phyCORE-P89C51RD2

QuickStart Instructions

Using PHYTEC FlashToolsOCF for on-chip Flash and the Raisonance Integrated Development Environment (RIDE) for 8051 and XA Demo 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

A product of a PHYTEC Technology Holding company

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

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

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

	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

2nd 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 Raisonance's Integrated Development Environment (RIDE) for 51+XA, and
- instructions on how to run example programs on the phyCORE-P89C51RD2, mounted on the PHYTEC phyCORE Development Board LD 5V, in conjunction with RIDE for 51+XA

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 Raisonance Integrated Development Environment (RIDE) for 51+XA

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 Raisonance Integrated Development Environment (RIDE) for 51+XA. 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 Raisonance tool chain and FlashTools OCF.

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

RAISONANCE

http://www.raisonance.com - or - http://www.amrai.com (US)
support@raisonance.com - or - support@amrai.com (US)
support@raisonance.fr (Europe)

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 Raisonance Integrated Development Environment (RIDE) for 51+XA

The Raisonance tool chain fully supports the entire Philips XA and 8051 derivative microcontroller families. It includes an ANSI-C compiler, macroassembler, linker/locator, simulator/debugger and ROM monitor/debugger within the RIDE development environment.

From the unique RIDE user interface projects can be developed for an application based either on an 8051 derivative, a XA derivative, or both. Such flexibility makes migrating from 8-bit to 16-bit architectures easier.

The Raisonance tool chain produces OMF object files that are supported by most in-circuit emulators. The OMF-to-HEX utility converts a Raisonance .aof format file into an Intel hexfile that is suitable for programming into the on-chip Flash on the PHYTEC phyCORE-P89C51RD2 target board.

The Raisonance tool chain consists of the following tools; all integrated in RIDE:

- **C Compiler**
- **Assembler**
- **RTOS**
- **Linker/Locator**
- **Simulator/Debugger**
- **ROM Monitor**

All these tools and many utility programs are available in two versions: as Win32 DLL called internally from RIDE, and DOS-based executables (*.exe) that can be called externally. These two versions are strictly equivalent regarding the generated code.

The evaluation version of the RIDE development environment is limited in manipulable code size as follows:

- 4 kByte for the 8051.
- 8 kByte for the XA.

Other than these restrictions, the evaluation tool chain functions exactly as the full version does, enabling full evaluation of the features and functionality of Raisonance development tools. This demo version can be upgraded by entering a serial number.

Raisonance Integrated Development Environment (RIDE)

RIDE is a Windows-based Graphical User Interface for all Raisonance tools. All compiler, assembler, linker/locator and debugger options are configured with simple mouse clicks. RIDE runs under Windows 95/98/2000 and NT.

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

RC51 C Compiler

The RC51 compiler and MA51 assembler are designed specifically for 8051 controllers.

The Raisonance RC51 compiler provides the fastest and smallest code using industry benchmarks.

The Getting Started Guide from Raisonance provides more information on these tools. It also includes an introduction on ‘Migrating from the 8051 to the XA’.

Debug Environment

RIDE includes a simulator/debugger and a ROM Monitor that supports debugging either via software on a host-PC or in target hardware. All the debugging functions are enabled in the demo version with the same restrictions in manipulable code size as follows:

- 4 kByte for the 8051,
- 8 kByte for the XA (Page Zero mode only).

2 Getting Started

What you will learn with this Getting Started example:

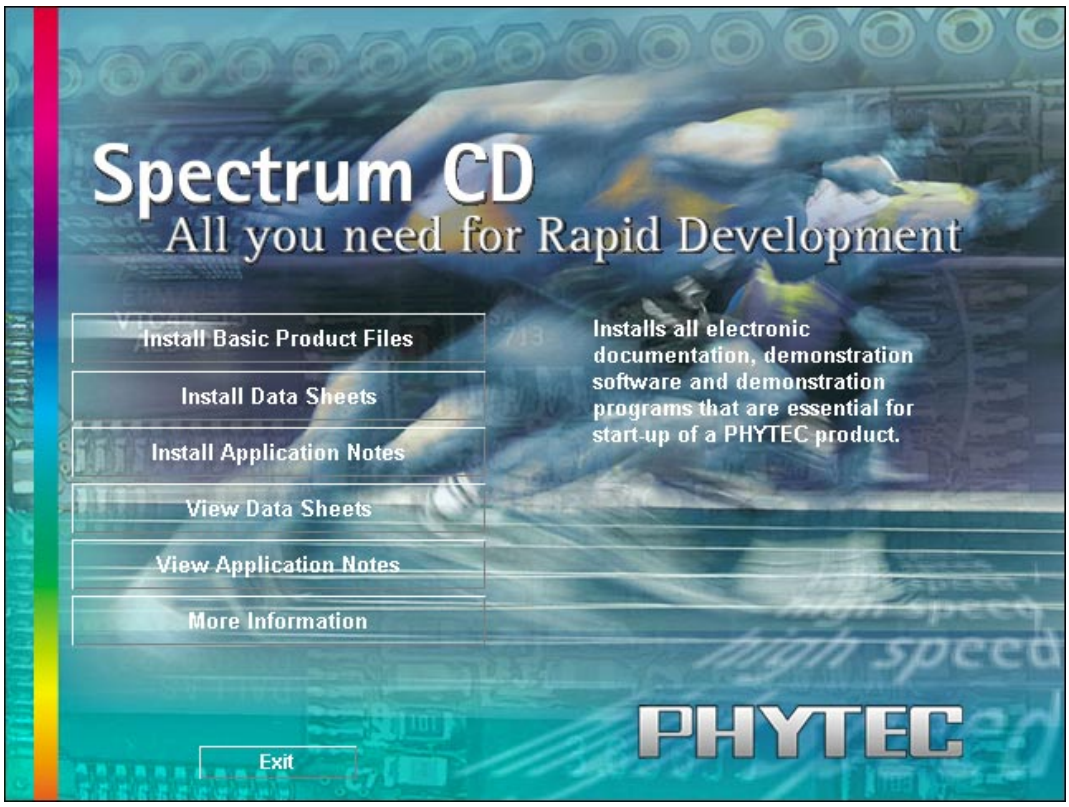
- installing Rapid Development Kit software
- starting PHYTEC FlashToolsOCF for Windows download utility
- interfacing the phyCORE-P89C51RD2, mounted on the phyCORE Development Board LD 5V, to a host-PC
- downloading example user code in Intel hexfile format from a host-PC to the controller's on-chip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

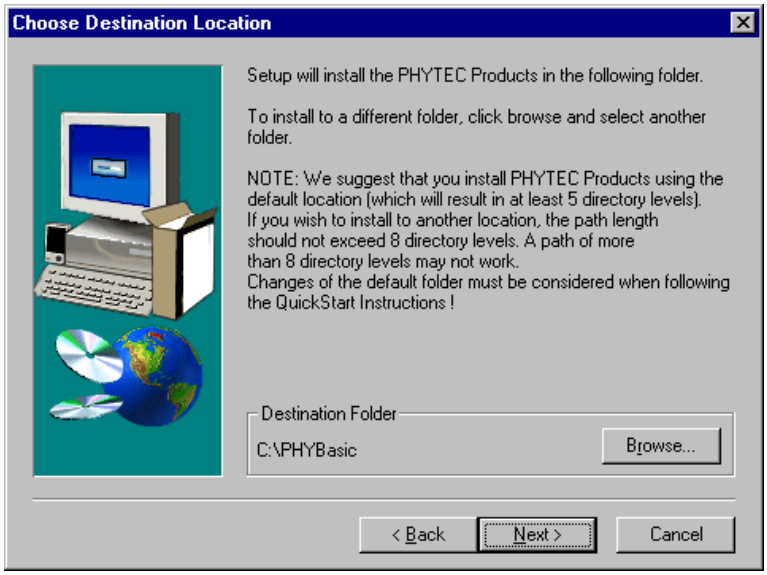
The following window appears:



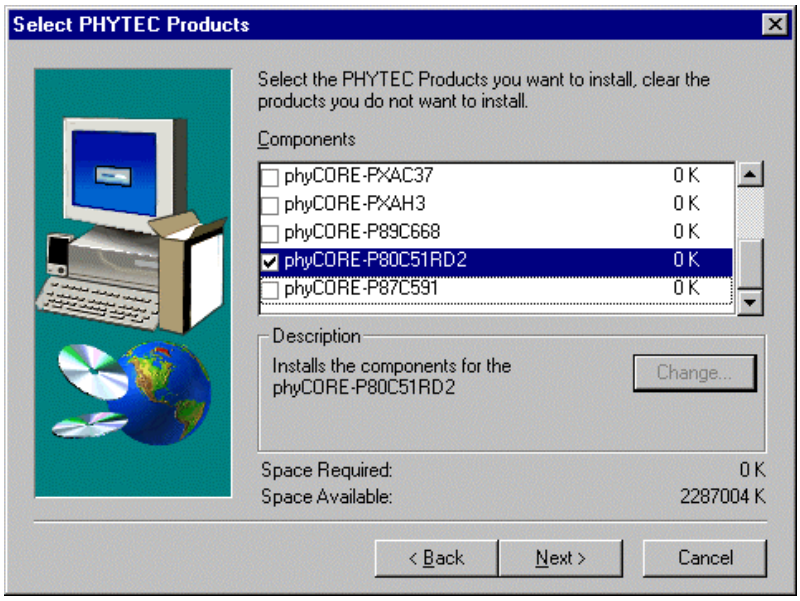
- Choose the *Install Basic Product Files* button.
- After accepting the Welcome window and license agreement, select the destination location for installation of Rapid Development Kit software and documentation.

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.



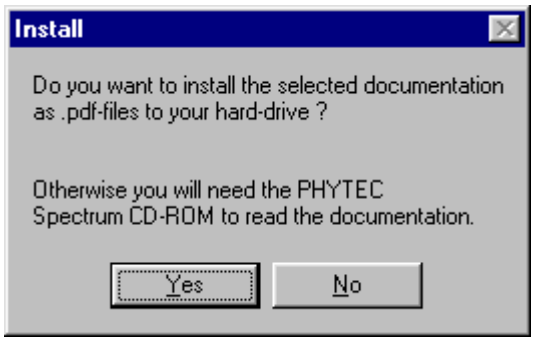
- In the next window, select your Rapid Development Kit of choice from the list of available products. By using the *Change* button, advanced users can select in detail which options should be installed for a specific product.



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.



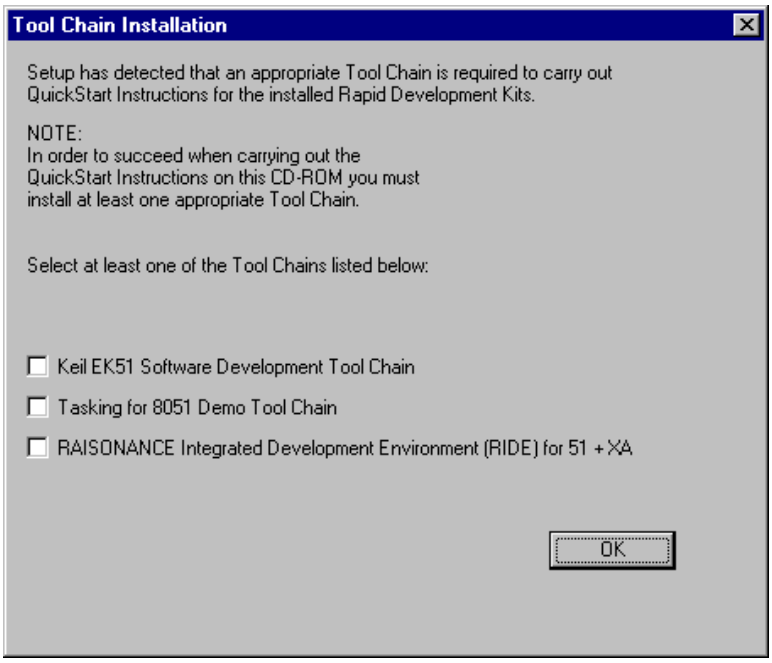
If you decide **not** to copy the documentation to your hard-drive, you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.

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 Raisonance Software Development tool chain for 51 + XA[1].



The applicable Raisonance 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 the Raisonance tool chain from the Spectrum CD-ROM even if another version of RIDE is 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.

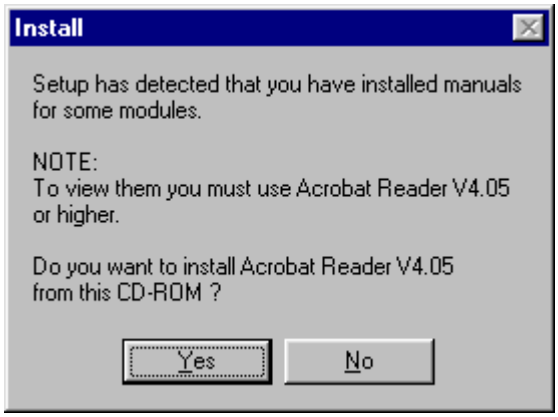
- After accepting the Welcome window and license agreement, select the destination location for installation of the Raisonance tool chain. The default location is ***C:\Ride***.

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

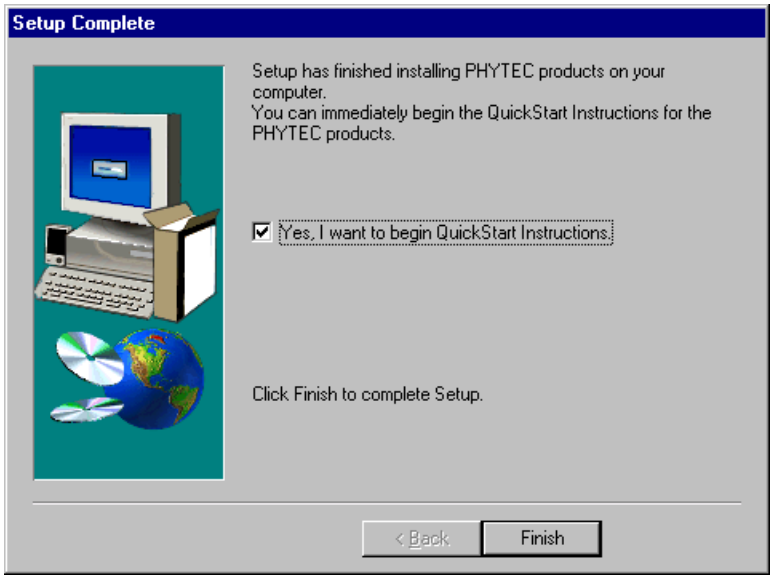
The applicable Raisonance Software Development tool chain for 51+XA will be installed to your hard drive. Additional software, such as Adobe Acrobat Reader, will also be offered for installation.

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.

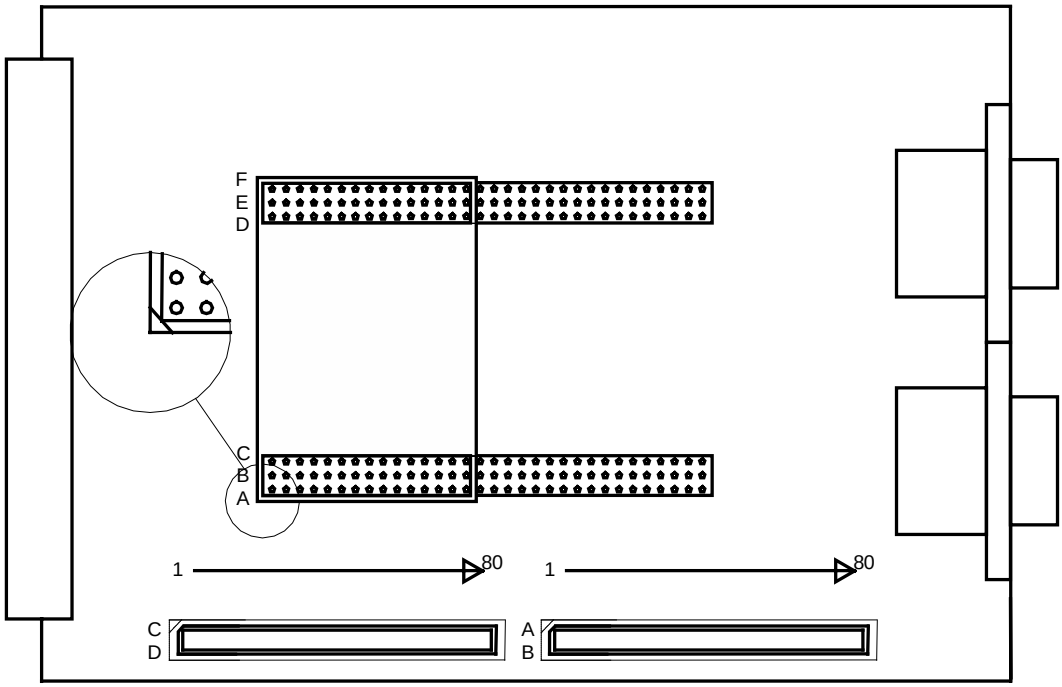


Figure 1: Mounting the phyCORE-P89C51RD2 onto the phyCORE Development Board LD 5V

- 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.

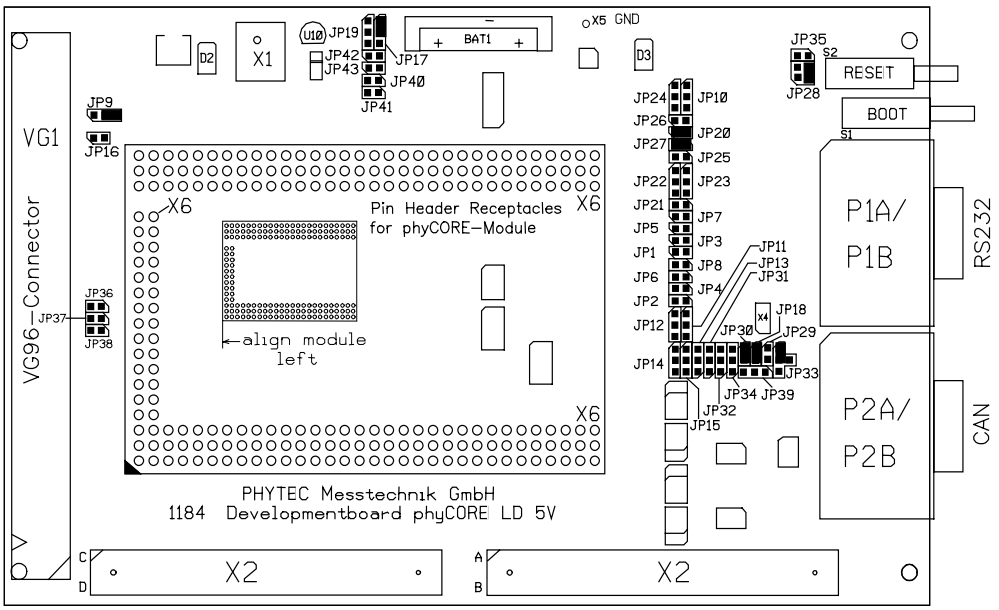


Figure 2: Important Connectors, Buttons and Suitable Jumper Settings on the phyCORE Development Board LD 5V

- 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*).

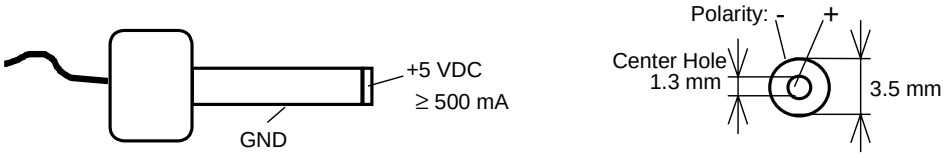


Figure 3: Power Connector

- 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 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.

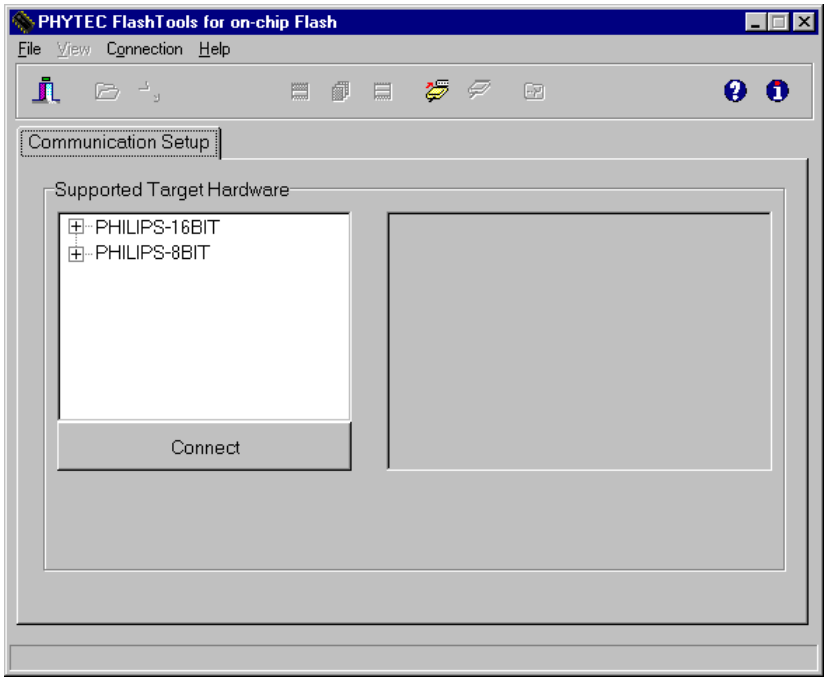
The PC-side software FlashToolsOCF has been designed to work together with the Boot Loader firmware located in an on-chip BOOT-ROM of the P89C51RD2 controller. Proper connection of a PHYTEC phyCORE module to a host-PC enables the software portion of FlashToolsOCF to recognize and communicate to the firmware portion.

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

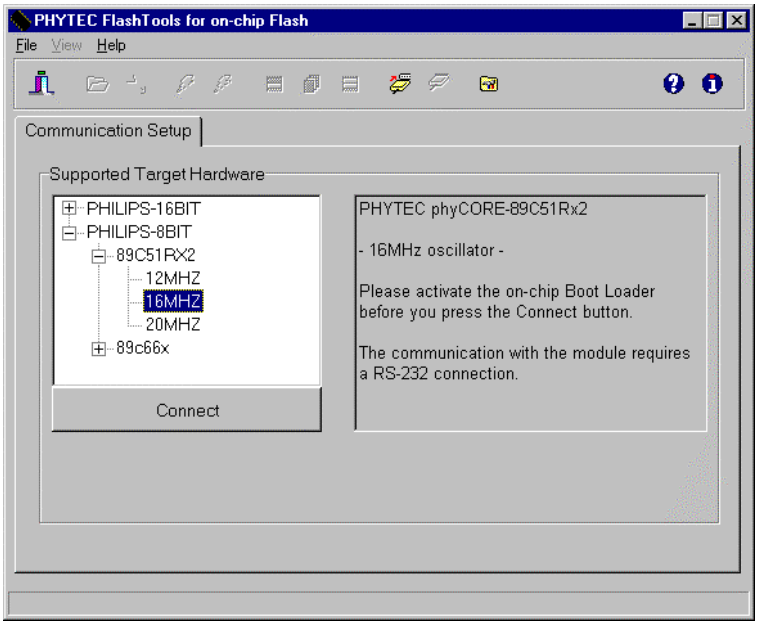
It is recommended that you drag the FlashToolsOCF icon onto the desktop of your PC. This enables easy start of FlashToolsOCF by double-clicking on the icon.

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.



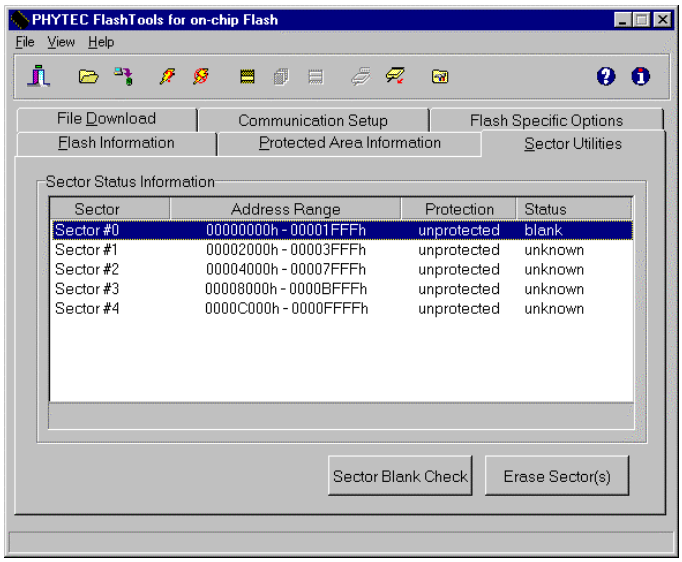
Note:
Always ensure that the phyCORE-P89C51RD2 is in Flash programming mode before pressing the *Connect* button (*refer to section 2.2*).

- 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.

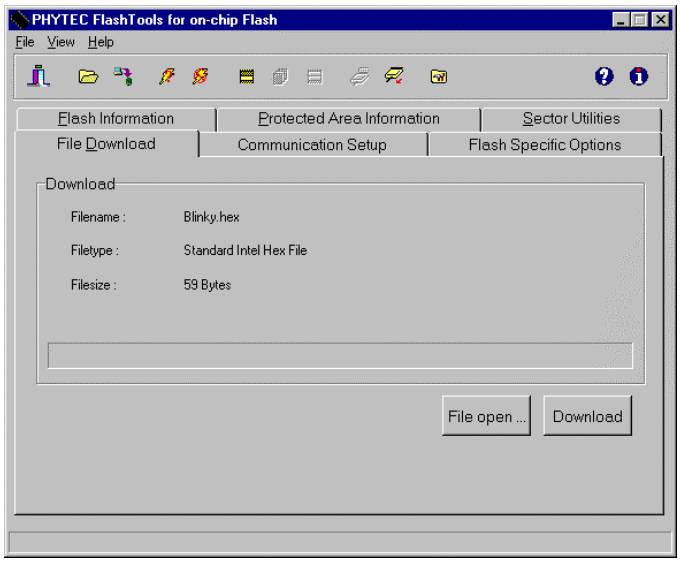
The microcontroller firmware tries to automatically adjust to the baud rate selected within the baud rate tab. However, it may occur that the selected baud rate cannot be attained. This results in a connection error. In this case, try other baud rates to establish a connection.
Before attempting each 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:

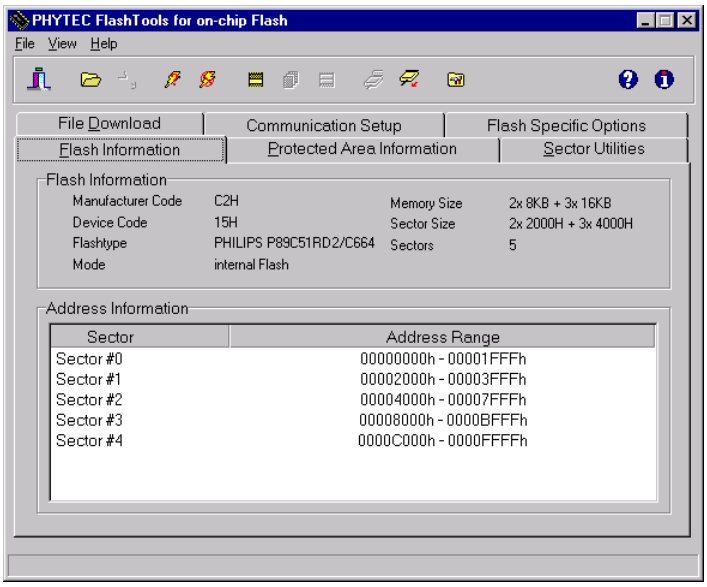
Sector Utilities allows blank check and erasure of individual sectors of Flash memory:



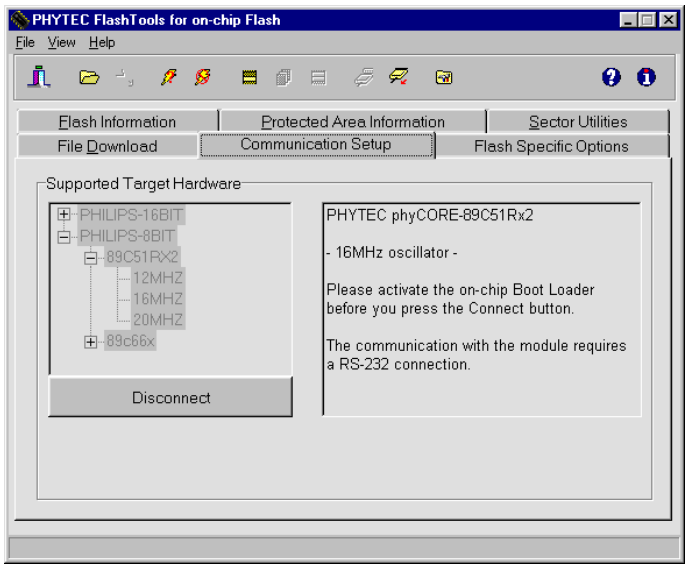
File Download downloads specified hexfiles to the target hardware:



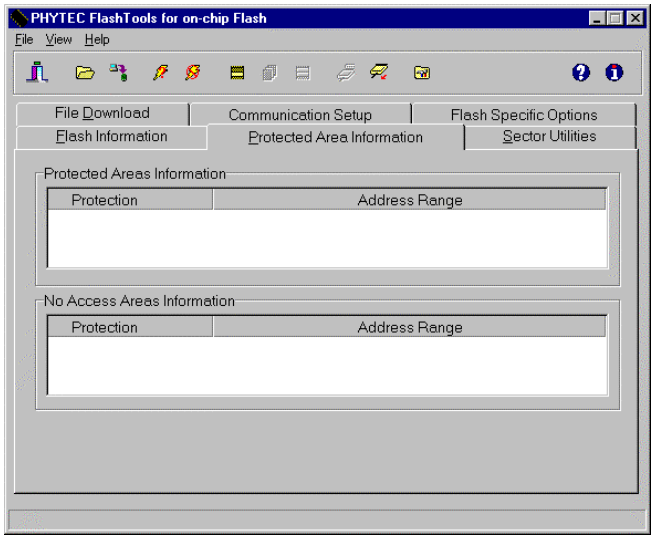
Flash Information shows Flash type, sector and address ranges in Flash memory:



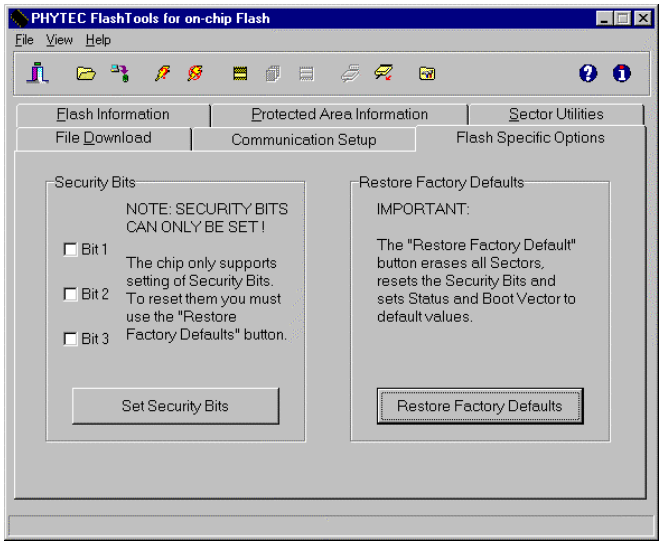
Communication Setup allows connection to and disconnection from the board (this is the same window that was used when you first entered FlashToolsOCF):



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"

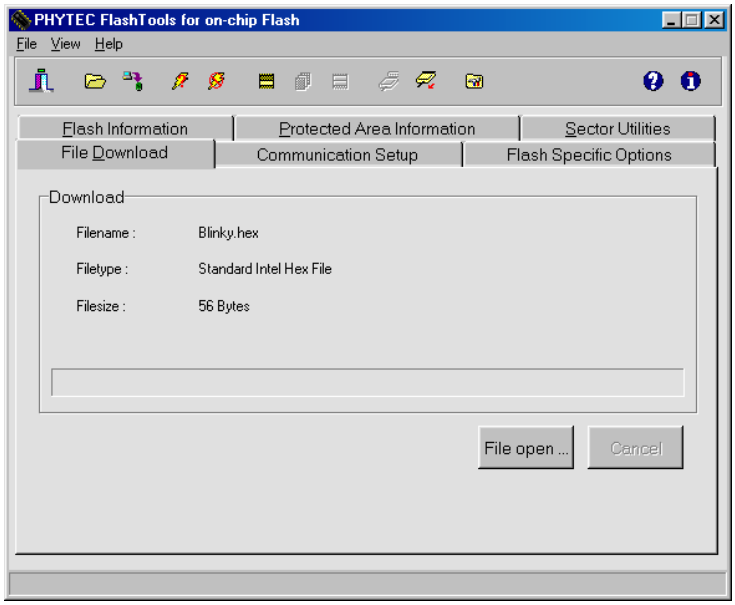
The "Blinky" example downloads a program to the Flash that, when executed, manipulates the LED D3 on the phyCORE Development Board LD 5V that is located above the jumper field (*refer to Figure 2*).

- 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\Raisonance\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.

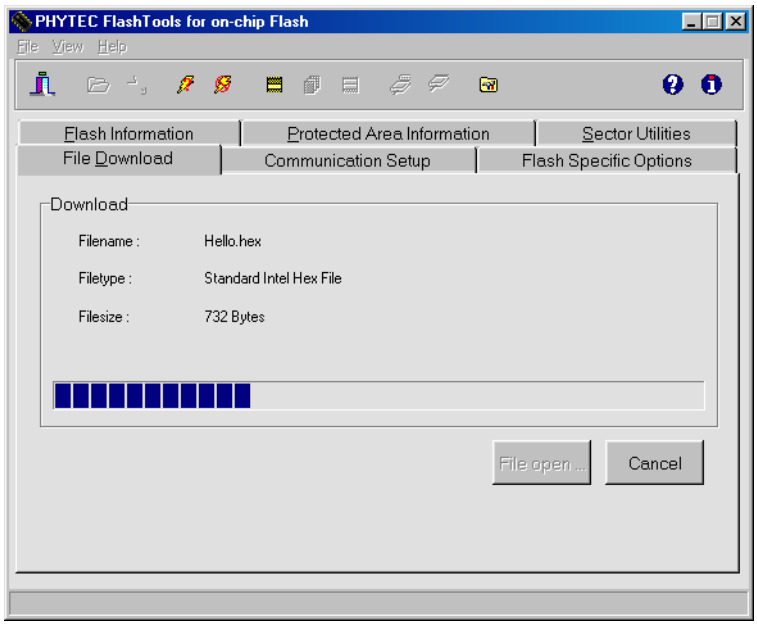
2.4.2 "Hello"

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\Raisonance\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.



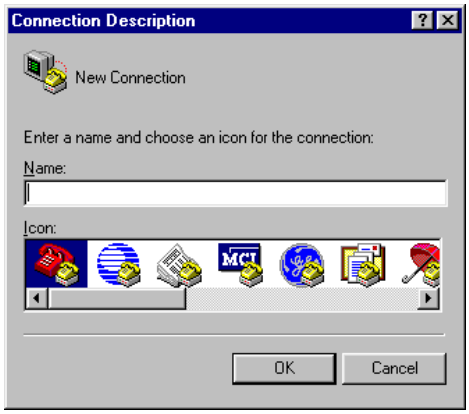
If the selected Flash Sector into which you wish to download code is not empty (i.e. erased), a warning dialog box will appear, indicating "Location not empty! Please erase location and try again". In this event, select the *Sector Utilities* tab from the FlashToolsOCF tabsheet, highlight *Sector #0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

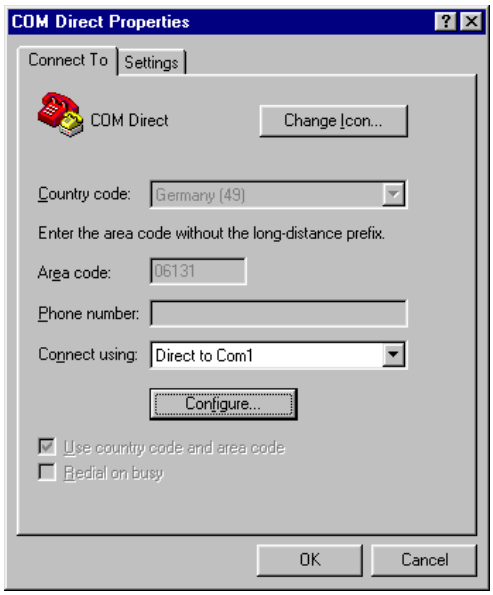
- The Connection Description window will now appear. Enter "COM Direct" in the *Name* text field.



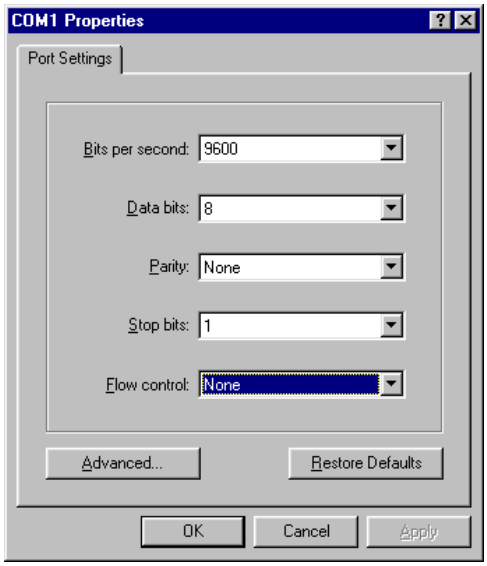
- Next click on *OK*. This creates a new HyperTerminal session named "COM Direct" and advances you to the next HyperTerminal window.

[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).

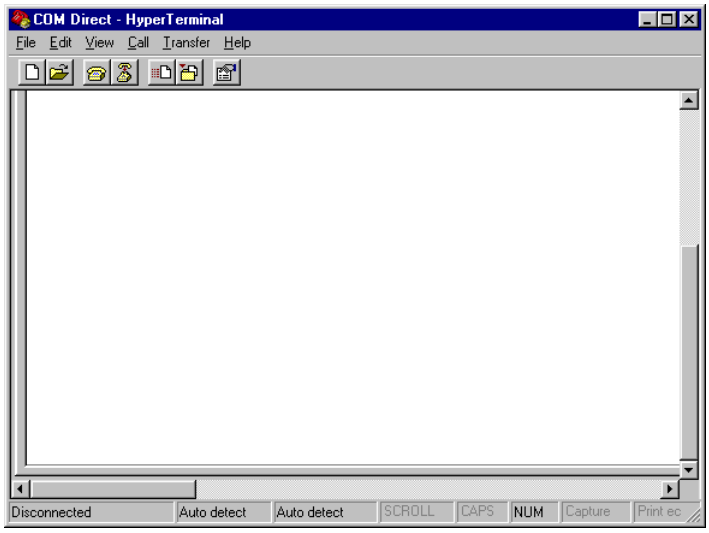


- Click the *Configure* button in the *COM Direct Properties* window to advance to the next window (*COM1/COM2 Properties*).



- 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 Raisonance tool chain
- how to configure the Raisonance tools within the Integrated Development Environment (RIDE) for 51+XA demo version
- 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 Raisonance Tool Chain

The Raisonance Integrated Development Environment (RIDE) for 51+XA demo 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 ***install.exe*** from within the ***\Software\Raisonance*** folder of your PHYTEC Spectrum CD.

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

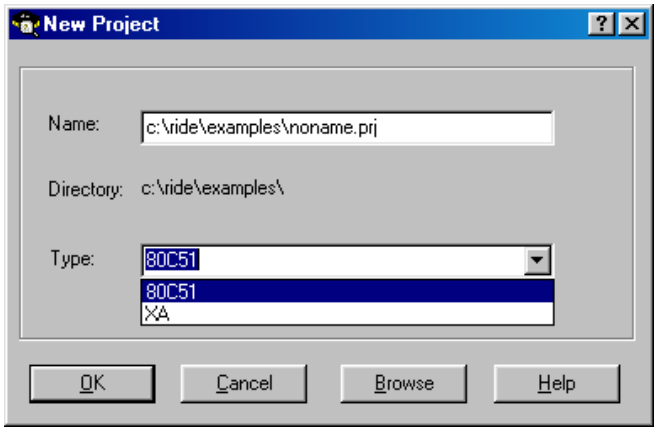
- Start the tool chain by selecting *Ride IDE* from within the *Programs*|*Raisonance Kit* program group.

After you start RIDE, the window shown below appears. From this window you can create projects, edit files, configure tools, compile, assemble, link and debug.

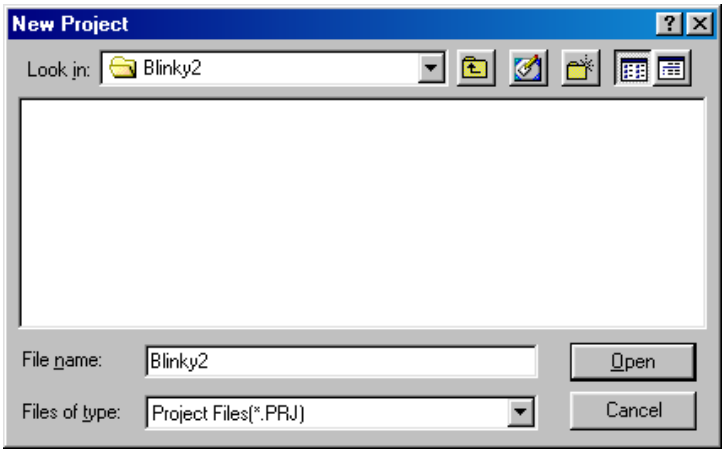
3.2 Creating a New Project and Adding an Existing Source File

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

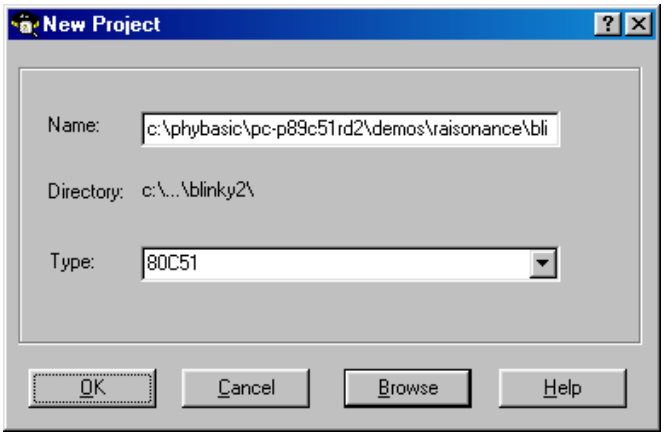
- To create a new project file open the ***Project*** menu and choose ***New*** within the RIDE menu bar. The window as shown below appears.



- Make sure that the correct architecture *Type* is selected. For this example select *80C51* (XA may be currently selected).
- Click on the *Browse* button and change to the project directory created by the installation procedure (default location ***C:\PHYBasic\pC-P89C51RD2\Demos\Raisonance\Blinky2***
- In the text field '*File name*', enter the file name of the project you are creating. For this example, enter the name ***Blinky2***.

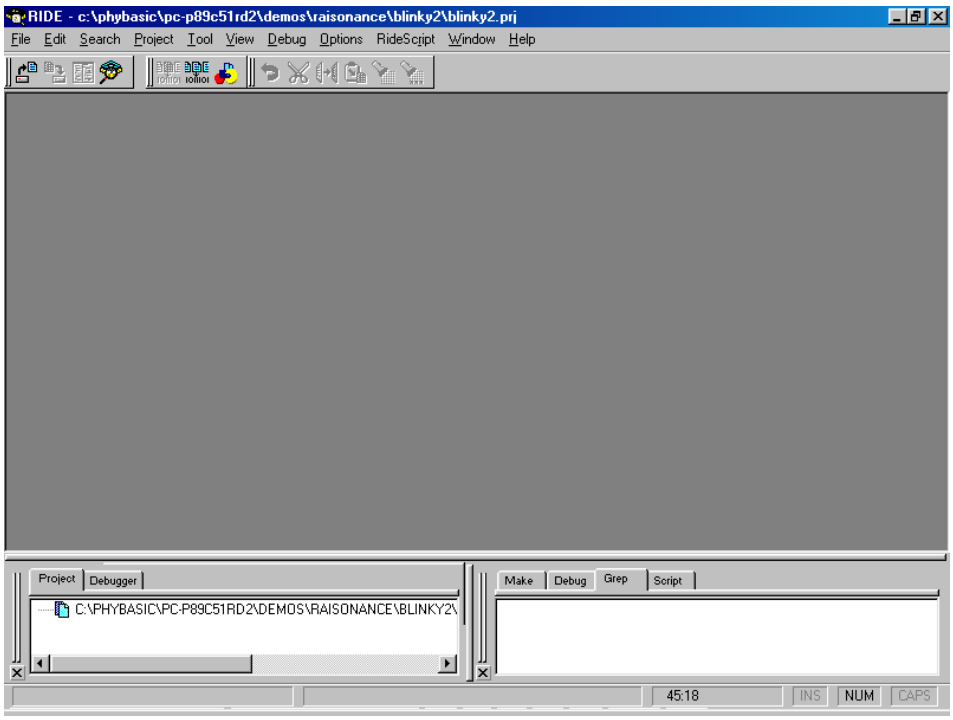


- Click on *Open*.

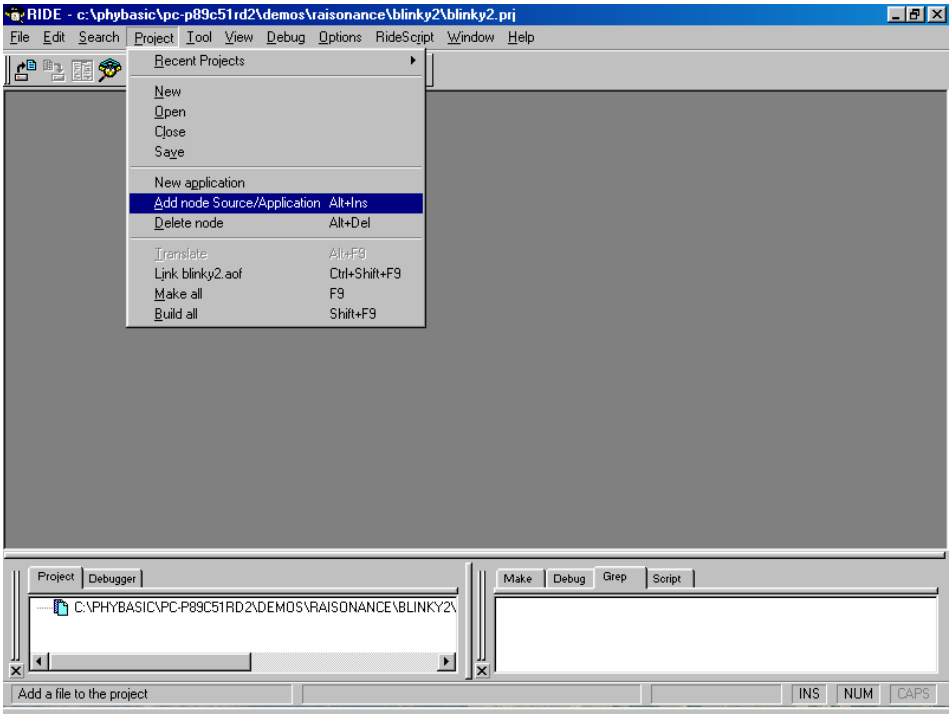


- Click on *OK* in the *New Project* window.

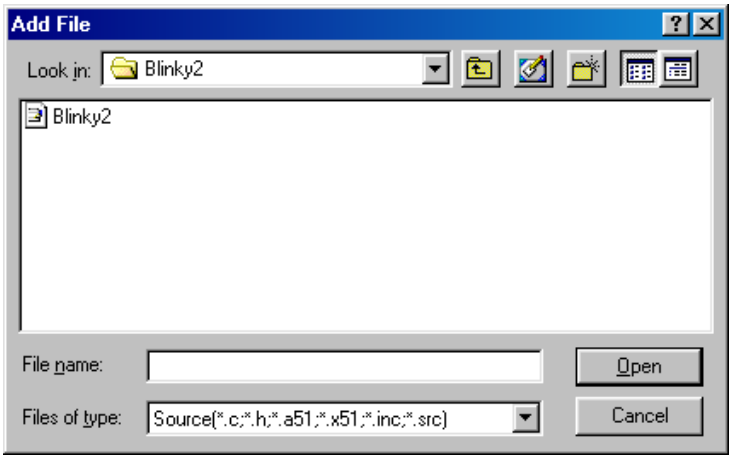
- The following window will appear:



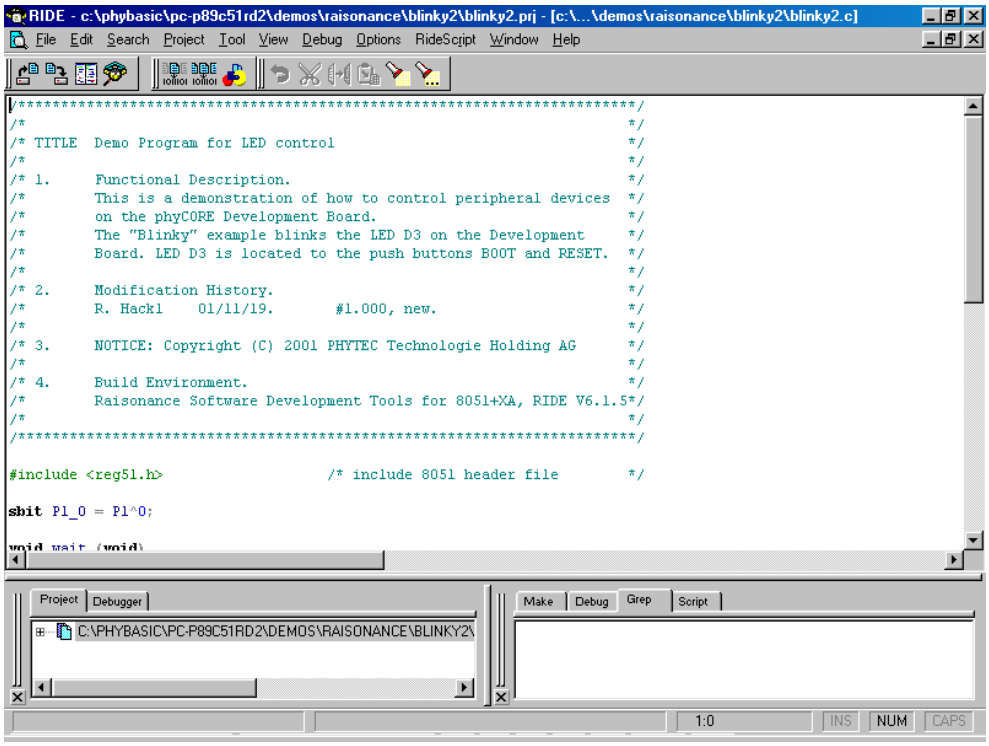
- From here, you will be able to add various files to your project.
- Open the *Project* menu and choose *Add node Source/Application*.



- The following window will appear:



Select the file ***Blinky2.c*** in the ***Add File*** window and click on *Open*.



At this point you have created a project called ***blinky2.prj*** and added an existing C source file called ***blinky2.c***.

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

3.3 Modifying the Source Code

- The source file ***blinky2.c*** is now open within the RIDE editor. If you closed the file, double click on the reference inside the project tree.
- Locate the following code section. Modify the section shown below (the values shown in bold and italic font) from the original 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< 40000; i++) /* delay for 40000 counts          */
       {
        wait ();             /* call wait function             */
       }
   }                         /* end of while(1)                */
}                            /* EOF                            */
```

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 *Save* icon  .

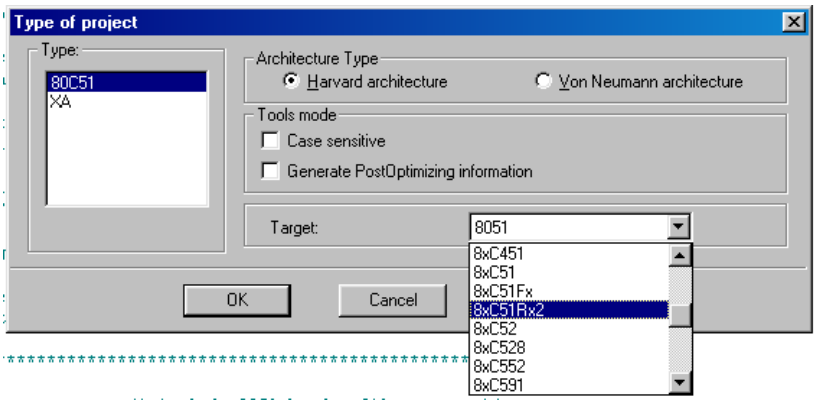
3.5 Setting Tool Chain Options

Raisonance tools include a Make utility that controls compiling and linking of various source files. 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. RIDE allows you to set various options with mouse clicks and these are all saved in your project file.

Note:
In most cases, options can be set at the project level. However, specific local options sometimes have to be set up differently when a file or a group of files require special options. In this case, a popup menu allows you to specify options at the level of the node.

To configure the Target:

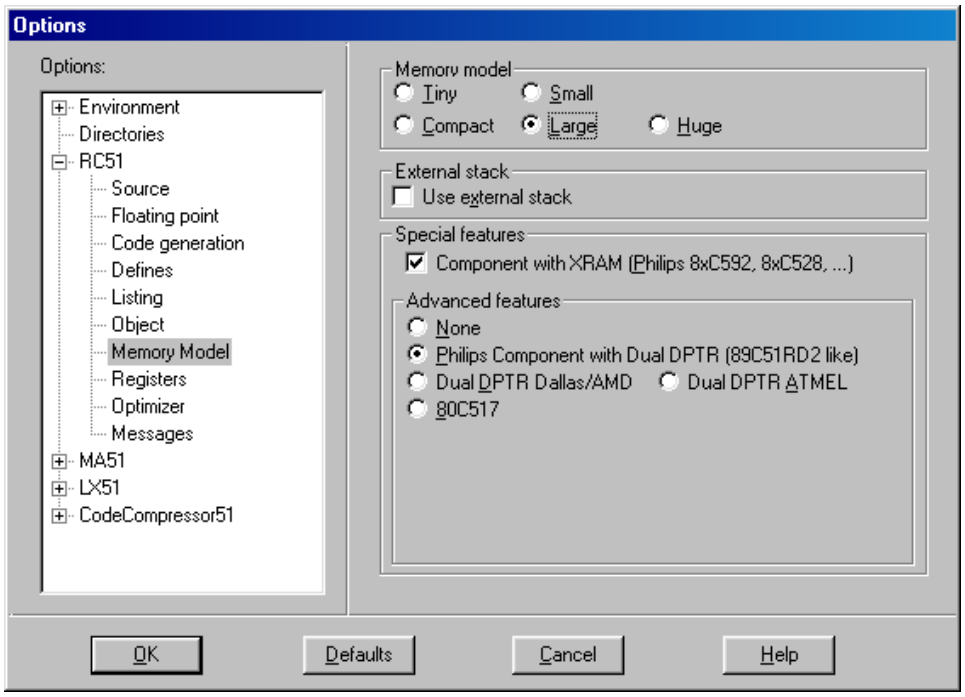
- Open the ***Options|Target*** menu and select the *8xC51Rx2* as shown below:



- Click on *OK* to save the configuration.

To configure the RC51 Compiler:

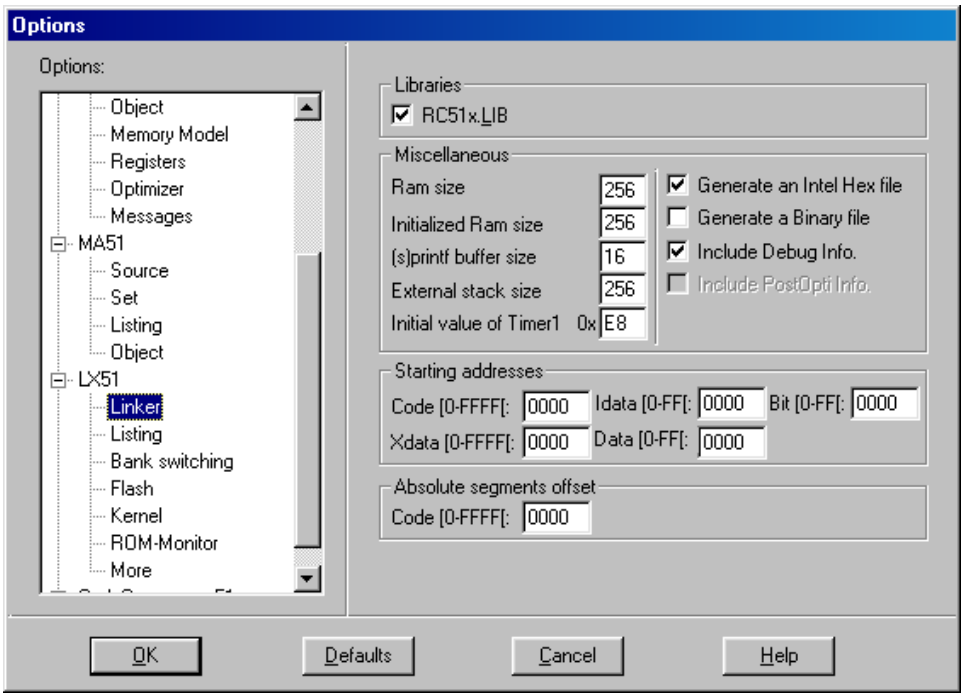
- Open the ***Options*|*Project*|*RC51*** menu and choose *Memory Model*.
- Select the *Large* memory model and activate the checkbox *Component with XRAM* under *Special features* and *Philips Component with Dual DPTR* under *Advanced features.*



- Click on *OK* to save these settings.

To configure the LX51 Linker/Locator:

- Open the ***Options|Project*** menu and choose *LX51\Linker*.
- Check that the *Generate an Intel Hex file* checkbox is active. This option should be enabled by default.
- All others options are correct to run our first example. Click on *OK* to save the configurations.



The linker/locator options are now suitable for the ***Blinky2*** project, enabling you to build an absolute object file (****.aof***) and a hexfile.

3.6 Building the Project

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

- Click on the *'Make All' Command* icon  from the RIDE toolbar or open the *Project* menu and select *Build All* or *Make All*.

If the program specified (***Blinky2.c***) contains any errors, they will be shown in the ***Message Window*** at the bottom of 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. 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 ***.m51**) are generated to help the debugging or troubleshooting and error searching process.

- If a list of errors appears, double-click on the error to open the file and locate the error. Use the editor to correct the error(s) in the source code and (re-)build the project.

3.7 Downloading the Output File

- 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\Raisonance\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 the Raisonance RIDE environment and close all projects that might be open.
- Open the ***Project*** menu and create a new project called ***Hello2.prj*** within the existing project folder ***C:\PHYBasic\pC-P89C51RD2\Demos\Raisonance\Hello2*** (default location) on your hard-drive. Select the 80C51 architecture for this project.
- Add ***Hello2.c*** and ***Serinit.c*** from within the project folder to the project ***Hello2.prj***.
- *Save* the project.

At this point you have created a project called ***Hello2.prj*** consisting of the C source files ***Hello2.c*** and ***Serinit.c***.

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

The same tool chain options can be used as for the ***Blinky2*** project described in *section 3.5*.

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, 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.

3.8.5 Downloading the Output File

- 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\Raisonance\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 Raisonance RIDE 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 Raisonance*.

The Raisonance RIDE integrated debugger offers two operating modes that can be selected in the ***Options\Project\LX51\ROM-Monitor*** and ***Options\Debug*** 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. RIDE simulates a wide variety of peripherals, including the serial port, external I/O, and timers.
- The **Real Mode,** using either the Raisonance ROM monitor or an In-Circuit Emulator, allows target-based debugging. When using the ROM monitor, the debugger communicates with the target hardware via a monitor kernel that is running on the target system.

The following examples utilize the Simulator interface.

Note:
Because the P89C51Rx2 controller family features on-chip Flash memory, the phyCORE-P89C51Rx2 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 Raisonance ROM monitor interface is not possible. Programs can be evaluated before code is downloaded to the controller using the Raisonance Simulator.

4.1 Creating a Debug Project and Preparing the Simulator

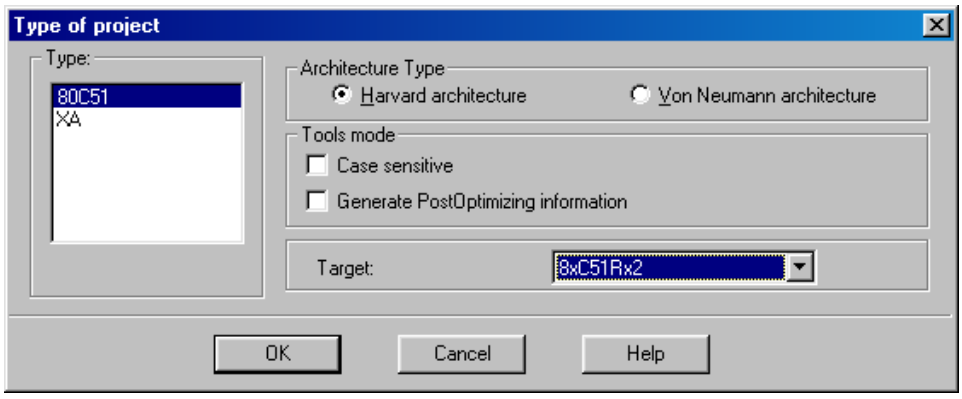
4.1.1 Creating a New Project

- Start the RIDE environment and close all projects that might be open.
- Open the ***Project*** menu and create a new project called ***Debug.prj*** within the existing project folder ***C:\PHYBasic\pC-P89C51RD2\Raisonance\Demos\Debug*** (default location) on your hard-drive. Select the 80C51 architecture for this project.
- Add ***Debug.c*** and ***Serinit.c*** from within the project folder to the project ***Debug.prj***.
- *Save* the project.

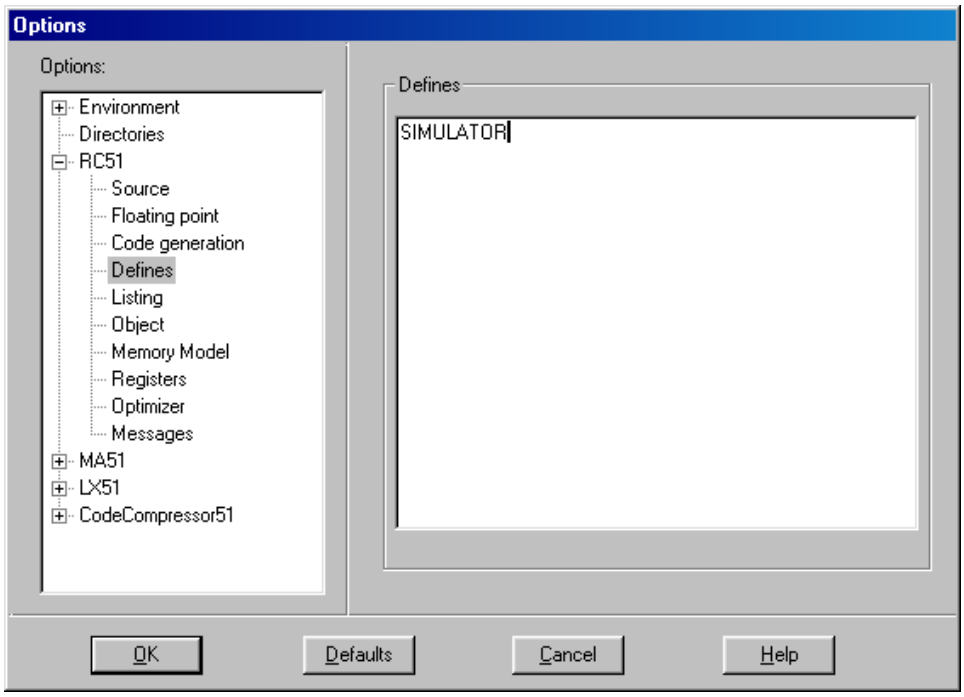
At this point you have created a project called ***Debug.prj***, consisting of the C source files ***Debug.c*** and ***Serinit.c***.

4.1.2 Setting Options for Target

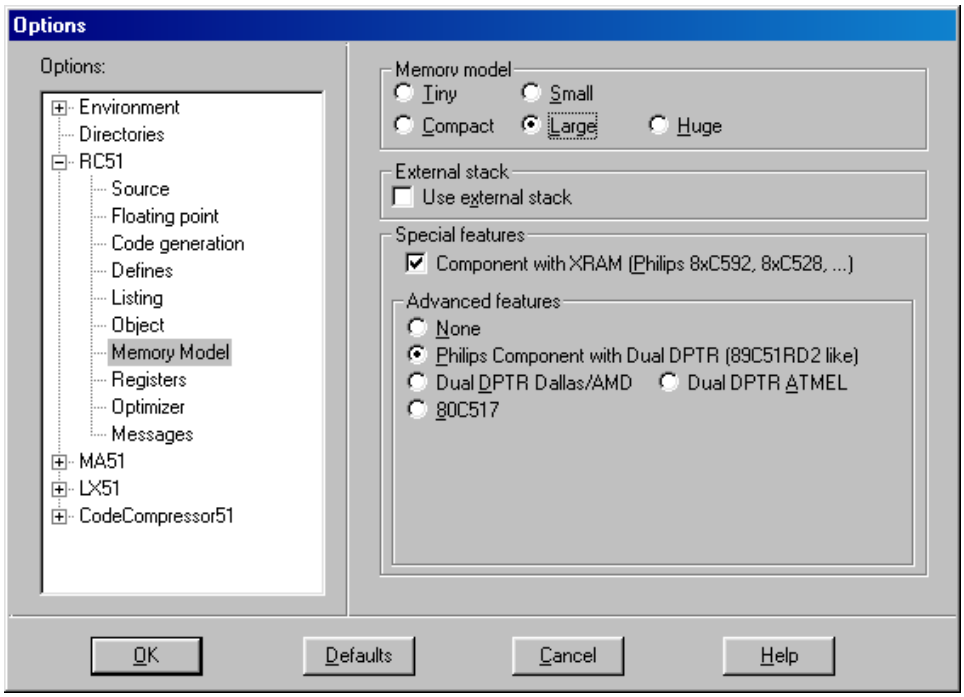
- Open the ***Options|Target*** menu, select the *8xC51Rx2* and a *Harvard architecture* as shown below:



- Click on *OK* to save the configuration.
- Open the ***Options|Project|RC51*** menu and choose *Defines*. Add **SIMULATOR** in the *Defines* input field.

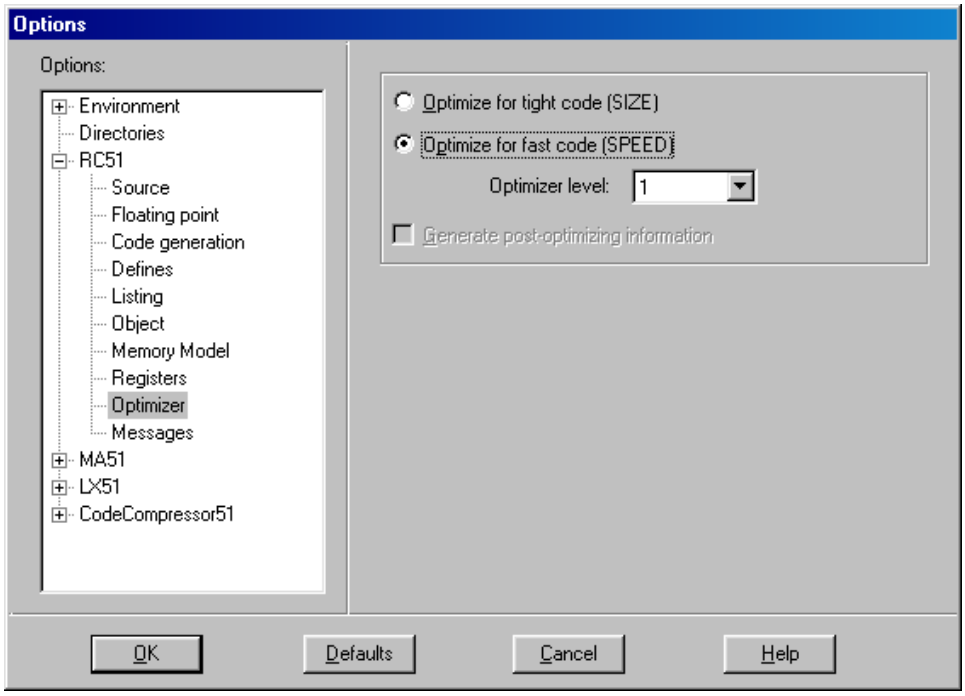


- In the ***Options|Project|RC51*** menu, now choose *Memory Model*.

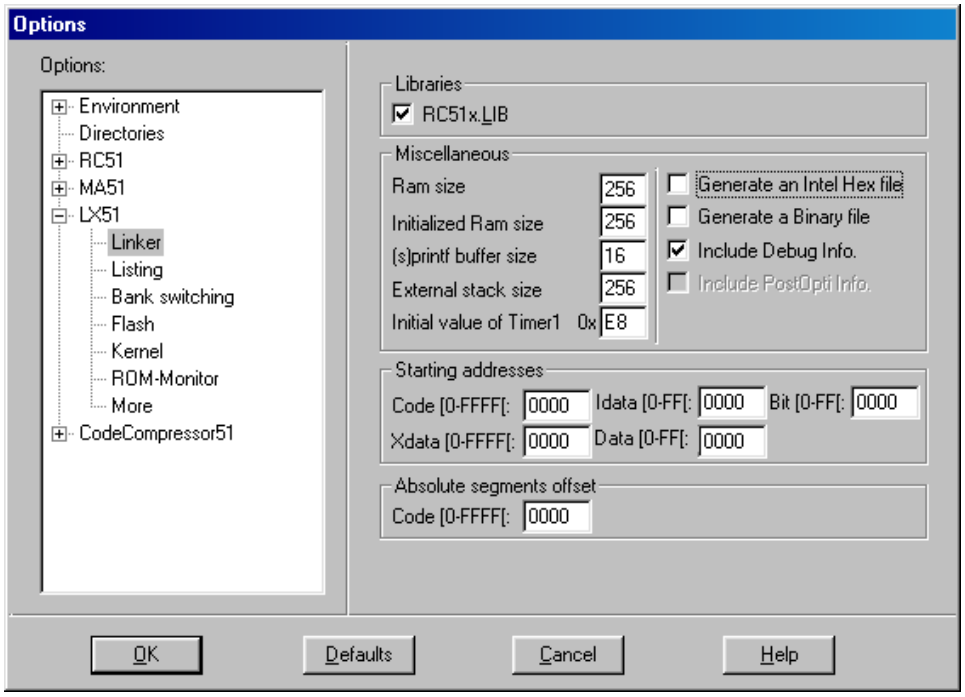


- Select the *Large* memory model. Leave all other options at their default setting.

- In the ***Options*|*Project*|*RC51*** menu, now choose *Optimizer*.
- Select the *Optimizer level 1* as shown below. This is necessary because the created ***Debug.aof*** file can be better debugged when the C file is compiled with this optimization level.



- Open the ***Options|Project\LX51*** menu and choose *Linker*.
- Check that the *Generate an Intel Hex file* checkbox is disabled. This option should be enabled by default.



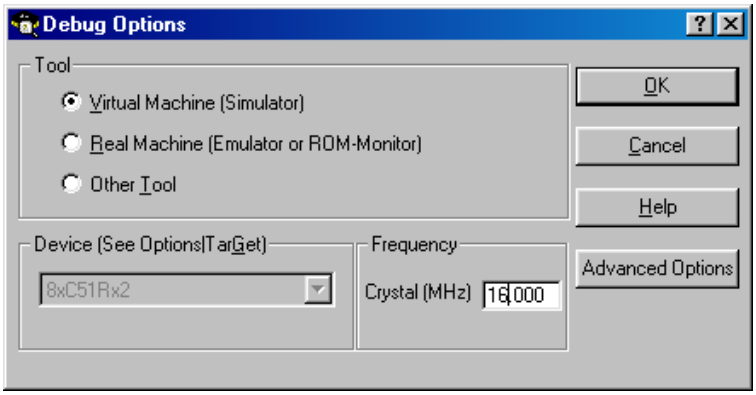
- Click on *OK* to save these settings.

The linker/locator options are now suitable for the ***Debug*** project, enabling you to build an absolute object file (****.aof***).

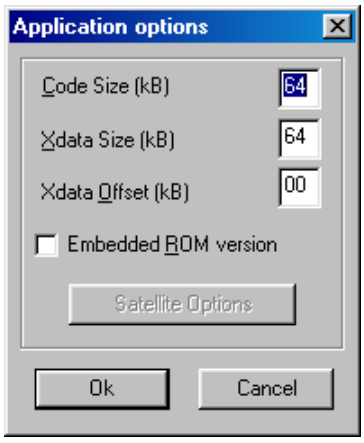
- Click on the *'Make All' Command* icon from the RIDE toolbar or open the *Project* menu and select *Build All* or *Make All*.

4.2 Preparing the Simulator

- Open the ***Options|Debug*** menu.
- Select the *Virtual Machine* as shown below:



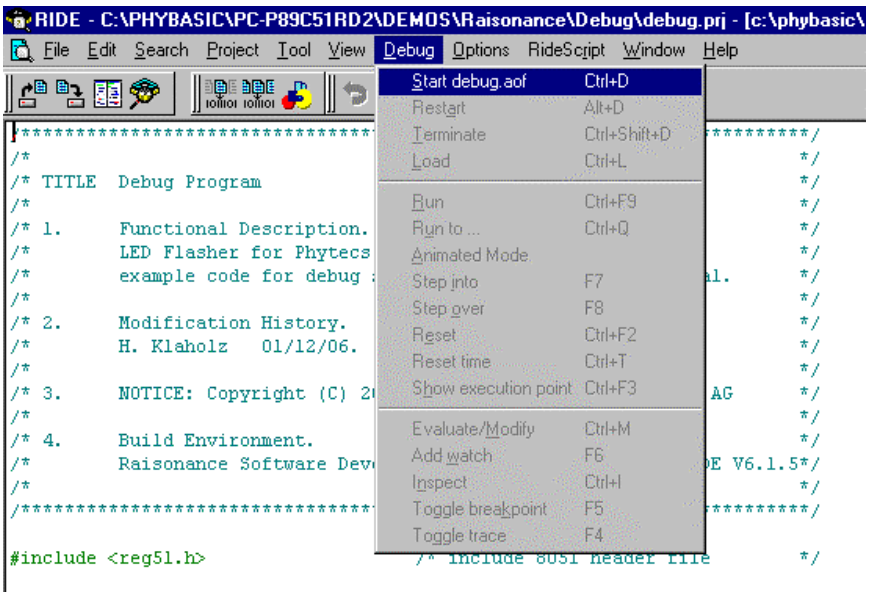
- Set the *Crystal* frequency to *16.000* MHz.
- Click on the *OK* button.
- The ***Application options*** window will appear:



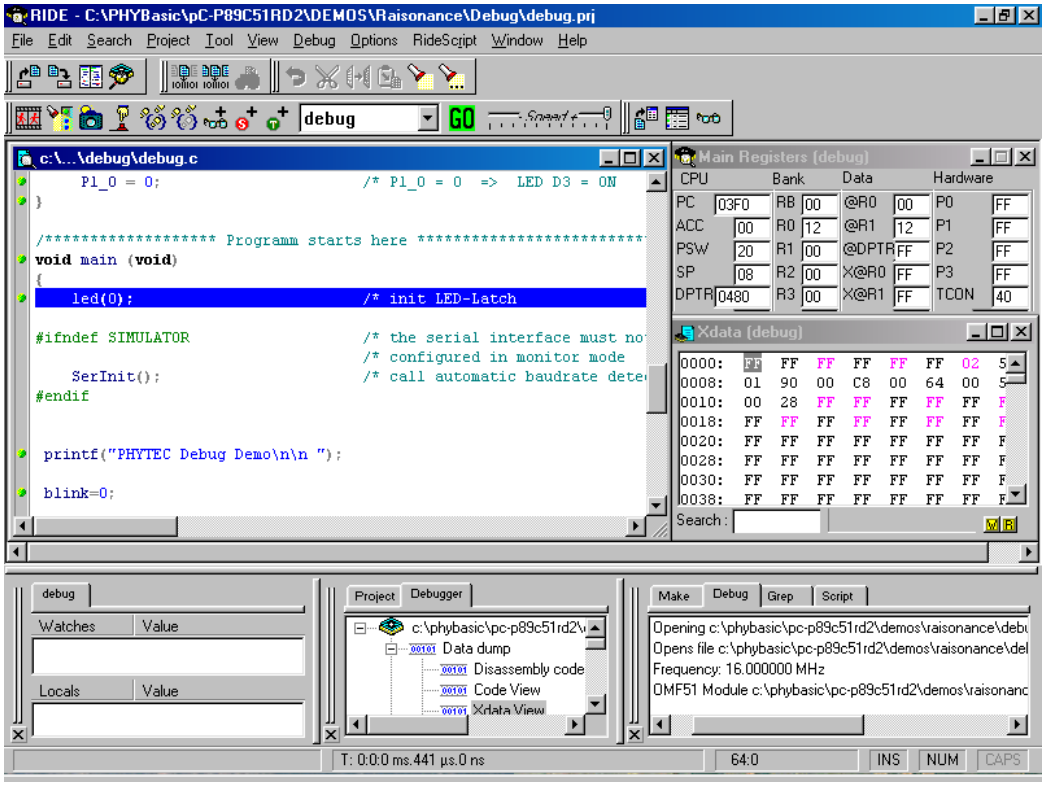
- Click on the *OK* button.
- The ***Debug Options*** window will reappear.
- Click on the *OK* button again.

4.3 Starting the Simulator

- To start the RIDE debug environment, select *Start debug.aof* in the ***Debug*** menu.



The ***Project*** window changed to the ***Debugger*** page. The debug toolbar is also displayed. In the lower part of the debug screen you will see the ***Watch*** window.



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.

4.4 The Raisonance Debug Features

- The ***Debugger*** toolbar gives access to the following debug commands: *Reset*, *Go*, *Stop*, *Step Into*, *Step Over*, *Step Out* and *Run to Cursor line*.
- RIDE uses *Step (into function calls)* to single step one line at a time. *Step (into function calls)* is also used to enter a function in the same fashion. Depending on the current window (either Disassembly or Source), the meaning of "Stepping" will be slightly different, and automatically adapted to the context. In a Source window, stepping will be performed at the source level (e.g. line to line). In a Disassembly window, stepping will be performed at the instruction level. To *Step into*, click on the  button, or press <F7>, or open the *Debug* | *Step Into* menu.
- *Step (over function calls)* means to skip over a function that you are not interested in. To *Step over*, click on the  button, or press <F8>, or open the *Debug* | *Step Over* menu.
- To reach the cursor location, open the *Debug* | *Run to* menu.
- You can reset the application by clicking on the *Reset application*  button in the debug toolbar. The program will arrive at **`main()`** when the Reset is performed from a *Source* window, or at the reset vector if it is performed from the *Code disassembly* window.
- The *Go*  icon will change into a *Stop*  icon during the program execution.

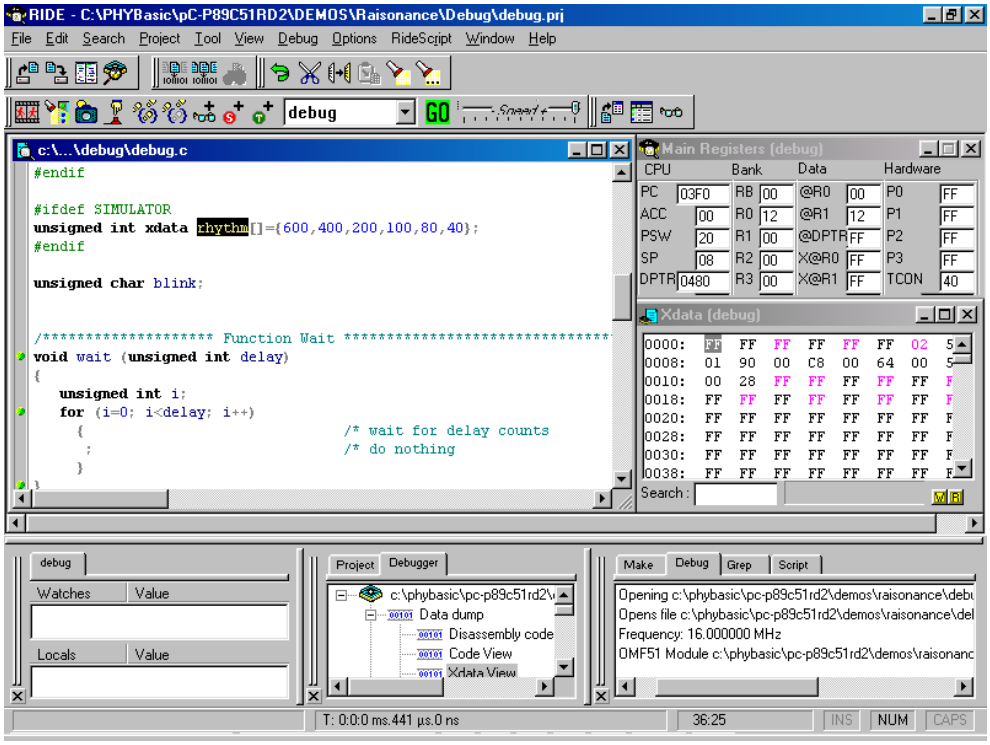
  Clicking the *Go* icon runs the program without active debug functions. To stop program execution at a desired point, a breakpoint can be placed before the *Go* icon is clicked.

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

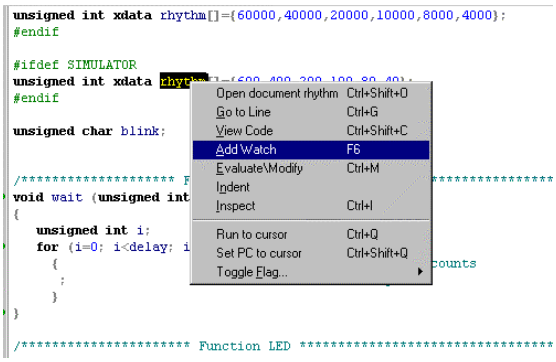
4.5 Using the Raisonance Debug Features

4.5.1 Watch Window

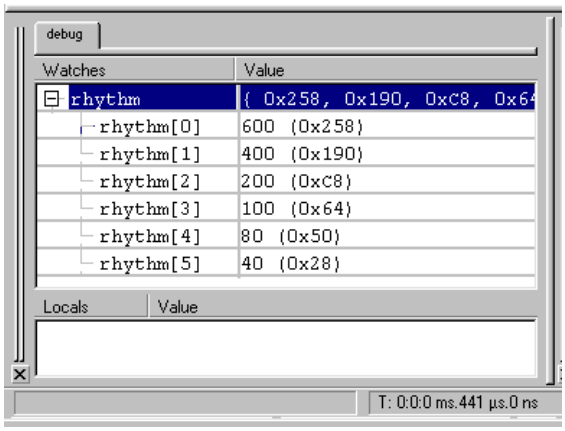
- Go to the code line where the constant ***rhythm*** is defined below the line *#ifdef SIMULATOR*. Select the constant by double-clicking on the constant name.



- Right-click on the constant ***rhythm*** and select *Add Watch* in the pop-up window. You may also use the <F6> function key to enable the ***Add Watch*** window.

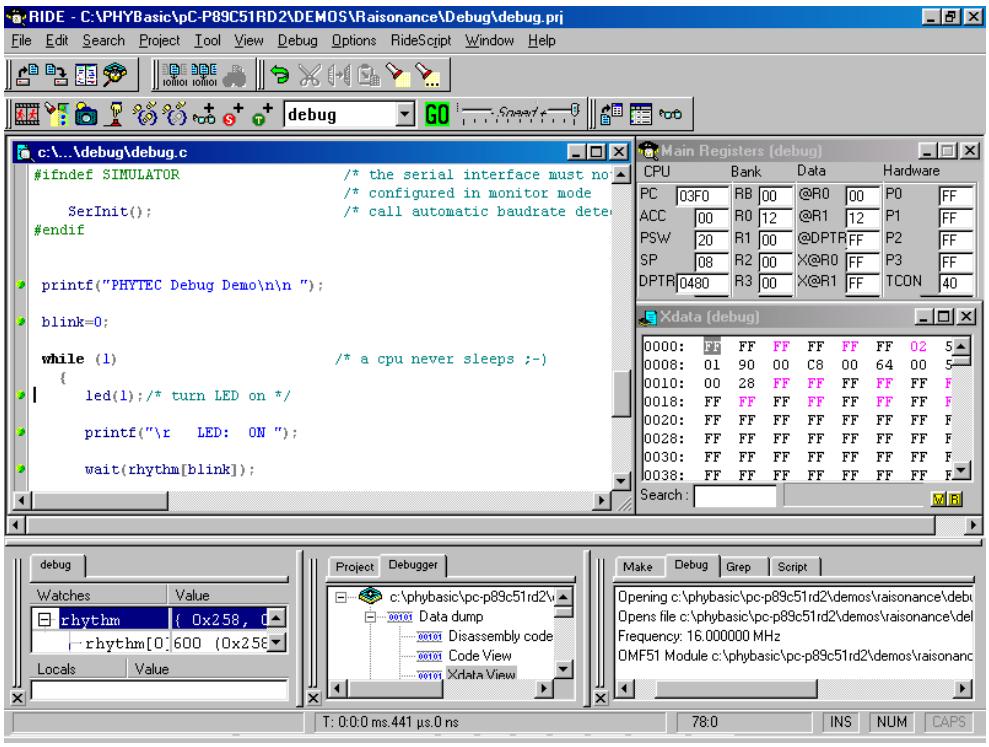


- The ***Watch*** window now shows the constant "**rhythm[]**". The small ⊞ sign in front of **rhythm** indicates that this is an array with a group of array elements. Click the ⊞ sign to expand the view and to see all array elements of "**rhythm[]**".

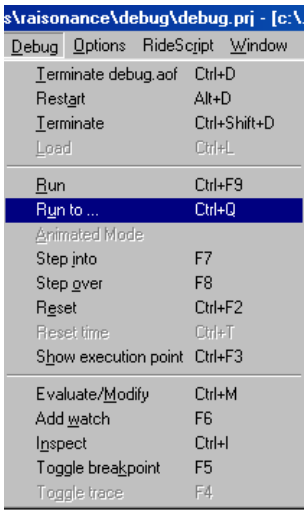


4.5.2 Run to ...

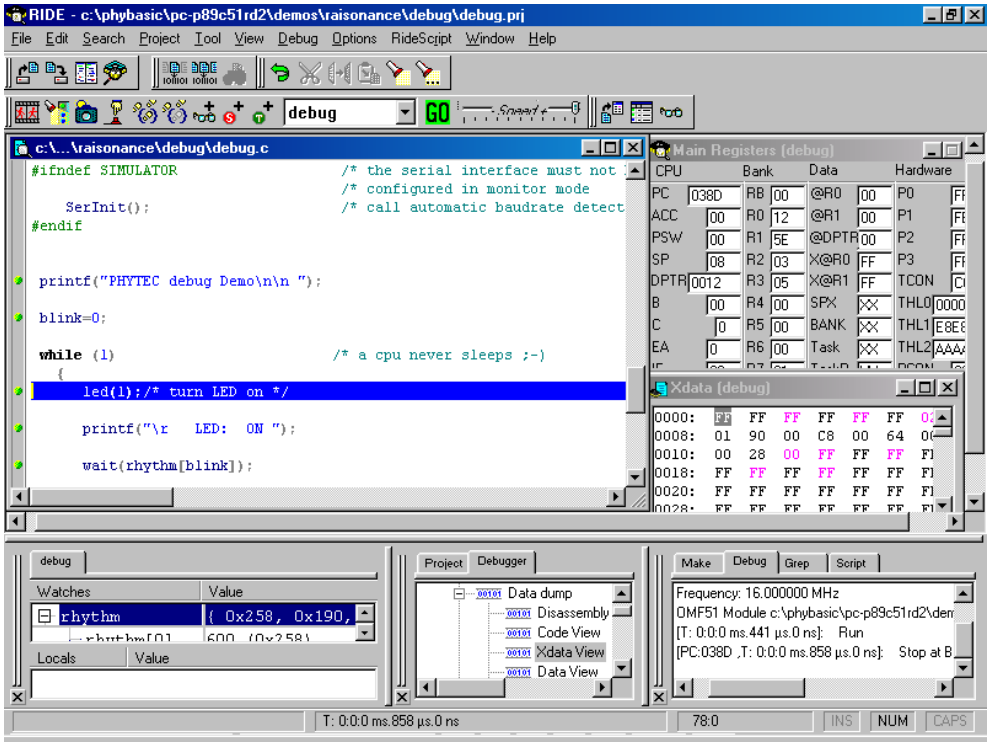
- The ***Run to...*** command executes the program until it reaches the code line where the cursor is currently located. Go with the cursor to the code line *led(1);*.



- To run the program and stop at the selected code line, select ***Run to...*** in the ***Debug*** menu.

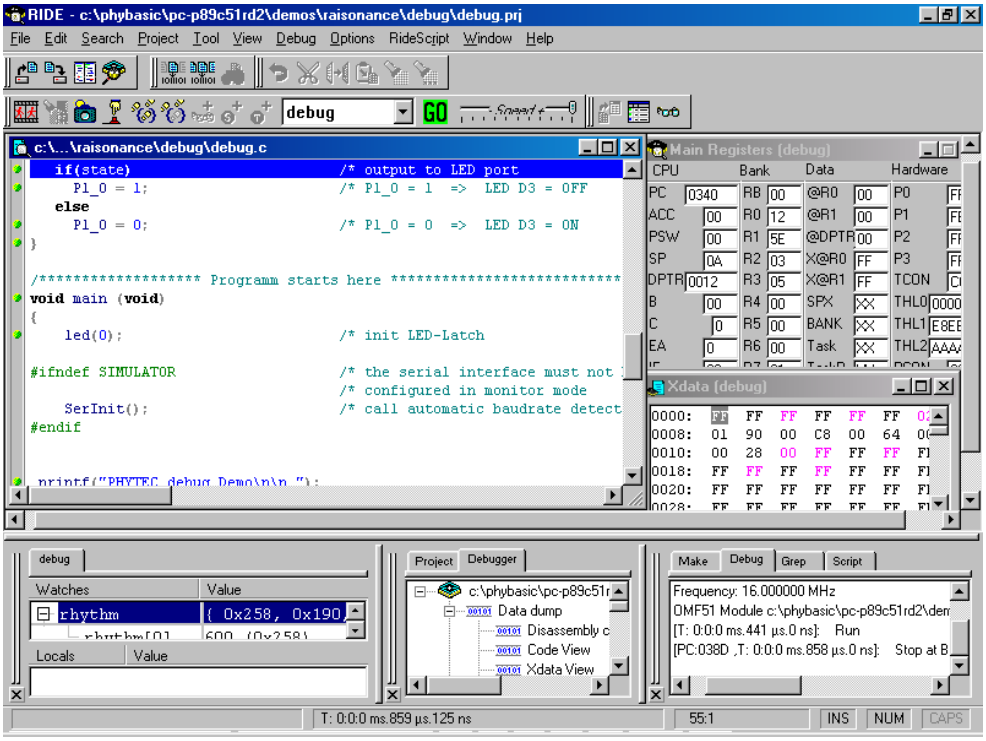


- The Debugger executes the program until it reaches the code line where the cursor is currently located. This code line is now highlighted in blue color as shown below:



4.5.3 Step Into and Step Over

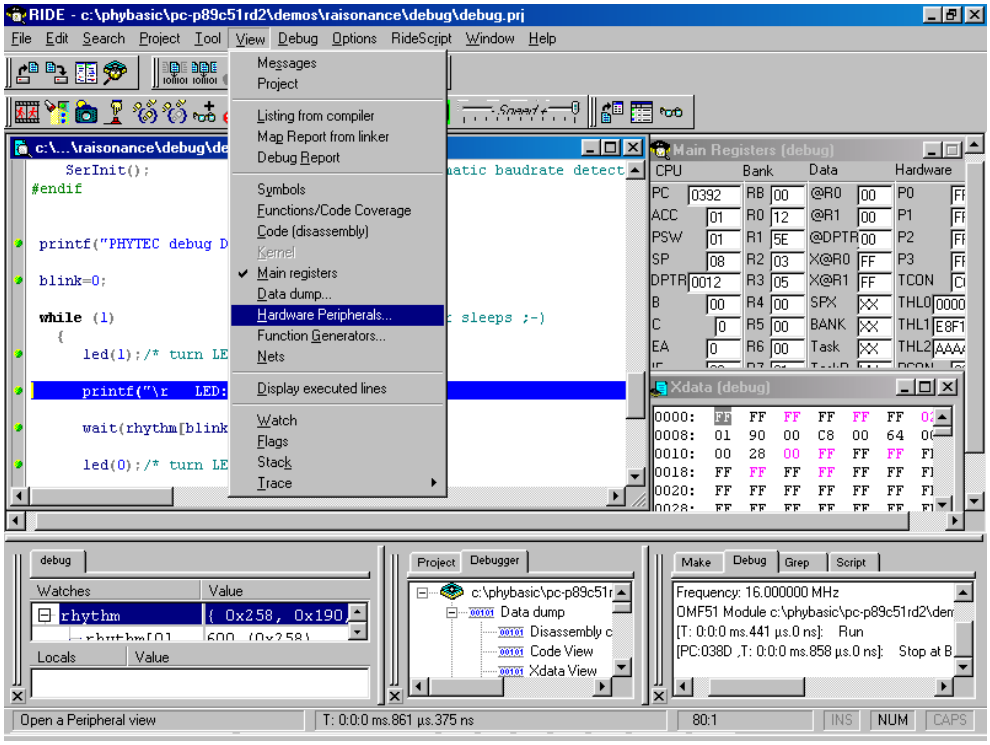
- Click on the ***Step Into*** icon to enter the *'led()'* function.



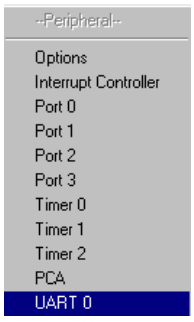
- Now click on the ***Step Over*** icon twice to single-step through the *'led()'* function.

4.5.4 Hardware Peripherals

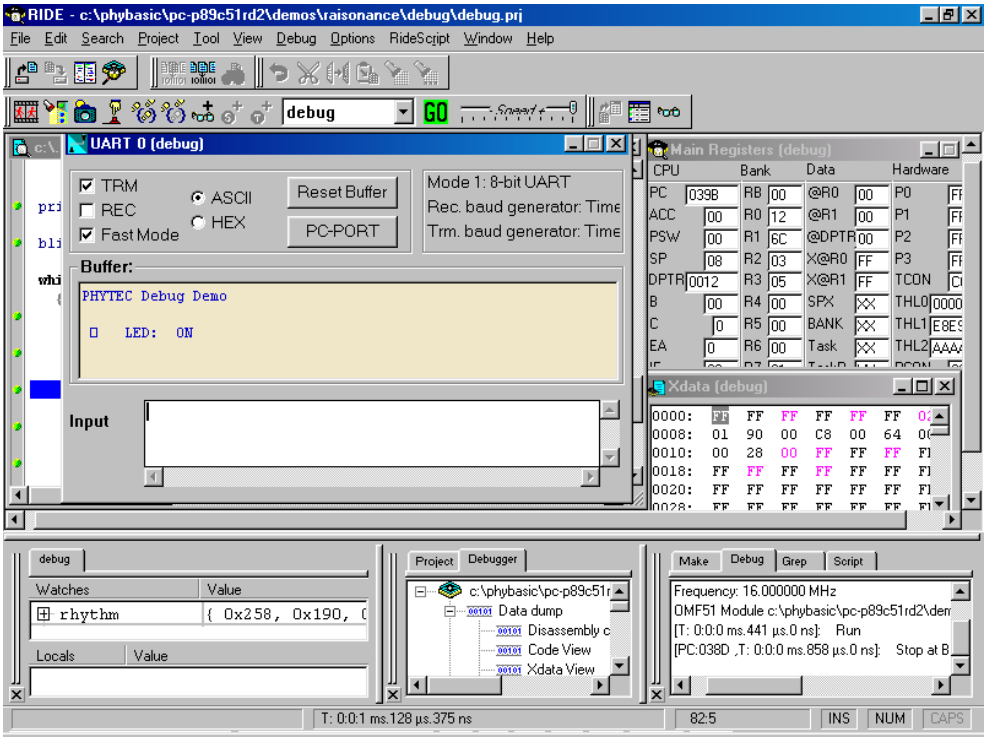
- Open the ***View|Hardware Peripherals*** menu.



- Select *UART0* in the pop-up window as shown below:

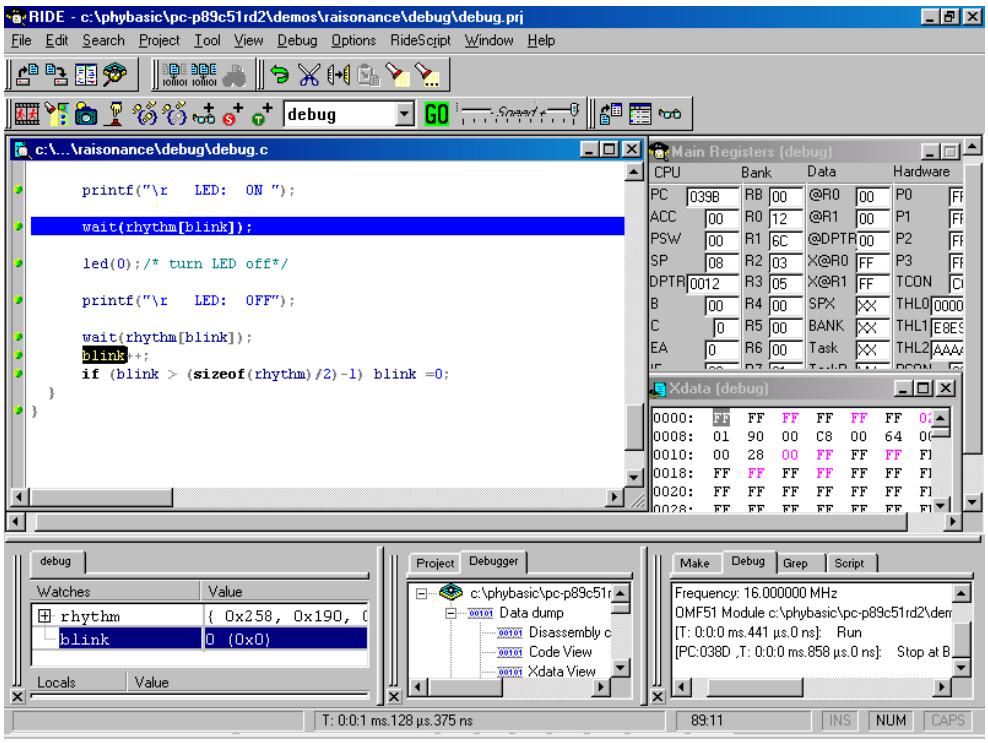


- The UART 0 window will now appear. You can see the string *PHYTEC Debug Demo* in the Buffer section of the screen because the first *printf* instruction has been executed already.
- Now click on the ***Step Over*** icon and watch the UART 0 window.

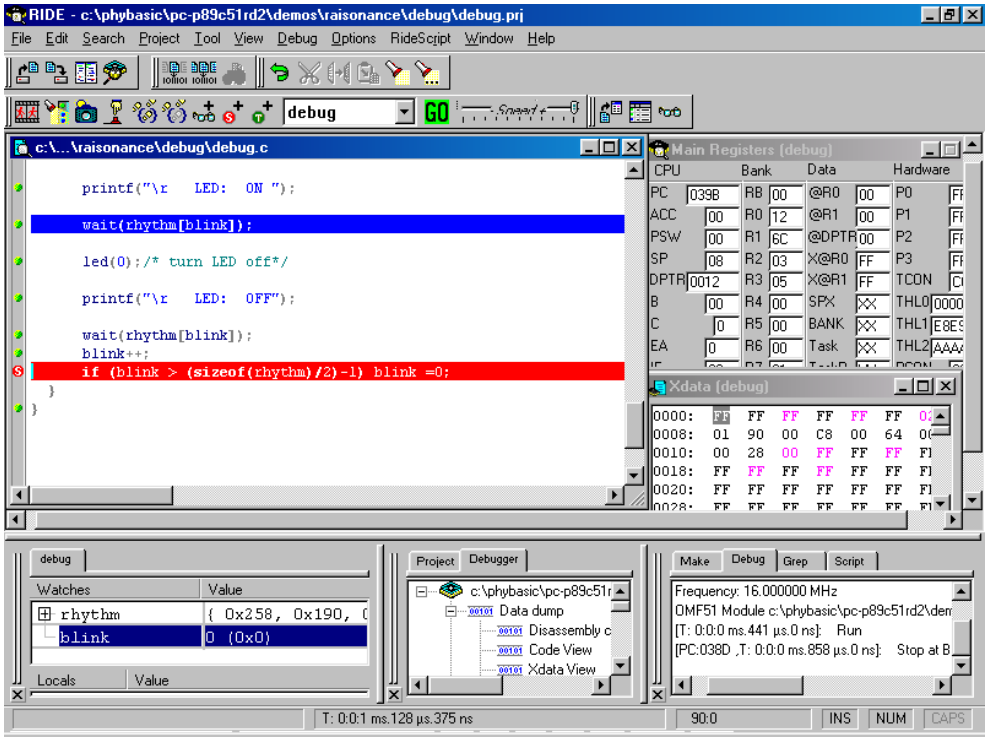


4.5.5 Breakpoints

- Select the variable *blink* and add *blink* to the ***Watch*** window.



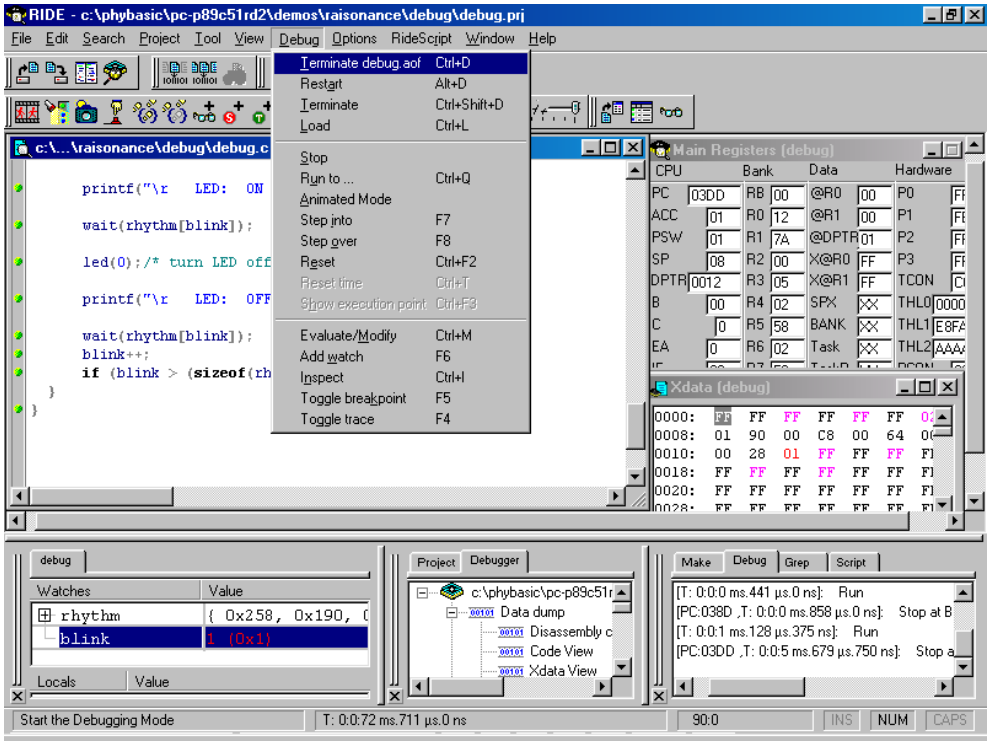
- Click on the green icon in front of the code line *if(blink >...)* to set a breakpoint here.
- The red marker on the left-hand side of the selected line indicates the breakpoint.



- Click on the *Go* GO icon and the program will run and stop at the breakpoint.
- Also notice that the variable *blink* in the Watch window has changed its value to *'1(0x1)'*.

4.5.6 Running, Stopping and Resetting

- To run your program without stopping at any time, delete all breakpoints by clicking on the red icon in front of the code line.
- Click on the *Go* GO icon.
- To exit the current debug session go to the ***Debug*** menu and click on ***Terminate debug.aof***.



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.

- Open the ***Options|Project|RC51*** menu and choose *Defines*. Delete the **SIMULATOR** define in the *Defines* input field. This will include various *printf* statements in the application program that can be viewed with a terminal emulation program. Use of the *printf* statements is now possible because the serial interface is no longer required for other communication tasks.
- Click on *OK* to save this setting.
- Open the ***Options|Project\LX51*** menu and choose *Linker*.
- Enable the *Generate an Intel Hex file* checkbox.
- The linker/locator options are now suitable for the ***Debug*** project, enabling you to build an absolute object file (****.aof***) and a hexfile.
- Click on the *'Make All' Command* icon  from the RIDE toolbar or open the *Project* menu and select *Build All* or *Make All*.
- Download the created ***Debug.hex*** file (located in ***C:\PHYBasic\pC-P89C51RD2\Demos\Raisonance\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.
- The application is now waiting for receipt of a known character over the serial interface. Start the HyperTerminal program and push the <Space> bar as described in *section 2.4.2* This starts the automatic baud rate detection. Now you can 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 Raisonance 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 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 Raisonance 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-588e_2, 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

PHYTEC

Ordering No. L-588e_2
Printed in Germany