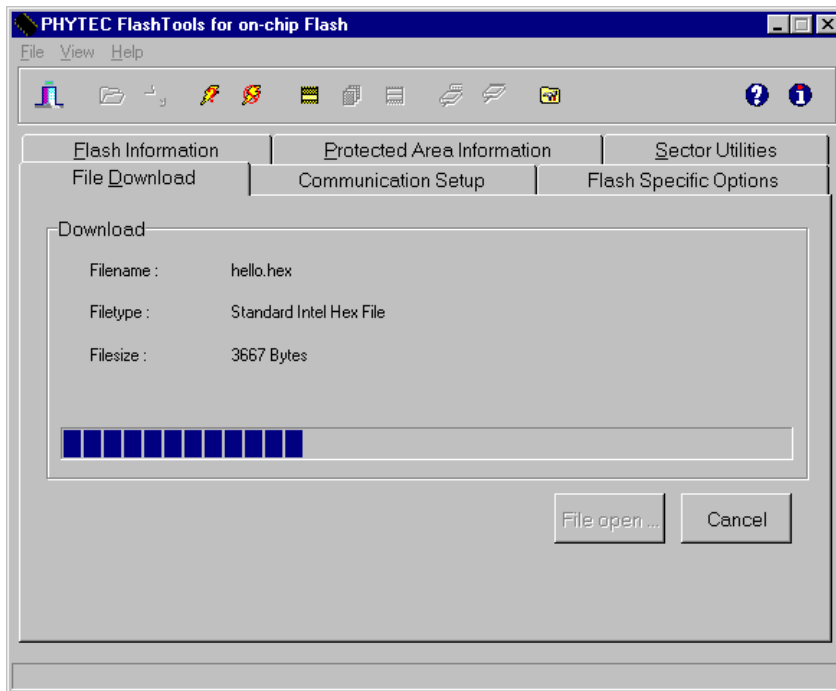
If the selected Flash Sector into which you wish to download code is not empty (i.e. erased), a warning dialog box will appear, indicating "Location not empty! Please erase location and try again". In this event, select the *Sector Utilities* tab from the FlashToolsOCF tabsheet, highlight *Sector #0 - #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.

---

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

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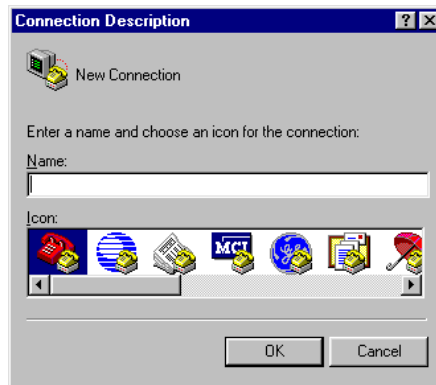

- Next click on *OK*. This creates a new HyperTerminal session named "COM Direct" and advances you to the next HyperTerminal window.
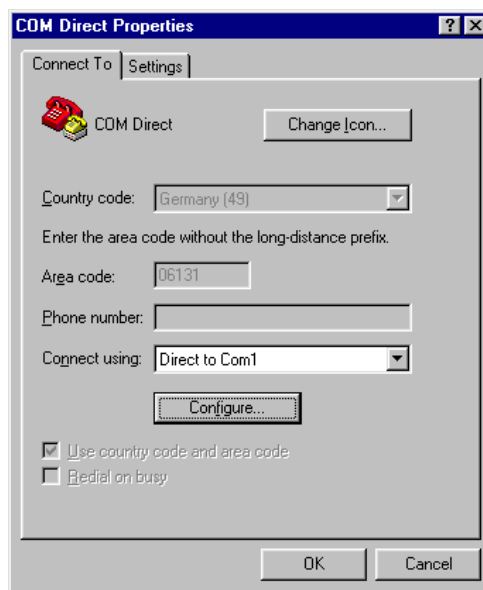
- The *COM Direct Properties* window will now appear. Specify *Direct to COM1/COM2* under the *Connect Using* pull-down menu (be sure to indicate the correct COM setting for your system).

- 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.
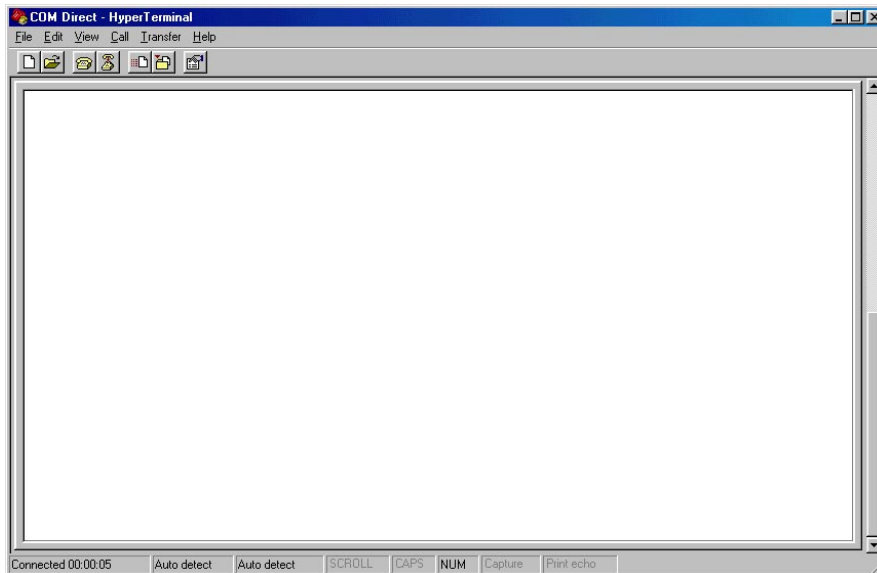
- Resetting the phyCORE Development Board LD 5V (at S2) will execute the ***Hello.hex*** file loaded into the Flash.
- Now push the <Space> bar on your keyboard once to start the automatic baud rate detection on phyCORE-P89C51RD2 module.
- Successful execution will send the character string "*Hello World*" 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 Tasking tool chain
- how to configure the Tasking tools within the Embedded Development Environment (EDE)
- 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 Tasking Tool Chain

The Tasking 8051 Embedded Development Environment (EDE) evaluation software should have been installed during the install procedure, as described in *section 2.1*.

You can also manually install the tool chain by executing ***setup.exe*** from within the ***\Software\Tasking\DCC51*** folder of your PHYTEC Spectrum CD.

Note:
It is necessary to use the Tasking 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 *EDE* from within the *Programs/TASKING 8051 Demo* program group.

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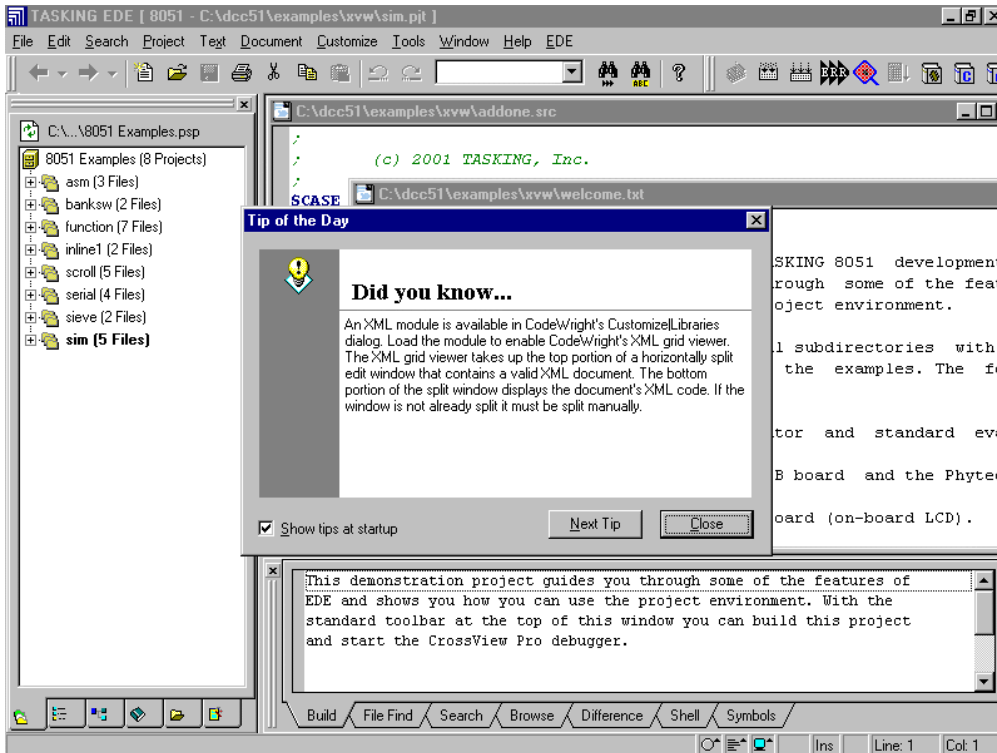

- You can close the **Tip of the Day** window.

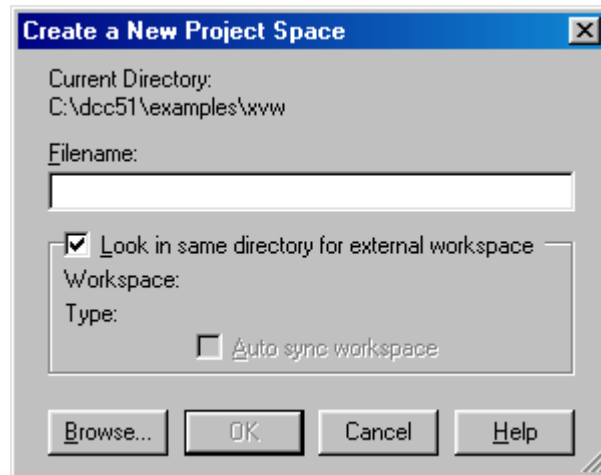## 3.2 Creating a New Project and Adding an Existing Source File

EDE 8051 automatically loads the most recently opened project. If you find an existing project when starting EDE 8051, close it by selecting the *Project* menu and *Project Space\Close* the project.
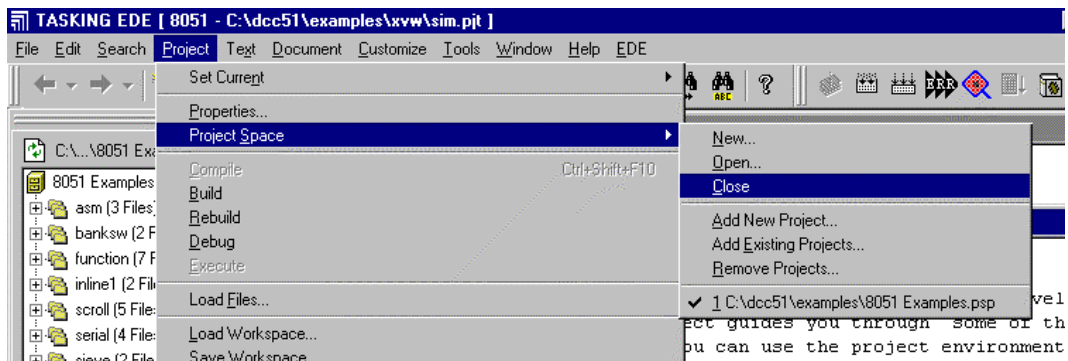
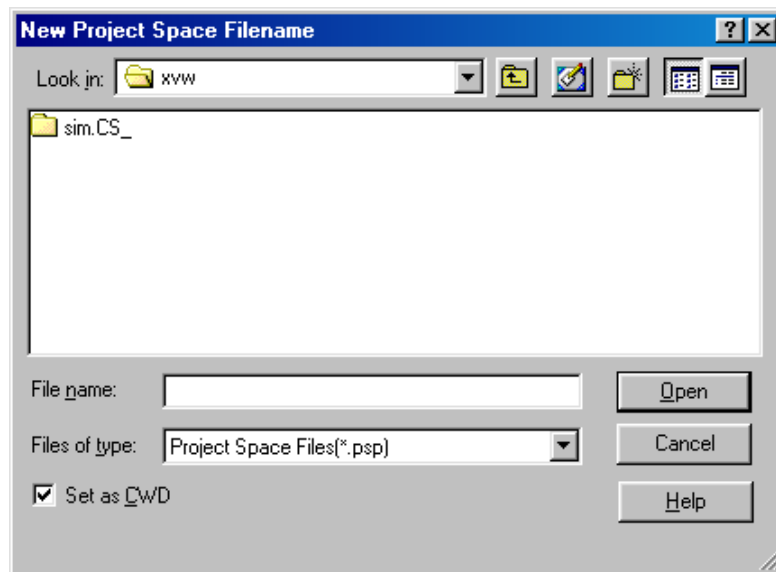- To create a new project file select from the EDE 8051 menu *Project/Project Space\New...*.
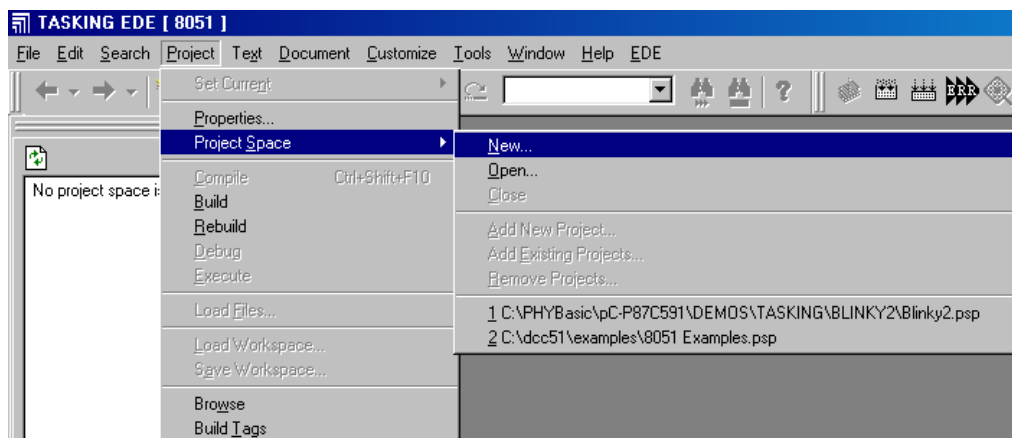
- The following window will appear:

- Click on the **Browse** button in the *Create a New Project Space* window.
- This opens the *New Project Space Filename* window as shown below:

- In the *Look in:* pull-down menu, change to the project directory created by the installation procedure (default location *C:\PHYBasic\pC-P89C51RD2\Demos\Tasking\Blinky2*).
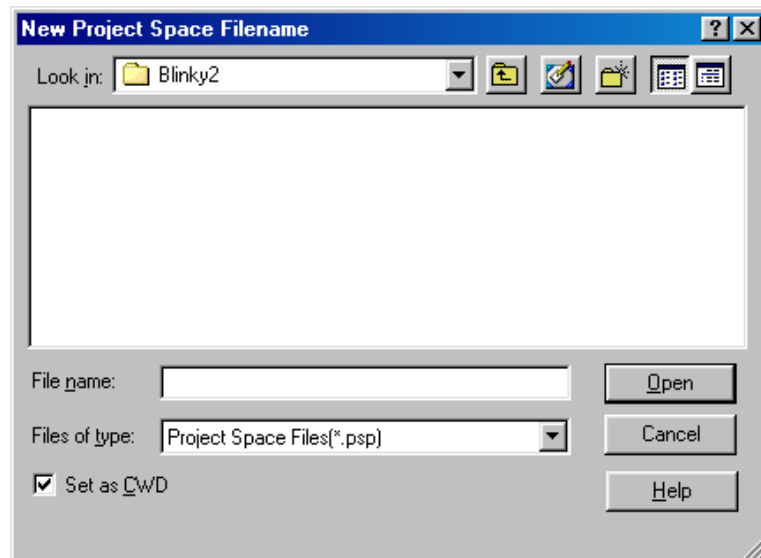
- In the text field '*File name*', enter the file name of the project space you are creating. For this example, enter the name *phyCORE-P89C51RD2* and click on *Open*.
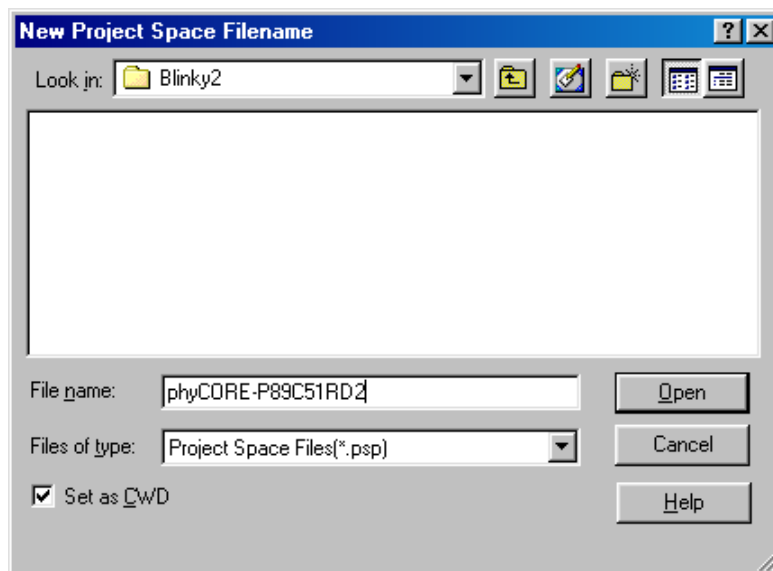
- Click on **OK** in the *Create a New Project Space* window.
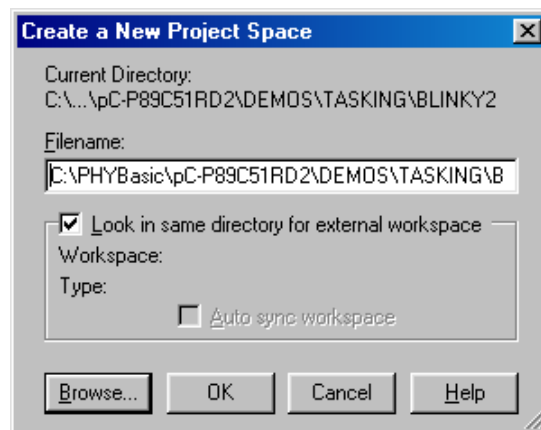
- The *Project Properties* window will now appear:
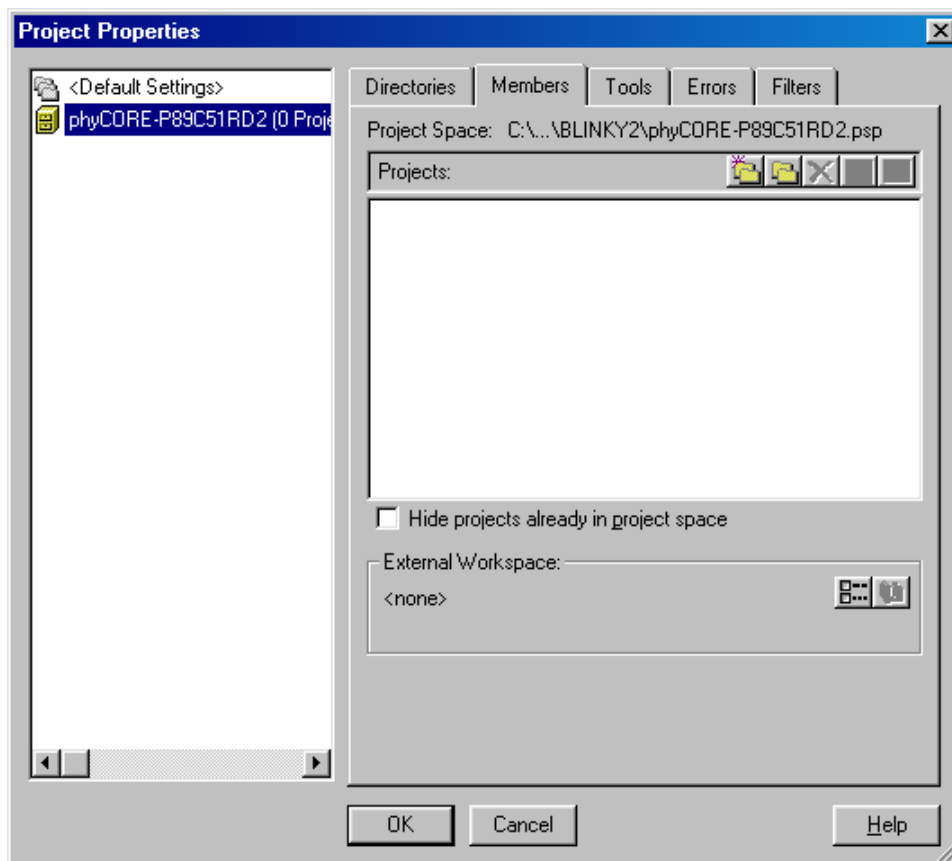
- Click on the *Add a new Project* [icon] icon. The *Add New Project to Project Space* window will now appear:
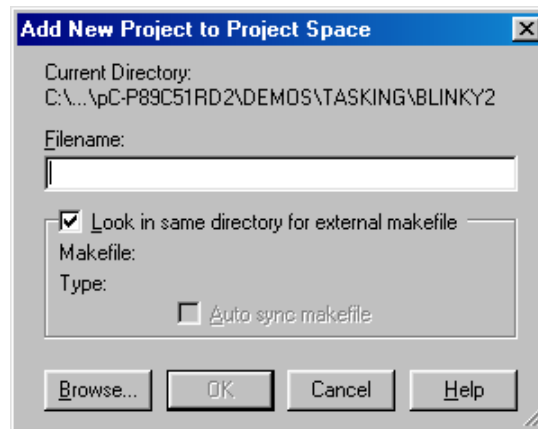
- In the text field '*Filename*' of the *Add New Project to Project Space* window, enter the file name of the project you are creating. For this example, enter the name **Blinky2** and click on **OK**.
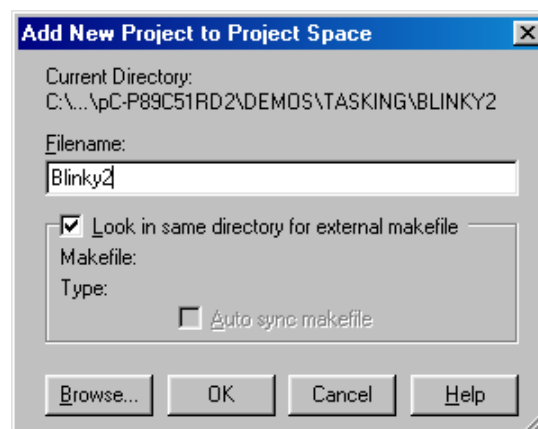
- The *Project Properties* window will now look like this:
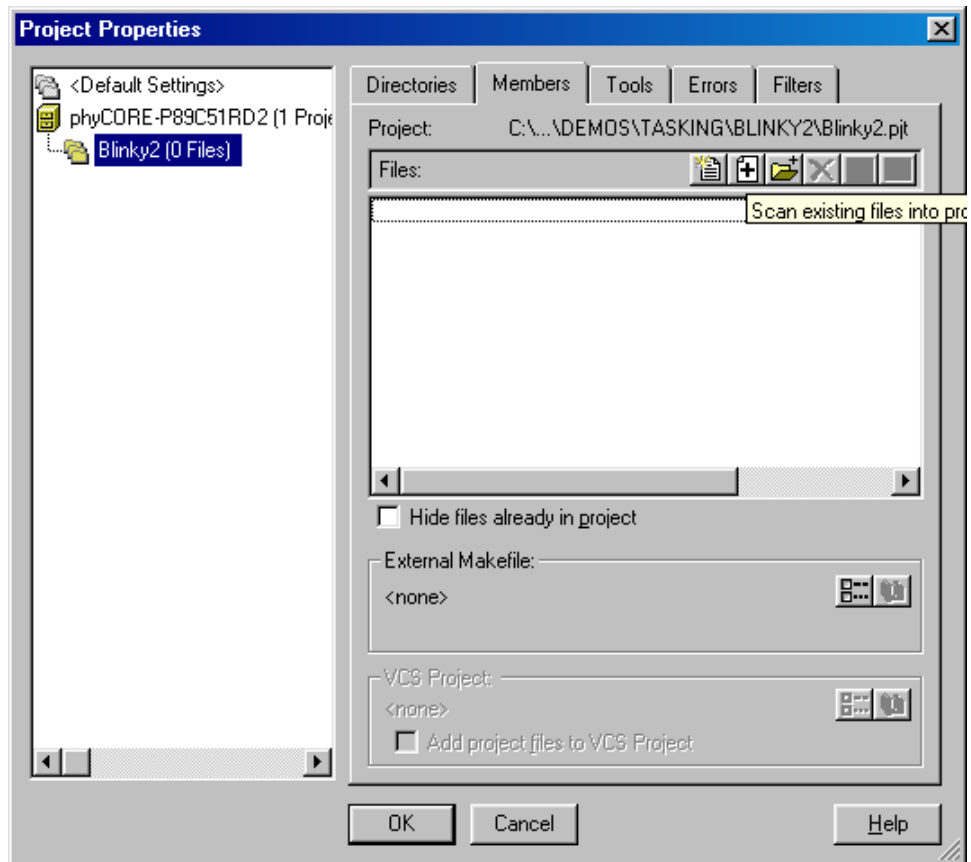
- Click on the *Scan existing files into project* icon.

- In the pull-down menu *Pattern,* select ***C/C++ Files*** as shown below:
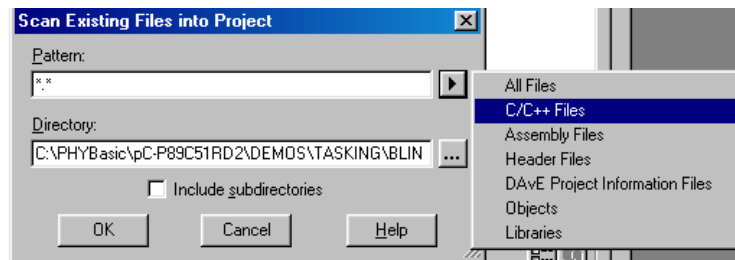
- The *Scan Existing Files into Project* window should now look like this:
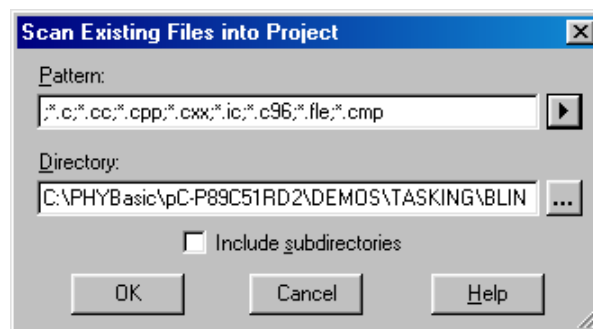
- Click on ***OK*** in the *Scan Existing Files into Project* window.

- The *Project Properties* window will now appear as follows:
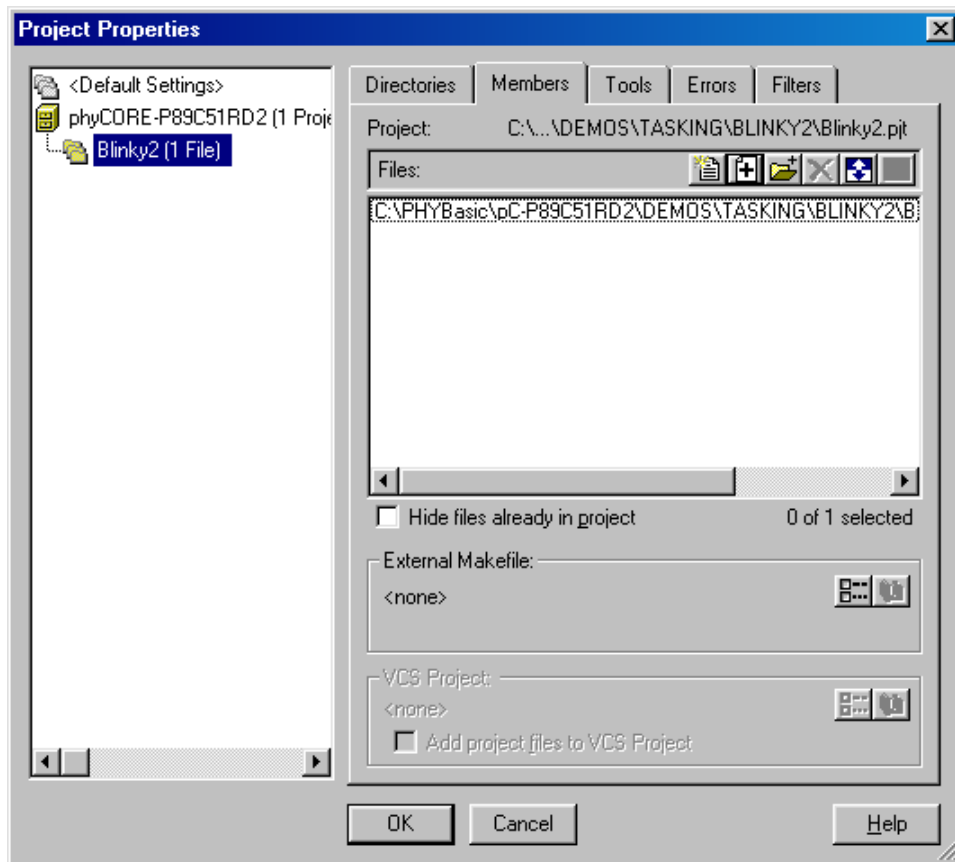
- Click on **OK** in the *Project Properties* window.

The EDE 8051 project window now shows the project space *phyCORE-P89C51RD2* with the *Blinky2* project including two source files. The group *Source Files* contains the C file *Blinky2.c* and the assembler file *blinky2_cstart.asm*.
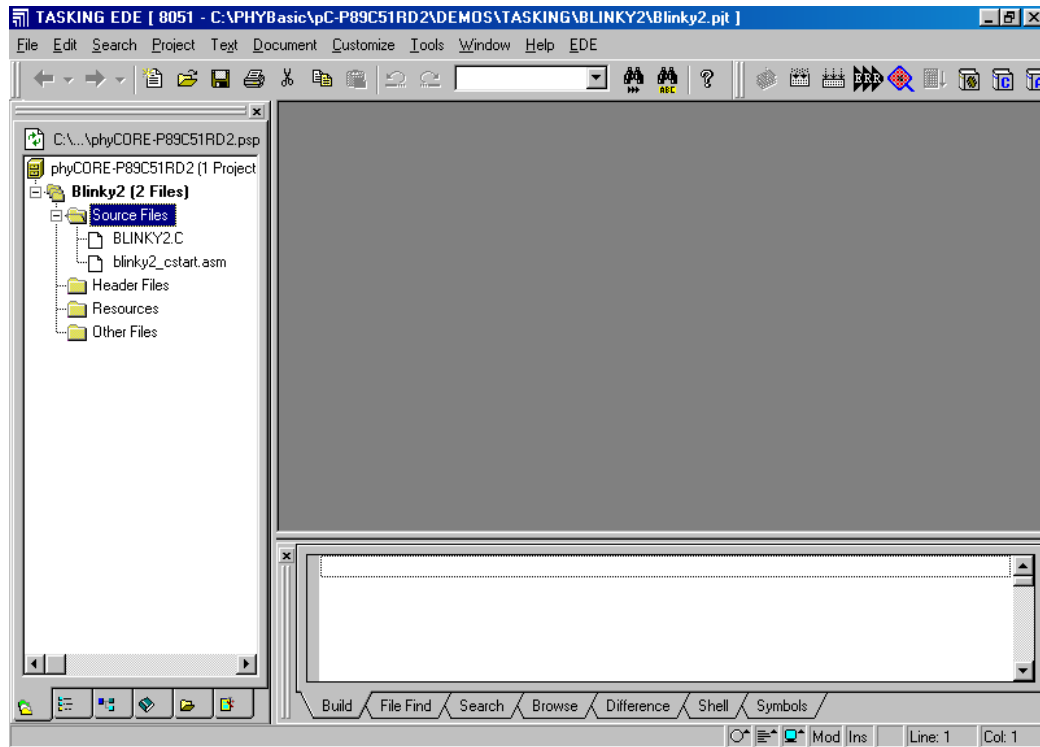
At this point you have created a project space called **phyCORE-P89C51RD2,** a project called *Blinky2* and added an existing C source file called *Blinky2.c*. In addition, the EDE 8051 automatically created an assembler file *blinky2_cstart.asm* that contains specific startup code.

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

3.2 Modifying the Source Code

- Double-click on *Blinky2.c* to open it in the source code editor.
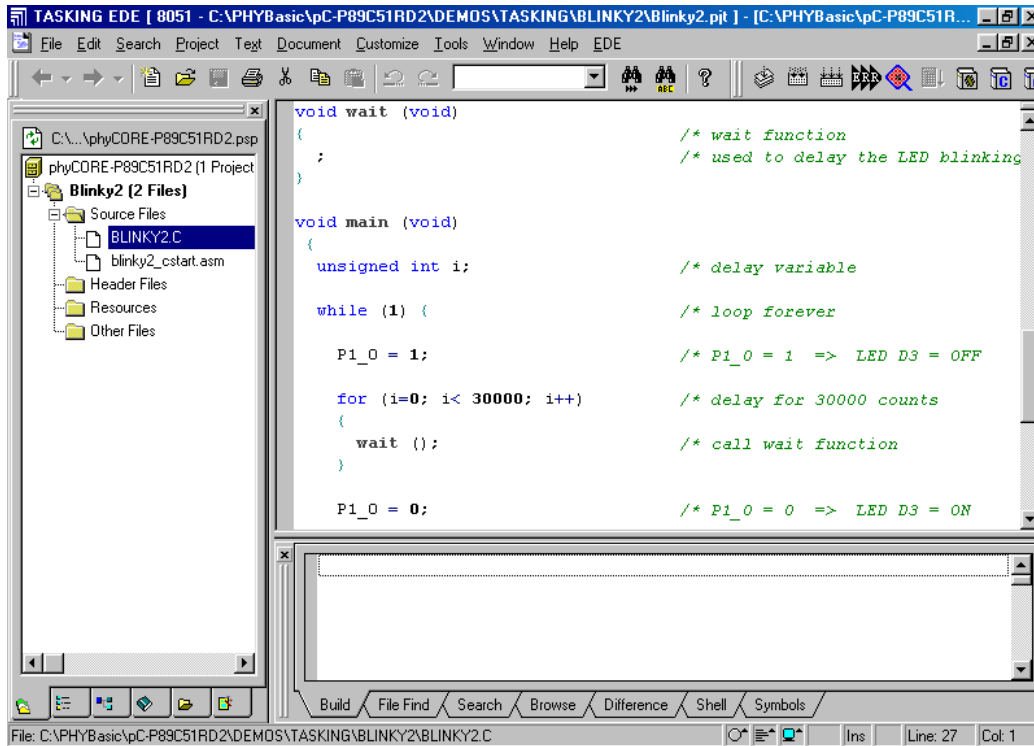
- Locate the following code section. Modify the section shown below (the values shown in bold and italic) from the original (30,000) counts to the indicated values:

```
while (1) {                     /* loop forever                 */
    P1_0 =  1;                  /* P1_0 = 1 LED D3 = OFF         */
    for (i=0; i< 30000; i++) /* delay for 30000 counts        */
    {
      wait ();                  /* call wait function           */
    }
    P1_0 =  0;                  /* P1_0 = 0 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.

3.3 Saving the Modifications

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

3.4 Setting Tool Chain Options

Tasking EDE for 8051 includes a Make utility that can control compiling and linking source files in several programming languages. Before using the macro preprocessor, assembler, C compiler or linker/locator you must configure the corresponding options. Enter the changes as indicated below and leave all other options set to their default values. EDE allows you to set various options with mouse clicks and these are all saved in your project file.

To configure the CPU:

- Open the ***EDE\Processor Options*** menu from the EDE menu bar.
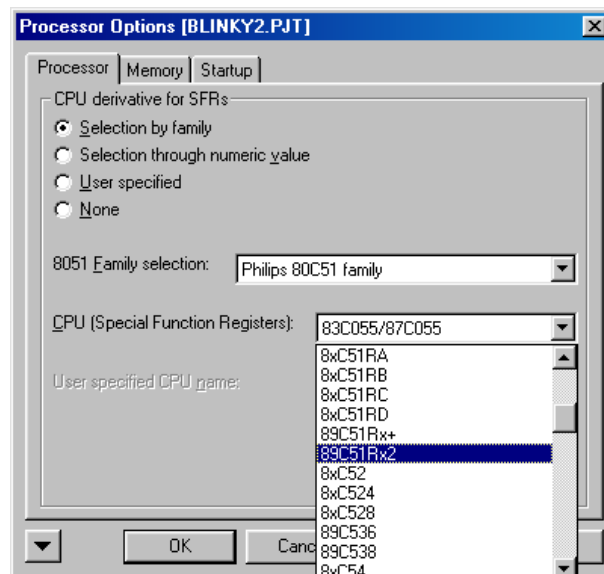
- Activate the radio button *Selection by family*. Choose the *Philips 80C51 family* in the *8051 Family selection* pull-down menu. In the pull-down menu *CPU*, select the *89C51Rx2[3]* controller and click on *OK* to save these settings.

---

[3] :   The P89C51RD2 is not available in the Philips CPU list of the current Tasking 8051 tool chain version. However, the P89C51Rx2 family includes the P89C51RD2 controller.

To configure the CC51 Compiler:

- Open the ***EDE\C Compiler Options\Project Options*** menu from the EDE menu bar.
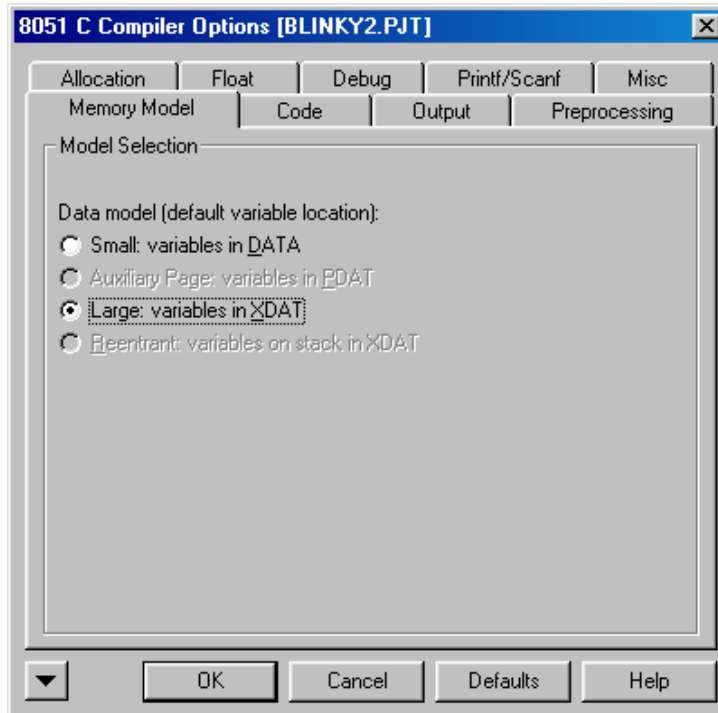
- Select the radio button *Large: variables in XDAT* within the ***Memory Model*** tab and click on ***OK*** to save these settings.

To configure the MPP51 Macro Preprocessor:

No modifications are necessary when default settings are used.

To configure the ASM51 Assembler:

No modifications are necessary when default settings are used.

To configure the LINK51 Linker:

- Open the *EDE\Linker Options* menu from the EDE menu bar.
- Activate the *Intel HEX records for EPROM programmers* checkbox.
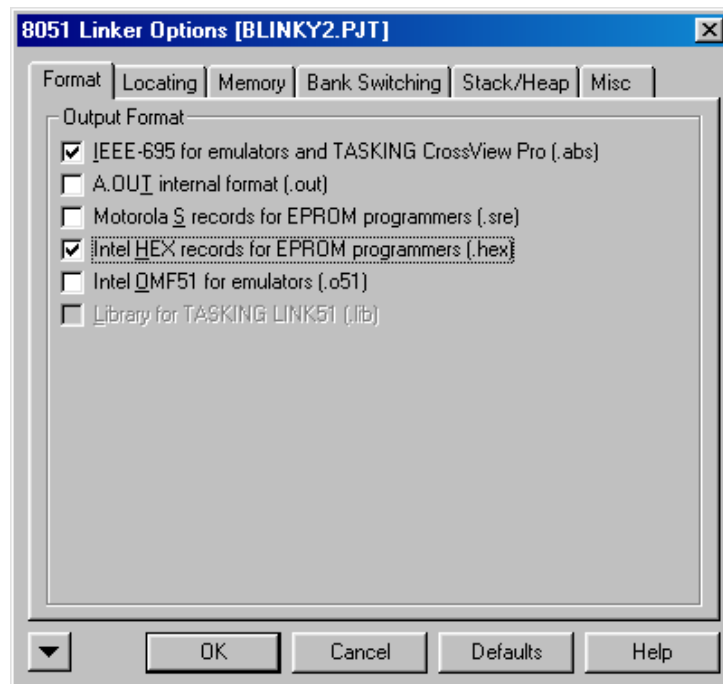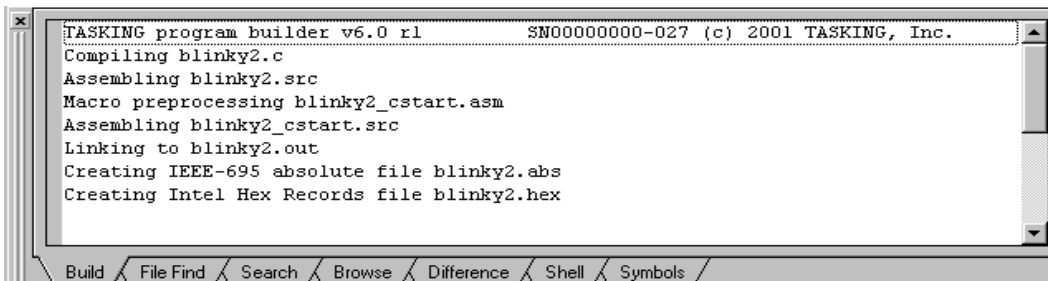
- Click on *OK* to save these settings.

3.5 Building the Project

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

- Click on the *Execute 'Rebuild' Command* 🎛 icon from the EDE tool bar or open the **Project** menu and select *Rebuild*. If the program specified (**Blinky2.c**) contains any errors, they will be shown in the **Output** window at the bottom of the screen.
- If there are no errors, the code is assembled and linked and the executable code is ready to be downloaded to the module. This is shown in the bottom line (status bar), which indicates *No errors found*. The created hexfile will have the name of the project with **.hex** as the filename extension (in this case **Blinky2.hex**).

```
TASKING program builder v6.0 rl        SN00000000-027 (c) 2001 TASKING, Inc.
Compiling blinky2.c
Assembling blinky2.src
Macro preprocessing blinky2_cstart.asm
Assembling blinky2_cstart.src
Linking to blinky2.out
Creating IEEE-695 absolute file blinky2.abs
Creating Intel Hex Records file blinky2.hex
```

```
Build \ File Find \ Search \ Browse \ Difference \ Shell \ Symbols
```

Note:
A machine-readable, executable hexfile has been created. Other files (e.g. assembler input files *.src*, list files *.lst* and map files *.l51*) 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 and (re-)build the project again.

If you received an environment memory error message during the make process you must change the properties of your **command.com** file. Go to **Windows\Command.com**, right-click on the file and open the properties window. In the memory tab you must change the initial environment to 4096.

3.6 Downloading the Output File

- Exit Tasking EDE for 8051.
- 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, a few seconds later, the Boot button.
- Start FlashTools4 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\Tasking\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.

3.7 "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 at section 3.1.*

3.7.1 Creating a New Project

- Start the Tasking EDE environment and close all projects that might be open.
- Open the **Project** menu and create a new project space called phyCORE-P89C51RD2 and a new project called **hello2** within the existing project folder
  C:\PHYBasic\pC-P89C51RD2\Tasking\Demos\Hello2
  (default location) on your hard-drive.
- Add **serinit.lib, serinit.h, _ioread.c, _iowrite.c** and **hello2.c** from within the project folder to the project **hello2** *(refer to section 3.2).*
- Your **Project** window should now look like this:
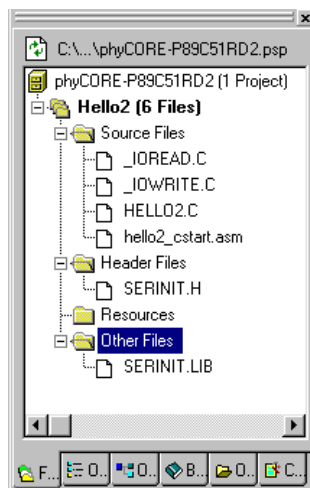
- *Save* the project.

At this point you have created a project called ***Hello2.pjt*** consisting of the following source and library files:

- ***Hello2.c*** – demo program in C
- ***Serinit.lib*** – library of functions for initialization of the serial port
- ***_ioread.c and _iowrite.c*** – modified functions of the standard Tasking library to activate the *printf* and *scanf* functions to allow use of the terminal program without the debugger
- ***Hello2_cstart.asm*** – assembler program to define specific module configurations (automatically created and updated by the EDE)

3.7.2 Modifying the Example Source

- Double-click on 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***.

3.7.3 Setting Tool Chain Options

- Change the tool chain options default settings to the correct values as shown in *section 3.4*. This includes settings for the CPU, the memory model (make sure "*Large: variables in XDAT*" is configured) and the linker options (*Intel HEX records for EPROM programmers* checkbox active).
- In the ***EDE/C Compiler Options\Project Options\Preprocessing*** tab, define the user macro **_NOSIMIO** (no simulated output).

This define allows the *printf* and *scanf* functions to be used by direct executable applications for communication with a terminal program. Otherwise the functions for serial communication would need a simulator/debugger.

3.7.4 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, save the file and (re-)build the project.

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

3.7.5 Downloading the Output File

- Exit Tasking EDE for 8051.
- 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
  C:\PHYBasic\pC-P89C51RD2\Demos\Tasking\Hello2\Hello2.hex
  directory (default location).
- 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.

3.7.6 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 Tasking EDE for 8051 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 Tasking*.

The Tasking EDE for 8051 Debugger offers two operating modes:

- The **CrossView Pro 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.

- The **CrossView Pro ROM Monitor** debugger allows for target hardware-based debugging. The debug environment on the PC communicates with the target hardware via a monitor kernel that is running on the target system. Debugging on the target hardware also enables testing peripheral components of the application.

The following examples utilize the CrossView Pro 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 CrossView Pro ROM Monitor is not possible. Programs can be evaluated before code is downloaded to the controller using the Tasking CrossView Pro Simulator.

4.1 Creating a Debug Project and Preparing the Debugger

4.1.1 Creating a New Project

- Start the Tasking EDE for 8051 environment and close all projects that might be open.
- Open the **Project** menu and create a new project space called **phyCORE-P89C51RD2** and a new project called **Debug** within the existing project directory
  C:\PHYBasic\pC-P89C51RD2\Demos\Tasking\Debug
  (default location) on your hard drive.
- Add **Debug.c** from within the project folder to the project **Debug** (*refer to section 3.2*).
- Your **Project** window should now look like this.
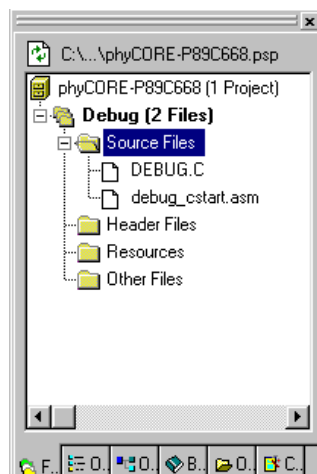
- *Save* the project.

- At this point you have created a project called **Debug.pjt** consisting of the following source files:

- **Debug.c** – debug program in C
- **Debug_cstart.asm** – assembler program to define specific module configurations (automatically created and updated by the EDE)

4.2 Setting Options for Target

To configure the CPU:

- Open the ***EDE/Processor Options...*** menu to configure the processor options for the ***Debug*** project.
- Activate the radio button *Selection by family*. Choose the *Philips 80C51 family* in the *8051 Family selection* pull-down menu. In the pull-down menu *CPU*, select the *89C51Rx2[4]* controller.
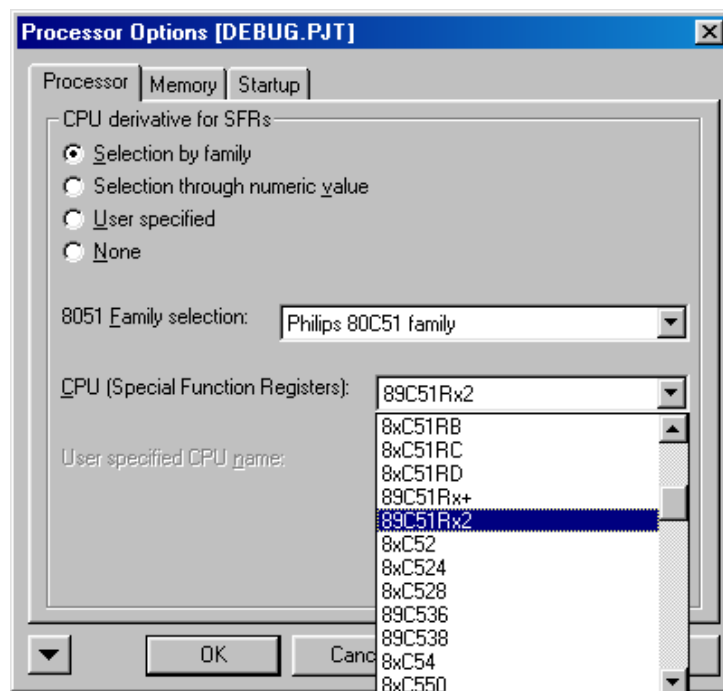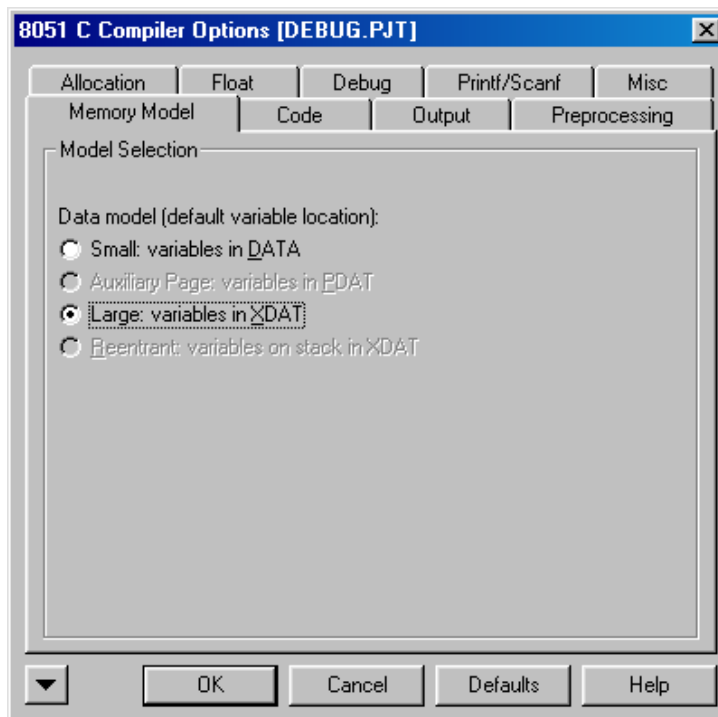

- Click on ***OK*** to save these settings.

---

[4] :   The P89C51RD2 is not available in the Philips CPU list of the current Tasking 8051 tool chain version. However, the P89C51Rx2 family includes the P89C51RD2 controller.

To configure the CC51 Compiler:

- Open the ***EDE\C Compiler Options\Project Options*** menu from the EDE menu bar.
- Select the radio button *Large: variables in XDAT* within the ***Memory Model*** tab.

- Click the ***OK*** button to save the settings

To configure the Safer C Compiler Options:

No modifications are necessary when default settings are used.

To configure the ASM51 Assembler:

No modifications are necessary when default settings are used.

To configure the LINK51 Linker:

- Open the *EDE\Linker Options...* menu from the EDE menu bar and select the *Format* tabsheet.
- Activate the *IEEE-695 for emulators and Tasking CrossView Pro (.abs)* checkbox.
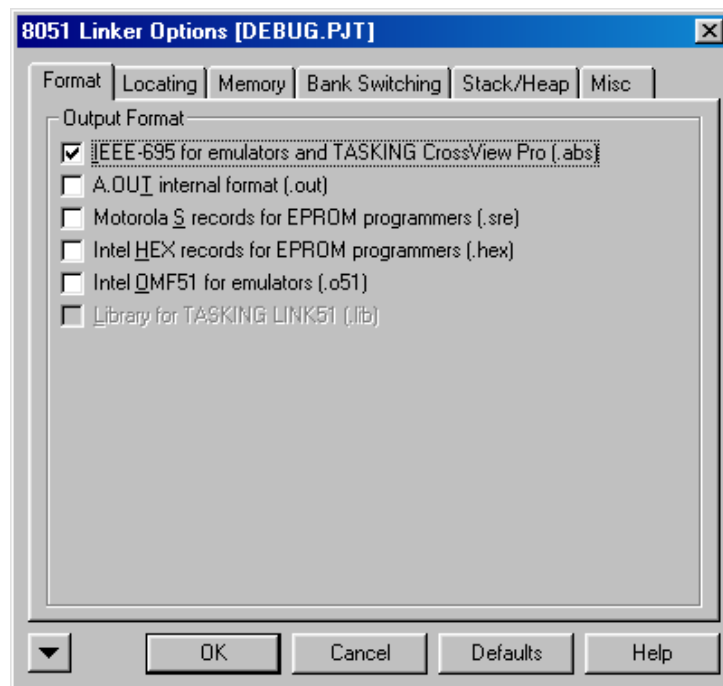
- Click the *OK* button to save the settings.

Build the Project:

- Click on the *Execute 'Rebuild' Command* [icon] icon at the EDE tool bar or open the *Project* menu and select *Rebuild*.
- 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, save the file and (re-)build the project.

If there are no errors, the code is assembled and linked and the code can now be debugged using the CrossView Pro debug environment.

4.3 Preparing the Debugger

- Open the ***EDE\CrossViewPro_Options*** menu from the EDE tool bar.
- In the ***Debugger*** tabsheet, select the radio button *Simulator* as shown below.
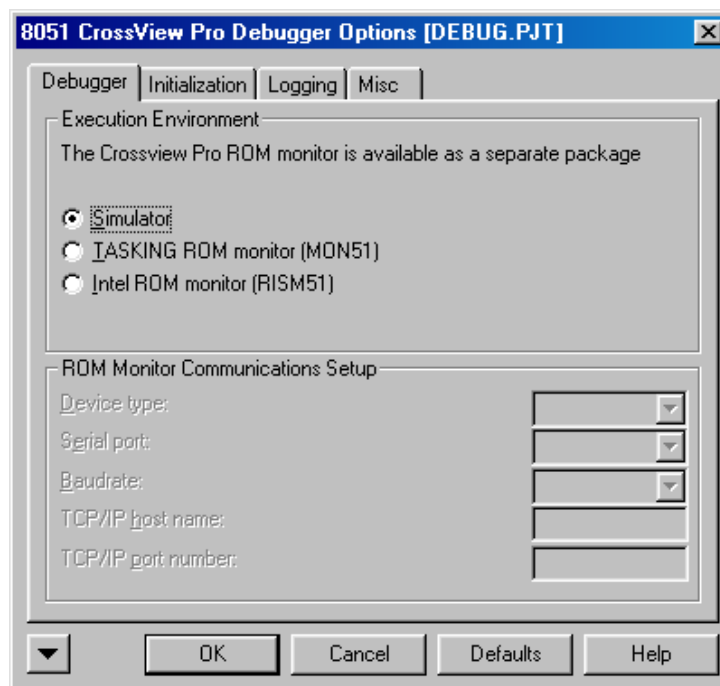

- Click on the ***OK*** button to save the settings.

4.4 Starting the Debugger

- To start the Tasking debug environment, click on the debugger icon ⬥ on the EDE toolbar.
- The CrossView Pro debugger will open the project (*.abs* file) and automatically perform a *go-to-main* action which executes the startup code.
- If the data transfer was successful, a ***CrossView Pro*** screen similar to the one shown below will appear:
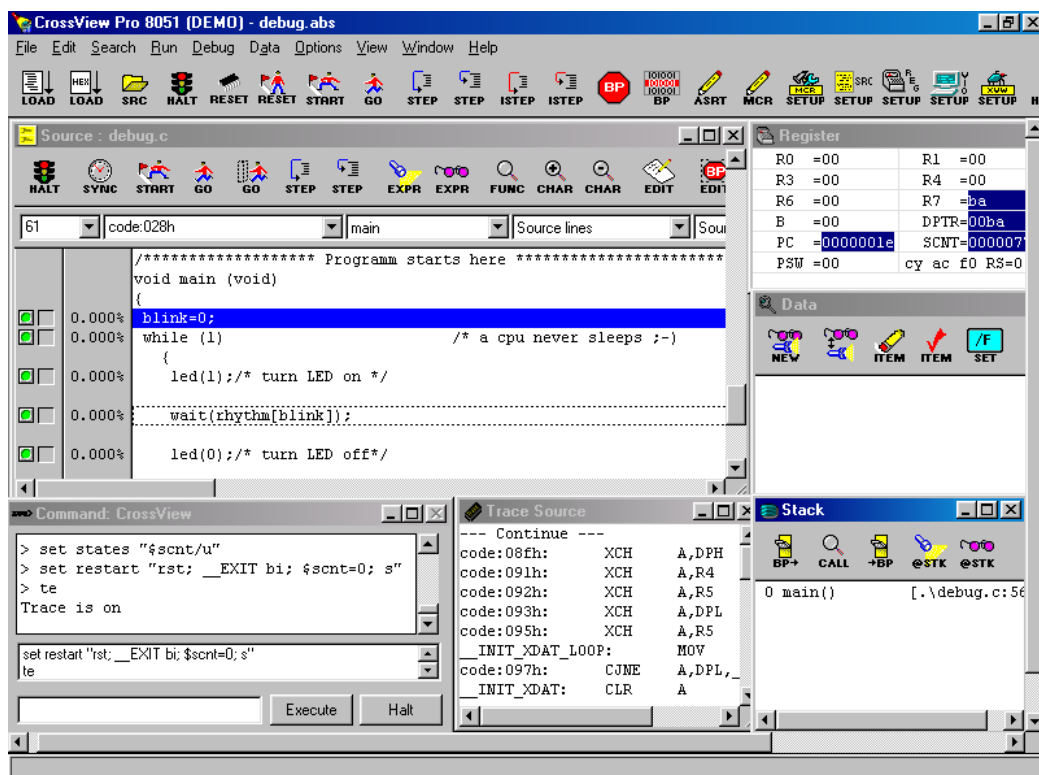
The window marked *"Source: debug.c"* is the Debug window. The *Command: CrossView* window can be used to enter commands. You may need to open, resize and /or move some windows to make your screen look similar to the screen capture. You can open inactive windows by choosing the desired window from the ***View*** pull-down menu.

4.5 Tasking CrossView Pro Debug Features

The CrossView Pro Debug environment provides a variety of commands supporting easy code debugging. The following section gives a short overview of some important debug commands:

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.

Clicking this button *Continue execution (same as F5)* runs the program without active debug functions. To stop program execution at a desired point, a breakpoint can be placed before the *Continue execution* button is pushed.

The *Run to cursor (same as F7)* command executes the program to the current cursor position within the code window. This allows use of the cursor line as a temporary breakpoint.

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

CrossView Pro uses *Step (into function calls)* to single step one instruction at a time. The *Step (into function calls)* command is also used to access a subroutine or function.

The *Step (over function calls)* command allows to execute a command line without single stepping into the function.

The *Show selected source expression* command displays the current value of a variable.

The *Watch selected source expression* command is used to display the value of a variable with its immediate updates.

4.6   Using the Tasking CrossView Pro Debug Features
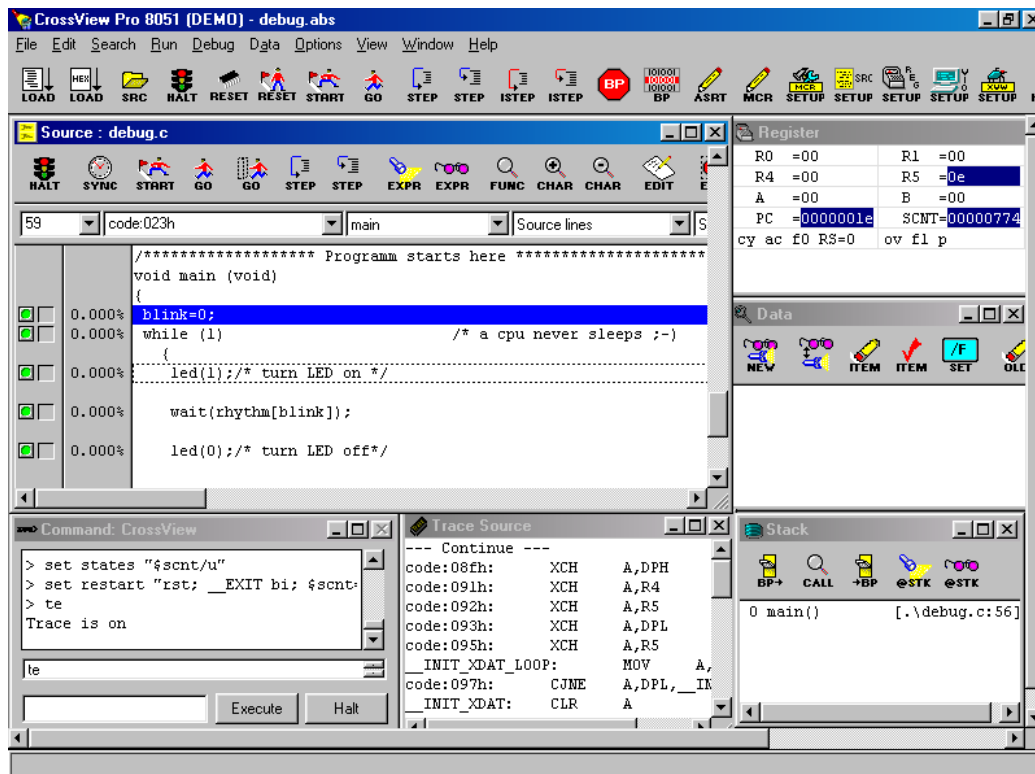
The following section describes the use of specific functions within the CrossView Pro debug environment in conjunction with the *Debug* example code.

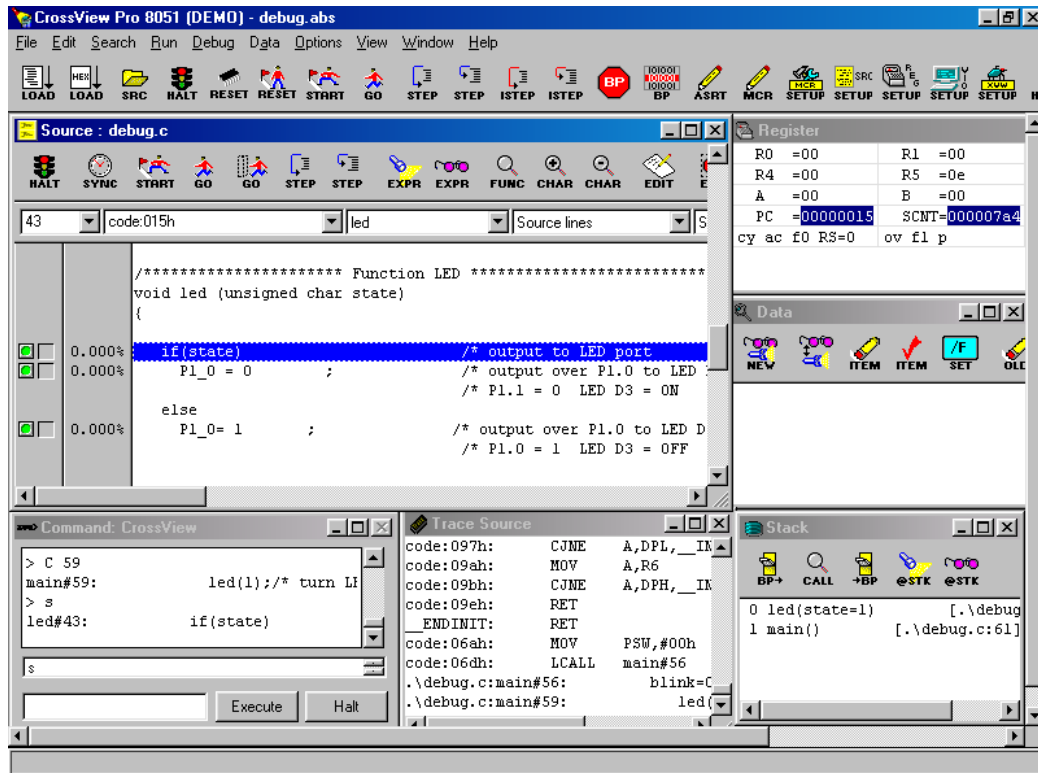### 4.6.1   Single Stepping and Watch Window

Run to cursor (F7)

- Click in the source code line *led(1);* of the Debug program and choose *Run to Cursor*. Your program will be executed until it reaches this line.

Step (into function calls)

- Click on the *Step (into function calls)* icon to enter the *'led(1)'* function.

- Click on the *Step (into function calls)* icon twice to single step through this function.

Step (over function calls)

- Click on the *Step (over function calls)* icon to execute the *wait(rhythm[blink]);* function without entering it. All subroutines are executed immediately and not displayed in the debug window.

Show and Watch Expression

- The debug commands *Show selected source expression* and *Watch selected source expression* can be used to display the value of a variable. Select the variable "**Blink**" in the source code line *blink++;*.
- Click on the *Show selected source expression* icon. The current value *"blink/n : 0"* is now displayed in the **Data** window.
- Click on the *Watch selected source expression* icon. The current value *"Watch blink/n : 0"* is now displayed in the **Data** window.
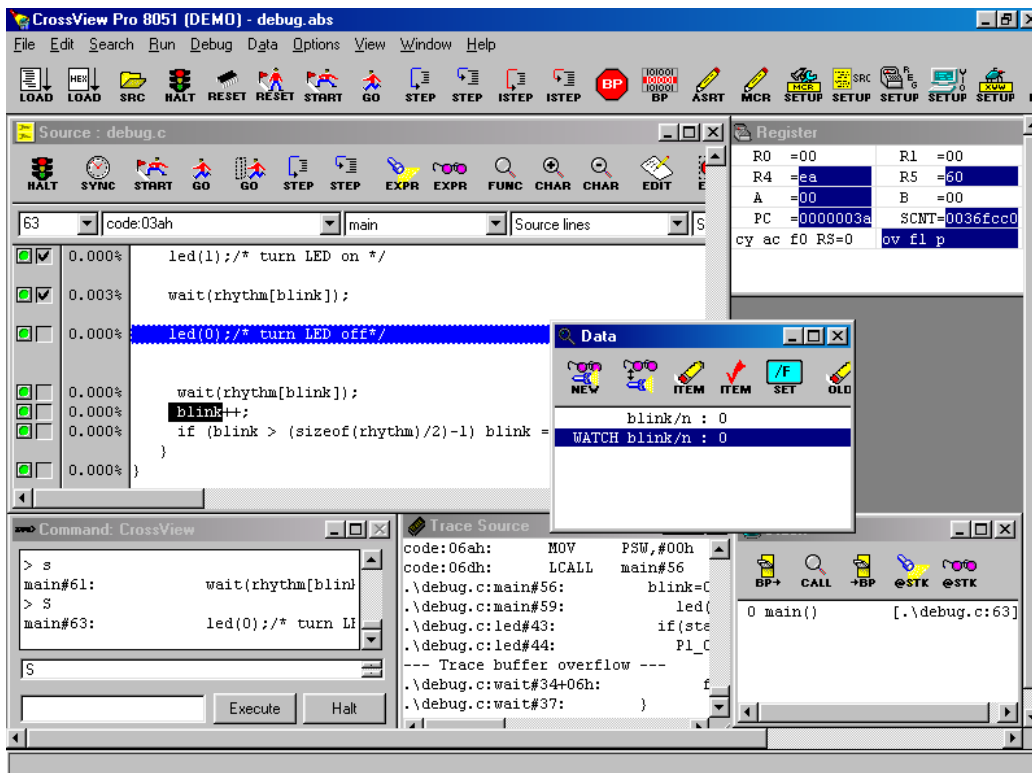
- Click in a different source code line in order to disable the selection of the variable "**Blink**".

4.6.2 Breakpoints

In the *Source* window debug.c you will find a green button in front of each code line. You can insert a breakpoint to any desired code line by clicking on the green button.

- Click on the green button in the source code line *"if (blink > (sizeof (rhythm)/2)-1) blink = 0;"*. The green button will change to a red button which indicates a breakpoint has been inserted in this code line.
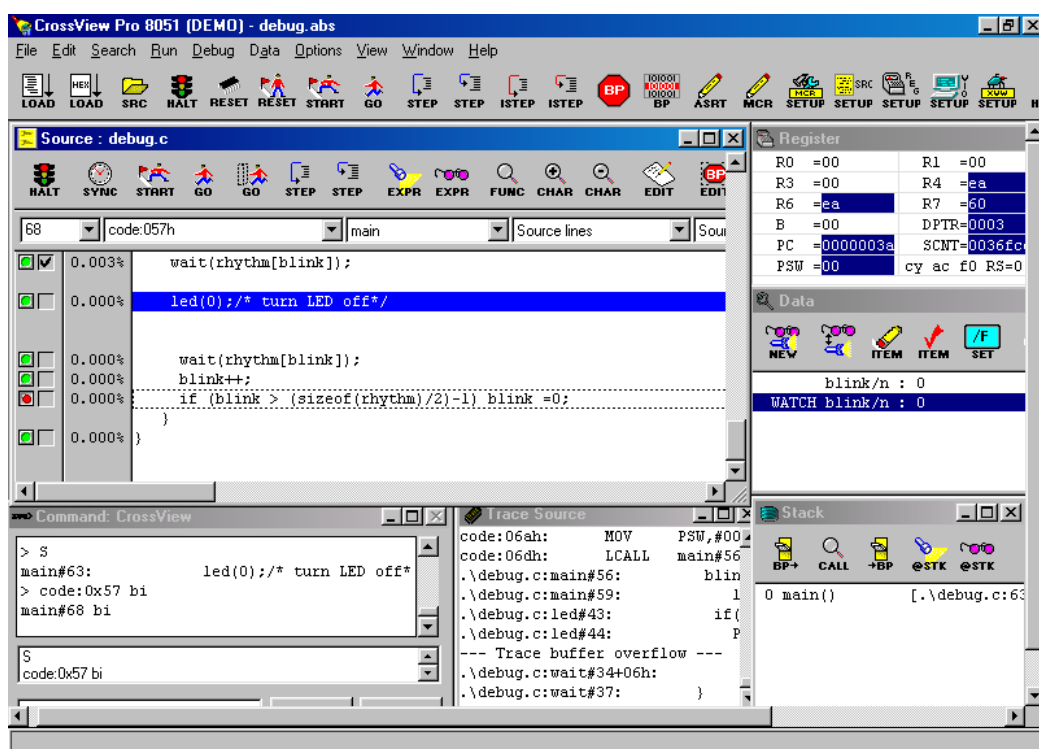
- Now click on the *Continue execution* icon. The program will run and stop at the breakpoint.
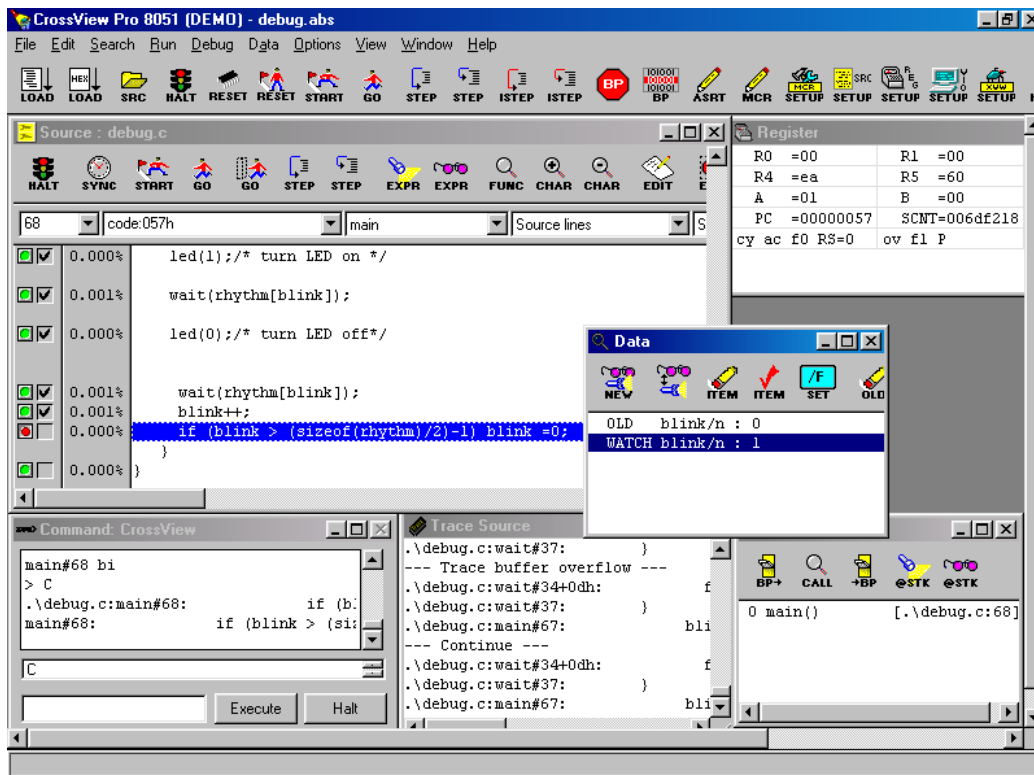
- Notice the changes of the variable "**Blink**" in the *Data* window. The value of this variable is updated in the *"Watch blink/n : "* line while the *"Show blink/n : "* line only indicates that the value has changed by adding the prefix *OLD*. No update of the variable value is shown in this line.

4.6.3 Exiting CrossView Pro

- Open the *File\Exit* menu from the EDE tool bar to close the current CrossView Pro debug session.
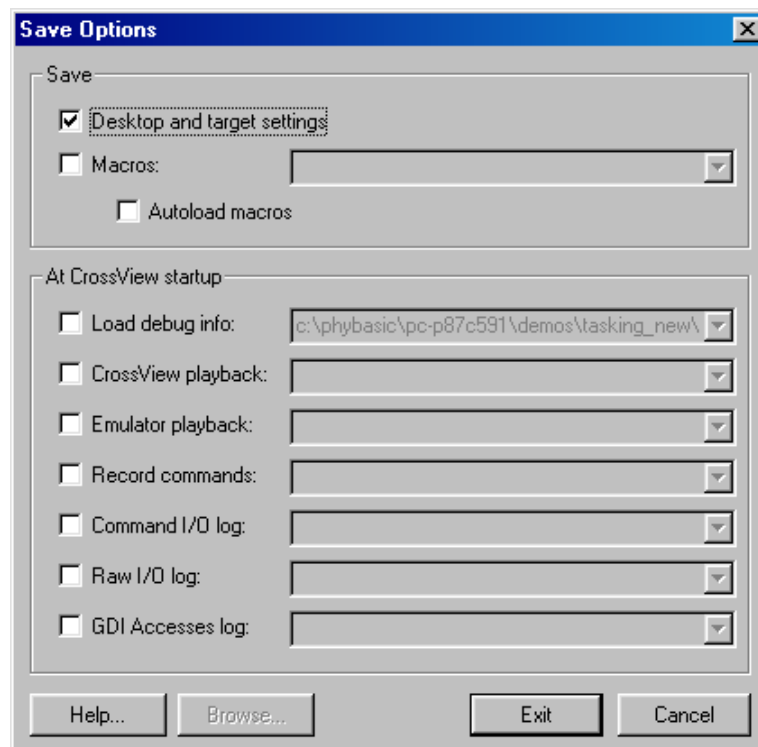
- Click on the *Exit* button in the *Save Option* menu to exit CrossView Pro.

4.7 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 Flash memory of the phyCORE-P89C51RD2.

- Open the ***EDE\Linker Options*** menu from the EDE menu bar.

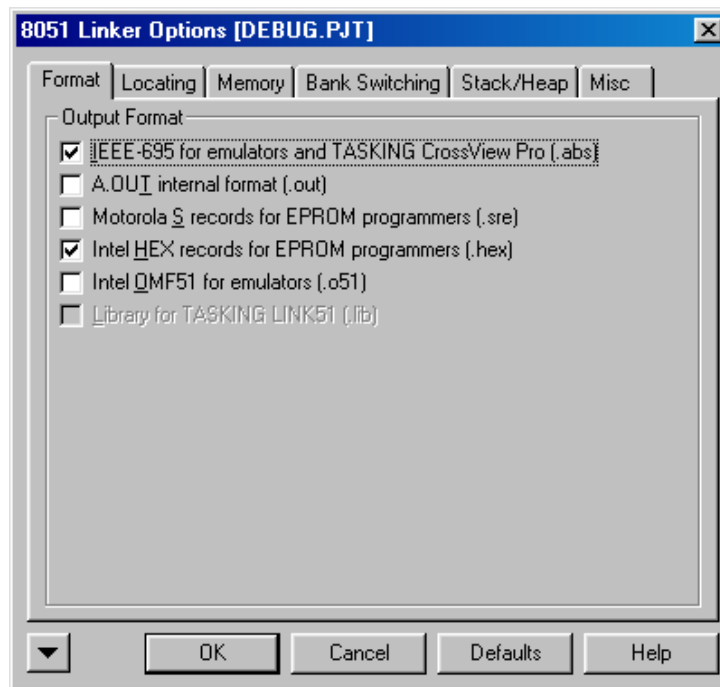- Activate the *Intel HEX records for EPROM programmers* checkbox and click on ***OK*** to save these settings.
- Click on the *Execute 'Rebuild' Command* 🖩 icon at the EDE tool bar or open the ***Project*** menu and select ***Rebuild***.
- Download the created ***Debug.hex*** file (located in ***C:\PHYBasic\pC-P89C51RD2\Demos\Tasking\Debug***) to the Flash memory. For general download procedure information *refer to sections 2.2 through 2.4*.
- Press the Reset button S2 on the Development Board to start the program.
- Now you can watch your final debug example execute. The program will flash the LED D3 with alternating on and off duration.

5 Advanced User Information

This section provides advanced information for successful operation of the phyCORE-P89C51RD2 in conjunction with the Tasking 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 Cstart.asm

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

The ***Project_cstart.asm*** routines always start at code memory address 0x0000. This is where the reset vector with the jump instruction to the actual ***Project_cstart.asm*** 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.

The configuration of the startup code is done via macro preprocessor defines which are generated automatically by EDE according to the settings in *EDE/Processor Options*.

The ***Project_cstart.asm*** file is automatically generated every time a new project is created if the checkboxes *Generate Startup* und *Add startup code* are activate. These options are available in the ***Startup*** tabsheet accessible within the ***EDE\Processor Options...*** menu. If no ***Project_cstart.asm*** file is included in your project you can add the file ***cstart.asm*** which is located in the ***\lib*** folder within the Tasking installation directory.

5.3 Linking and Locating

The Linker must combine several relocatable 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 space in regards to the memory types of the phyCORE-P89C51RD2. XDAT segments always must be located to Random Access Memory (e.g. RAM), CODE segments should be located to 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 Tasking tool chain distinguishes the following segment types:

- CODE: code
- XDAT: 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 XDAT will usually reside in the external RAM device.

To ensure proper execution of your application, it is required that all XDAT 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 0x03FF (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 XDAT segments should end below 0x03FF (or 0x7FFF if a 32 kByte external RAM is available). We recommend generation of a *.l51* map file for your project and inspection of the memory map information within this file.

Document: **phyCORE-P89C51RD2 QuickStart Instructions**
Document number: L-466e_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

© PHYTEC Meßtechnik GmbH 2002   L-466e_6