

# phyCORE<sup>®</sup>-LPC2294

## QuickStart Instructions

Using Keil's ULINK and the Keil ARM7/ $\mu$ Vision3  
Software Development Tool Chain

Note: The PHYTEC Spectrum CD includes the electronic version of  
the English phyCORE<sup>®</sup>-LPC2292/94 Hardware Manual

**Edition: June 2005**

A product of a PHYTEC Technology Holding company

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

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

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

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

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

	EUROPE	NORTH AMERICA
Address:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Ordering Information:	+49 (800) 0749832 <a href="mailto:order@phytec.de">order@phytec.de</a>	1 (800) 278-9913 <a href="mailto:sales@phytec.com">sales@phytec.com</a>
Technical Support:	+49 (6131) 9221-31 <a href="mailto:support@phytec.de">support@phytec.de</a>	1 (800) 278-9913 <a href="mailto:support@phytec.com">support@phytec.com</a>
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Site:	<a href="http://www.phytec.de">http://www.phytec.de</a>	<a href="http://www.phytec.com">http://www.phytec.com</a>

3<sup>rd</sup> Edition: June 2005

---

<b>1</b>	<b>Introduction to the Rapid Development Kit .....</b>	<b>1</b>
1.1	Rapid Development Kit Documentation .....	1
1.2	Overview of this QuickStart Instruction.....	2
1.3	System Requirements .....	3
1.4	The PHYTEC phyCORE®-LPC2294 .....	4
1.5	Keil ARM7/μVision3 Software Development Tool Chain .....	7
<b>2</b>	<b>Getting Started.....</b>	<b>9</b>
2.1	Installing Rapid Development Kit Software.....	9
2.2	Interfacing the phyCORE®-LPC2294 to a Host-PC.....	15
2.3	Downloading Example Code with Philips LPC2000 Flash Utility .....	18
2.4	Downloading Example Code with μVision3.....	21
2.4.1	"Blinky" .....	22
2.4.2	"Hello" .....	25
<b>3</b>	<b>Getting More Involved .....</b>	<b>31</b>
3.1	Creating a New Project and Adding an Existing Source File.....	31
3.1.1	Configure the Project Components.....	33
3.1.2	Adding Source Files to the Project.....	37
3.2	Modifying the Source Code.....	40
3.3	Setting Options for Flash Target .....	41
3.4	Building the Project .....	52
3.5	Downloading Code into Flash Memory.....	53
<b>4</b>	<b>Debugging.....</b>	<b>55</b>
4.1	Creating a Debug Project and Preparing the Debugger.....	56
4.1.1	Creating a New Project.....	56
4.1.2	Setting Options for Target.....	57
4.2	Starting the Debugger.....	65
4.3	Keil μVision3 Debug Features .....	67
4.4	Using the Keil μVision3 Debug Features.....	69
4.4.1	Breakpoints.....	69
4.4.2	Single Stepping and Watch Window .....	70
4.5	Running, Stopping and Resetting .....	71
4.6	Changing Target Settings for the "Executable Version" .....	72

## Index of Figures

Figure 1: Default Jumper Settings of the phyCORE® Development Board HD200 with phyCORE®-LPC2294 .....	15
Figure 2: JTAG Connector X701 on the phyCORE®-LPC2294 (Bottom View).....	16
Figure 3: ULINK Connected to the phyCORE® Module .....	17
Figure 4: Power Connector .....	17

# 1 Introduction to the Rapid Development Kit

## This QuickStart provides:

- general information on the PHYTEC phyCORE<sup>®</sup>-LPC2294 Single Board Computer (SBC),
- an overview of Keil's ARM7/μVision3 software development tool chain evaluation version, and
- instructions on how to run example programs on the phyCORE<sup>®</sup>-LPC2294, mounted on the PHYTEC phyCORE<sup>®</sup> Development Board HD200, in conjunction with the Philips LPC2000 Flash Utility for internal Flash, the Keil ULINK and software tools

Please refer to the [phyCORE<sup>®</sup>-LPC2292/94 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<sup>®</sup>-LPC2292/94 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<sup>®</sup>-LPC2292/94 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<sup>®</sup>-LPC2294 in conjunction with the Keil ARM7/μVision3 software development tool chain

## 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<sup>®</sup>-LPC2294 in conjunction with the Keil ULINK and ARM7/μVision3 software tools. It is structured as follows:

- 1) The "*Getting Started*" section uses three examples:  
*Blinky* to demonstrate the download of user code to the internal Flash device using the Philips LPC2000 Flash Utility  
*Blinky* and *Hello* to demonstrate the download of user code to the external Flash device using the Keil ULINK and ARM7/μVision3 software tools
- 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<sup>®</sup>-LPC2294 using the Keil tools.
- 3) The "*Debugging*" section provides a third example program - "Debug" - to demonstrate simple debug functions using the Keil μVision3 debug environment.

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

### 1.3 System Requirements

Use of this "Rapid Development Kit" requires:

- the PHYTEC phyCORE<sup>®</sup>-LPC2294
- the phyCORE<sup>®</sup> Development Board HD200 with the included DB-9 serial cable and AC adapter supplying 5 VDC /min. 500 mA
- the Philips LPC2000 Flash Utility software
- the Keil ULINK JTAG-USB adapter, **not** included in the standard Rapid Development Kit version<sup>1</sup>
- the PHYTEC Spectrum CD for ARM7
- an IBM-compatible host-PC (486 or higher running at least Windows95/NT)

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](mailto:support@phytec.com) - or - [support@phytec.de](mailto:support@phytec.de)



<http://www.keil.com>  
[support@keil.com](mailto:support@keil.com)

---

<sup>1</sup> : The Keil ULINK is included in the Rapid Development Kit version with the part number KPCM-023-SK-2294-Keil.

---

## **1.4 The PHYTEC phyCORE<sup>®</sup>-LPC2294**

The phyCORE<sup>®</sup>-LPC2294 represents an affordable yet highly functional Single Board Computer (SBC) solution in sub-miniature dimensions (60 x 53 mm). It is intended for use in memory-intensive applications running within a multi-node CAN bus system. The standard module is populated with a Philips LPC2294 controller.

All applicable data/address lines and signals extend from the underlying logic devices to two high-density Molex SMT pin header connectors (pin width is 0.635 mm/25 mil) lining the circuit board edges. This enables the phyCORE<sup>®</sup>-LPC2294 to be plugged like a "big chip" into target hardware.

The standard module runs at a 60 MHz internal clock speed (delivering 50/10 ns instruction cycle) and offers 1 MByte (up to 8 MByte) SRAM and 2 MByte (up to 16 MByte) Flash on-board for DATA and CODE storage.

The module communicates by means of two RS-232 transceivers, two (four optional) CAN bus interfaces, and a SMSC LAN91C111 10/100BaseT Ethernet controller which enables implementation of the module in embedded Internet applications. The phyCORE<sup>®</sup>-LPC2294 operates within a temperature range of -40°C to +85°C and requires only a 300 mA power source.

The Keil ARM/μVision3 software tools, in conjunction with the Keil ULINK adapter, enables easy on-board download of user programs.



**The phyCORE<sup>®</sup>-LPC2294 offers the following features:**

- subminiature Single Board Computer (60 x 53 mm) achieved through modern SMD technology
- populated with the Philips LPC2294 microcontroller (TQPF-144 packaging)
- improved interference safety achieved through multi-layer PCB technology and dedicated Ground pins
- controller signals and ports extend to two 100-pin high-density (0.635 mm) Molex connectors aligning two sides of the board, enabling it to be plugged like a "big chip" into target application
- 32-bit, demultiplexed bus mode
- max. 60 MHz clock frequency (ca. 50 ns instruction cycle, external; ca. 10 ns internal Flash)
- 1.5 Gbyte external address space
- 2 MByte (up to 16 MByte) on-board Flash<sup>1</sup>
- on-board Flash programming, no dedicated Flash programming voltage required through use of 3.3 V Flash devices
- 1 MByte (up to 8 MByte) RAM on-board, max. 2 MByte at 0 wait states<sup>1</sup>
- up to two CAN transceivers (Infineon TLE6250V33)
- RS-232 transceiver for two serial interfaces
- SMSC 91C111 Ethernet controller with configuration EEPROM
- 2 kByte (up to 8 kByte) SPI-EEPROM<sup>1</sup>
- I<sup>2</sup>C Real-Time Clock with internal quartz (can be battery buffered)
- one operating voltage for core & peripherals, 3.3 V, typ. < 280 mA
- controller 1.8 V core voltage generated on-board
- additional 5 V operating voltage for CAN transceivers, typ. <12 mA
- support of LPC2294 single chip mode

---

<sup>1</sup>: Please contact PHYTEC for more information about additional module configurations.

---

The phyCORE<sup>®</sup> Development Board HD200, 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 most PHYTEC phyCORE<sup>®</sup> high-density series Single Board Computers. Simple jumper configuration readies the Development Board's connection to the phyCORE<sup>®</sup>-LPC2294, which plugs into the receptacle contact strips mounted on the Development Board HD200.

### **phyCORE<sup>®</sup> Development Board HD200 Technical Highlights**

- low voltage socket for supply with regulated input voltage 5 VDC
- additional supply voltages 3.3 VDC or 2.5 VDC
- two DB-9 sockets (P1A, P1B) configurable as RS-232 interfaces
- two additional DB-9 plugs (P2A, P2B) configurable as CAN interfaces
- simple jumper configuration allowing use of the phyCORE<sup>®</sup> Development Board HD200 with various PHYTEC phyCORE<sup>®</sup> high-density SBCs
- socket for RJ45 Ethernet transformer module
- one control LED (D3) for quick testing of user software
- 2x 160-pin Molex connector (X2) enabling easy connectivity to expansion boards (e.g. PHYTEC GPIO Expansion Board)

## 1.5 Keil ARM7/ $\mu$ Vision3 Software Development Tool Chain

Keil Software development tools for the ARM7 TDMI Architecture support every level of developer from the professional applications engineer to the student just learning about embedded software development. The Keil ARM7 compiler supports all ARM7-compatible devices including the Philips LPC2000 devices. For a complete list of supported ARM7 derivatives go to:

<http://www.keil.com/dd/arm7chips.asp>

$\mu$ Vision3, the latest version of Keil's popular IDE, combines project management, source code editing, program debugging, and Flash programming in a single, powerful environment. This QuickStart provides an overview of the most commonly used  $\mu$ Vision3 features including:

- Project management, device setup, and tool configuration
- Integrates Keil ARM development tools in a single graphical user interface (GUI)
- Editor facilities for creating, modifying, and correcting programs
- JTAG/target debugging or CPU & peripheral simulation

Once installed, the default destination location for all ARM7 tools; executables; include, header and example files; as well as online help and documentation is the **C:\Keil\Arm** folder, while the  $\mu$ Vision3 IDE is located at **C:\Keil\Uv3**. You can start Keil  $\mu$ Vision3 by selecting it from the *Programs* menu using the Windows *Start* button. The Keil  $\mu$ Vision3 icon will also be placed on your desktop for easy startup of the development tools.

The new  $\mu$ Vision3 editor offers many standard and advanced software editing features like:

- **Incremental Find** positions the cursor while you type the search phrase.
- **Active Brace Checking** shows nesting and highlights mismatches while entering parentheses, braces, or brackets.
- **Text Block Functions** that comment, indent, uppercase, tabify, and remove whitespace from text blocks.
- **Document-Selective Settings** for tab spacing and syntax coloring in Assembler, C, and other file types.
- Detailed **Syntax Highlighting** that allows you to define a user keyword list. Colors are used in printed output.
- Document **Outlining** that provides a quick overview of complex source files.

The ARM7 evaluation version, as provided on the PHYTEC Spectrum CD, has the following limitations:

- The  $\mu$ Vision Debugger is limited to 16 kBytes.
- You may not use the Evaluation Version of the  $\mu$ Vision IDE/Debugger to create commercial products.
- The GNU ARM tools (compiler, assembler, and so on) that are provided are not limited or restricted in any way.

The full DKARM Developer's Kit can be purchased through PHYTEC. Please contact our sales representatives for a quote.

For more information on Keil ARM7/ $\mu$ Vision3 tools visit their website at:

<http://www.keil.com/arm/>

## 2 Getting Started

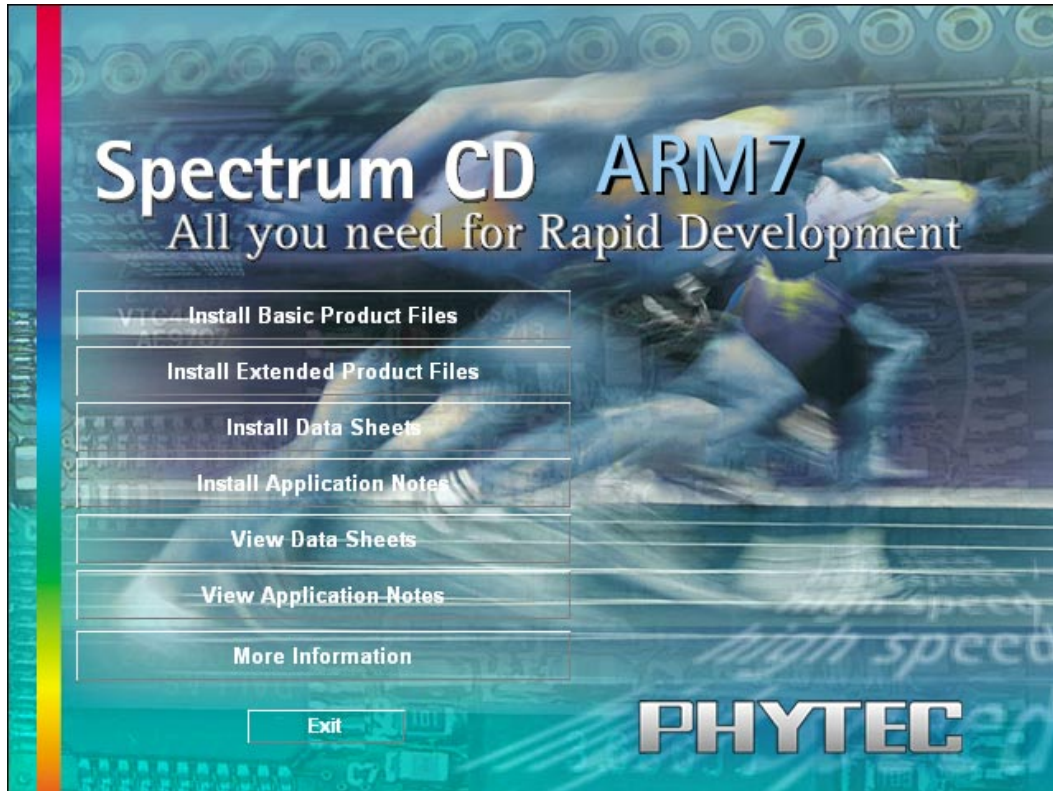
What you will learn with this Getting Started example:

- installing Rapid Development Kit software
- interfacing the phyCORE<sup>®</sup>-LPC2294, mounted on the Development Board, to a host-PC using the Keil ULINK
- downloading example user code in hex-file format from a host-PC to the internal Flash memory using the Philips LPC2000 Flash Utility
- downloading example user code from a host-PC to the external Flash memory using ARM7/μVision3 tools

### 2.1 Installing Rapid Development Kit Software

When you 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.

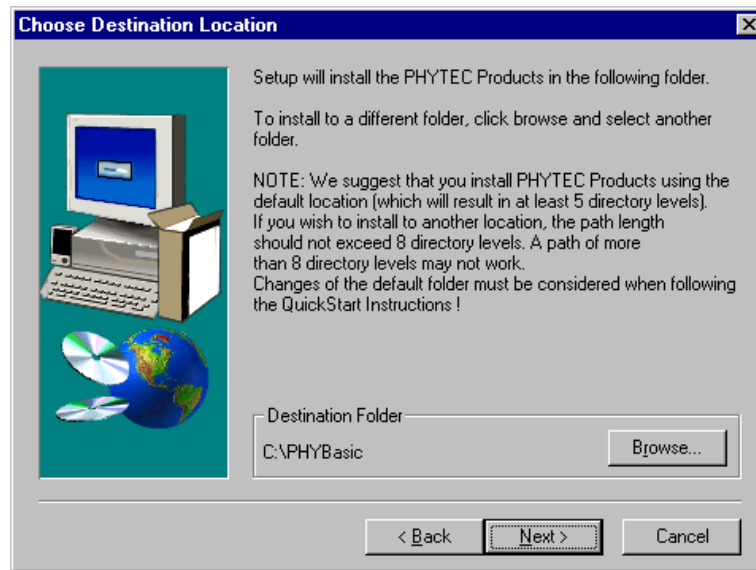
The following window appears:



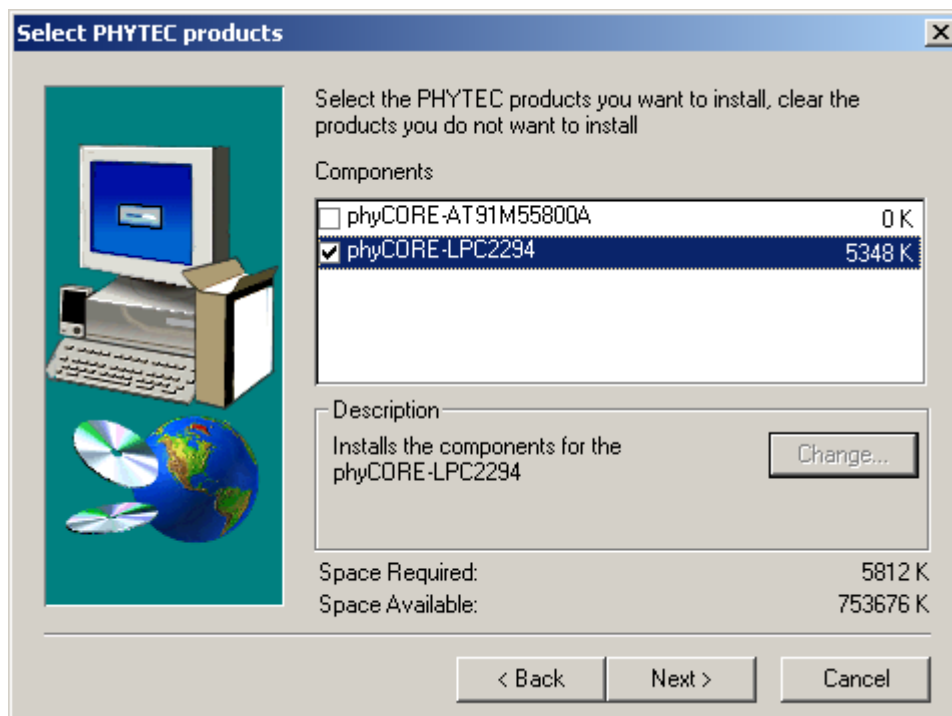
- Choose *Install Basic Product Files* Button.
- After accepting the *Welcome* window and license agreement select the destination location for installation of Rapid Development Kit software and documentation.

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.



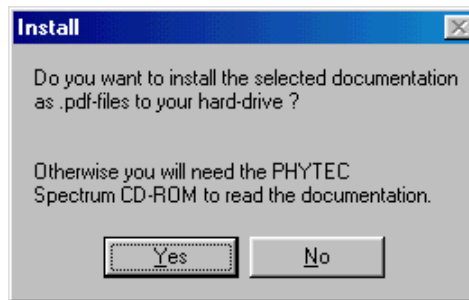
- In the next window select your Rapid Development Kit of choice from the list of available products.



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

All software and tools for this phyCORE<sup>®</sup>-LPC2294 RDK will be installed to the **|PHYBasic** directory 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.

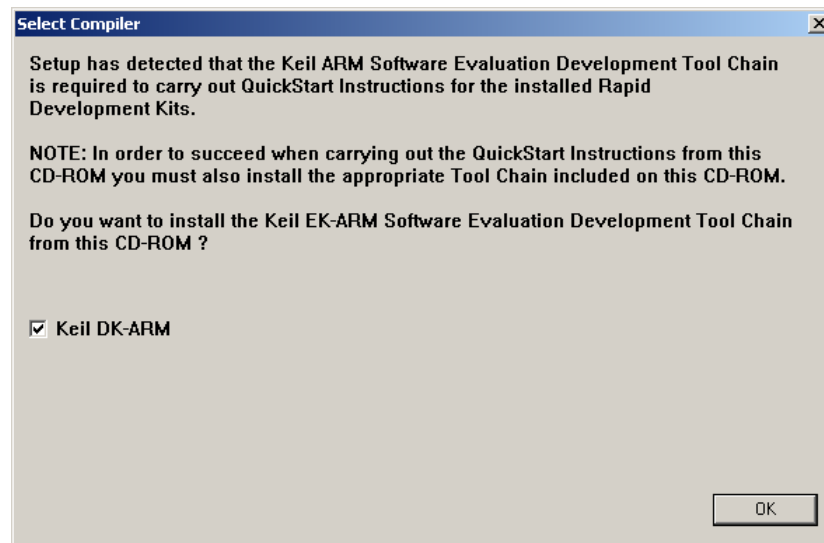


If you decide **not** to copy the documentation to your hard-drive you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.



If you decide to copy the electronic documentation to your hard-drive, the documentation for this phyCORE<sup>®</sup>-LPC2294 RDK will also be installed to the Kit-specific subdirectory.

- Setup will now add program icons to the program folder, named *PHYTEC*.
- In the next window, choose the Keil ARM7/ $\mu$ Vision3 Software Development Tool Chain.



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

We recommend that you install the Keil ARM7 tools and  $\mu$ Vision3 from the Spectrum CD-ROM even if other versions of  $\mu$ Vision3 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<sup>1</sup>.

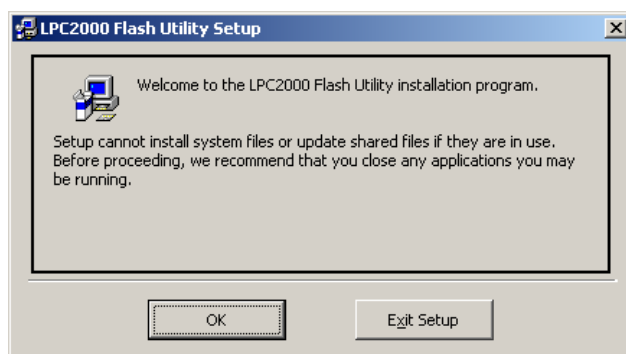
---

<sup>1</sup> : **Note:** If you have a full version of the Keil ARM7 tools already installed on your host-PC we recommend installing this evaluation version on a different desktop or laptop computer. If this is not possible we recommend to use the same version of the Keil ARM7 tools that we use in this QuickStart manual. PHYTEC can not guarantee successful completion of these QuickStart instructions if a different version of the Keil ARM7 tools is used.

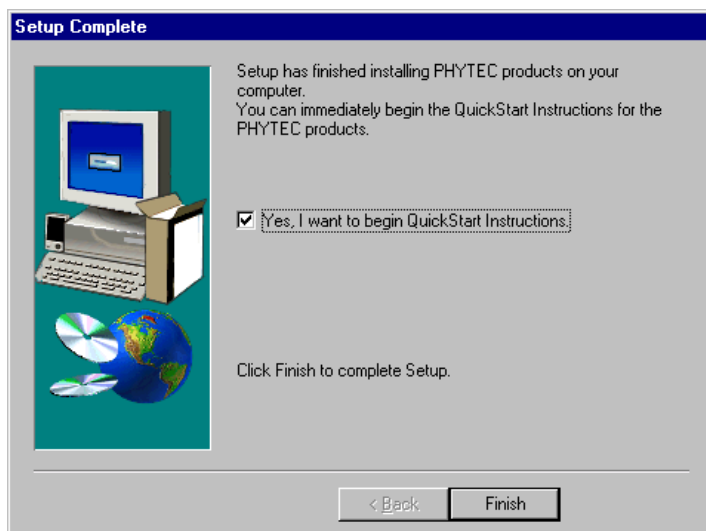
- After accepting the *Welcome* window and license agreement select the destination location for installation of the Development Tool Chain.

The applicable Keil ARM7/ $\mu$ Vision3 evaluation development tool chain will be installed to your hard-drive.

In the following windows you can decide to install Philips LPC2000 Flash Utility software and the Acrobat Reader. The LPC2000 Flash Utility 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.



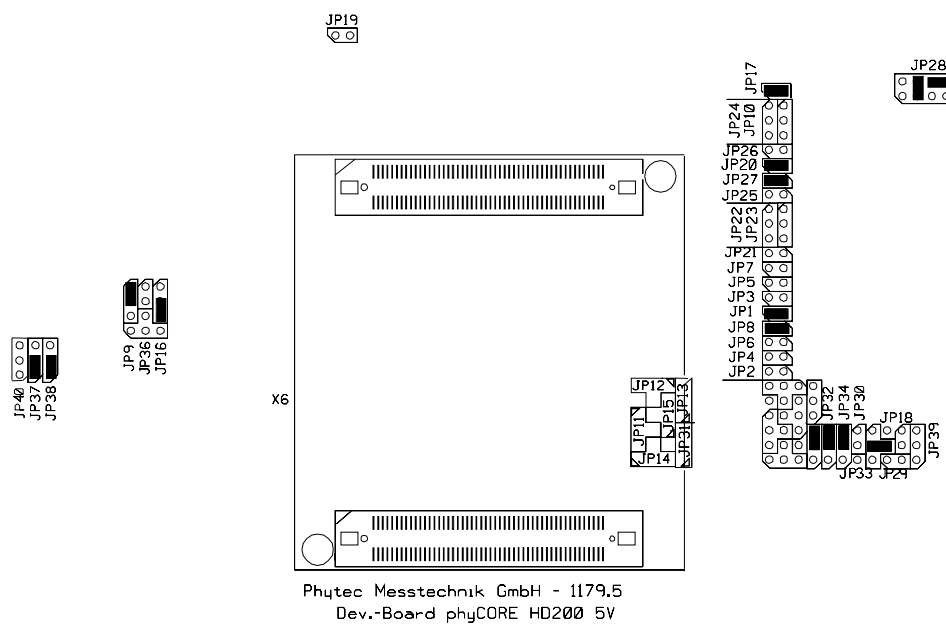
- Click *OK* and follow the Flash Utility Setup instructions.
- Decide if you want to begin the QuickStart Instruction immediately by selecting the appropriate checkbox and click on *Finish* to complete the installation.



## 2.2 Interfacing the phyCORE<sup>®</sup>-LPC2294 to a Host-PC

Connecting the phyCORE<sup>®</sup>-LPC2294, mounted on the phyCORE<sup>®</sup> Development Board HD200, to your computer is simple.

- Ensure proper jumper settings on the phyCORE<sup>®</sup> Development Board as shown in *Figure 1*.



*Figure 1: Default Jumper Settings of the phyCORE<sup>®</sup> Development Board HD200 with phyCORE<sup>®</sup>-LPC2294*

### Note:

If you do not have a Keil ULINK adapter, skip the following two bullet points.

- The ULINK JTAG adapter comes with various flat-band cables. In order to connect this device to the phyCORE<sup>®</sup>-LPC2294 module you need to install a flat-band cable with a 2.0 mm connector. If such a cable is not already installed on your ULINK open the enclosure and connect the correct 2.0 mm to 2.0 mm cable to the applicable header connector inside the ULINK. Make sure that pin #1 on the cable (black wire) matches pin #1 on the connector.

- Connect the other 2.0 mm cable connector onto pin header rows X701 on the phyCORE<sup>®</sup> module. Make sure that pin #1 on the ULINK cable (black wire) is correctly connected to pin #1 on JTAG connector X701 (located on the connector side of the PCB) of the phyCORE<sup>®</sup>-LPC2294 (refer to Figure 2).

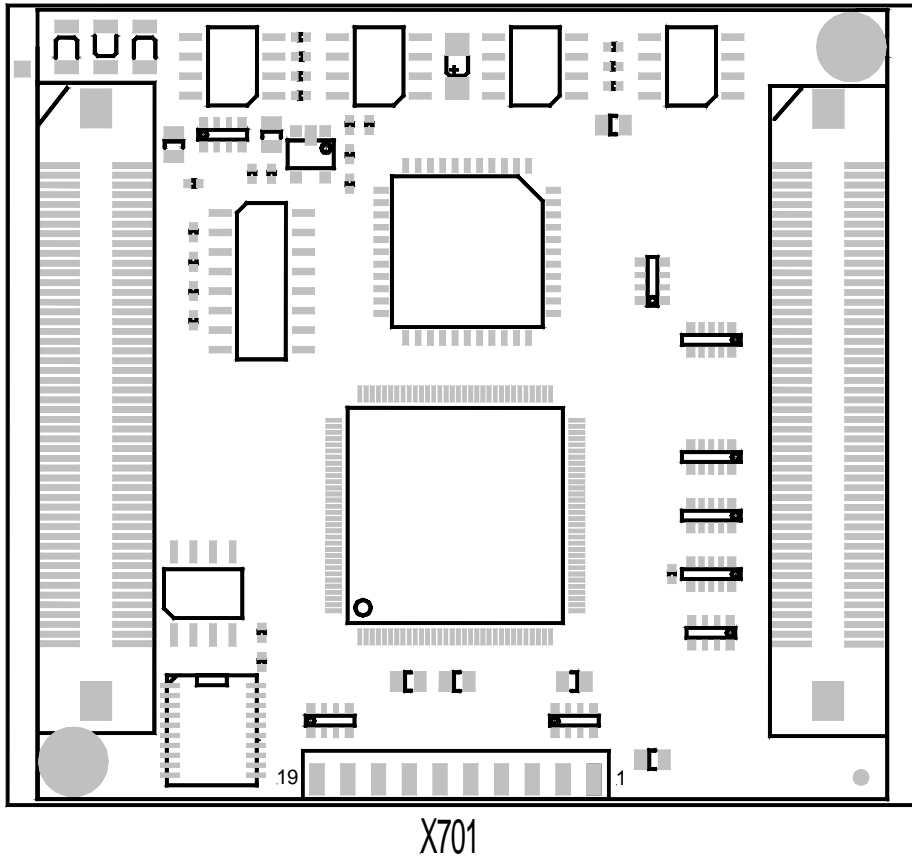


Figure 2: JTAG Connector X701 on the phyCORE<sup>®</sup>-LPC2294 (Bottom View)

- Mount the phyCORE<sup>®</sup> module, pins-down, onto the Development Board's receptacle footprint (X6) as shown in Figure 3 below. Ensure that pin 1 of the module, designated by the hash stencil mark, matches pin 1 of the receptacle on the Development Board.

Ensure that there is a solid connection between the module's pins and the Development Board receptacle. Also take precautions not to damage the connectors when the phyCORE<sup>®</sup> is removed from and inserted onto the Development Board.

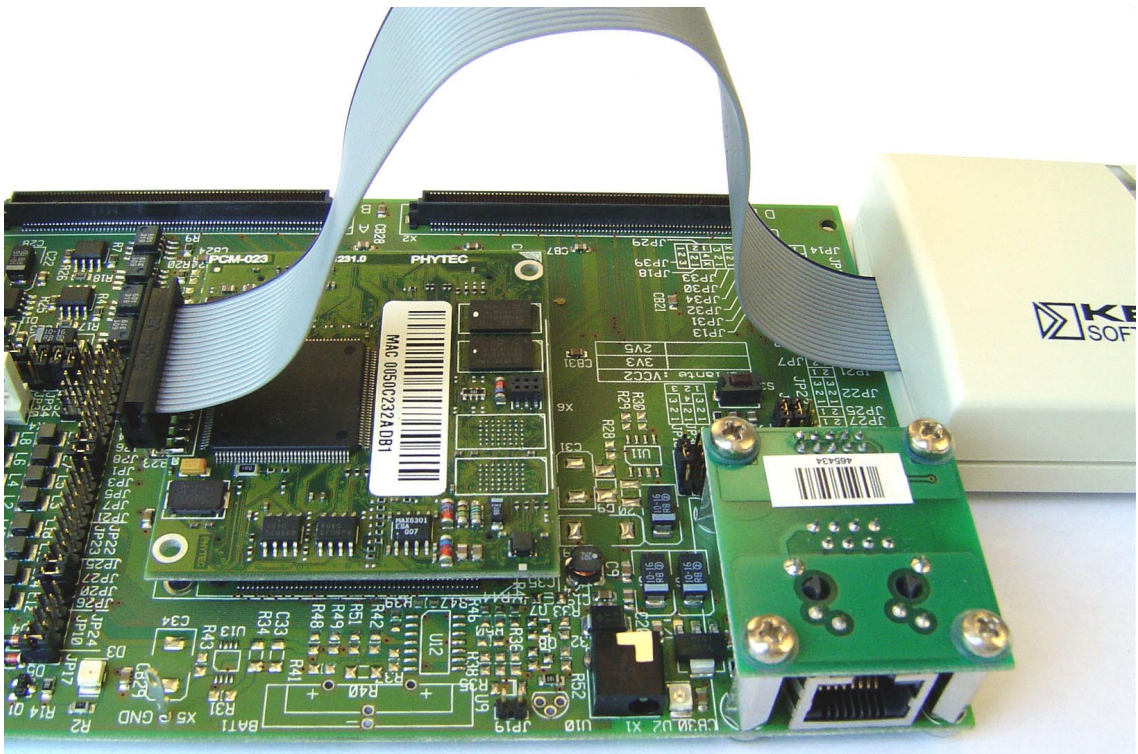


Figure 3: ULINK Connected to the phyCORE® Module

- Connect the USB end of the ULINK JTAG adapter to the USB port of your host-PC using the included USB cable.
- Connect the RS-232 interface of your computer to the DB-9 RS-232 interface on the phyCORE® Development Board HD200 (P1A = bottom) using the included serial cable.
- Using the included 5V DC power adapter, connect the power socket X1 on the phyCORE® module to a power supply (*refer to Figure 4 for the correct polarity*).

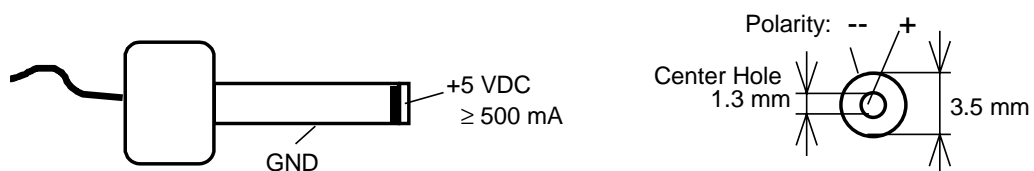


Figure 4: Power Connector

- The red power LED D2, located next to the power socket at X1, should light. This indicates that proper voltage is supplied to the phyCORE<sup>®</sup> module/Development Board combination (which is also referred to as "target hardware" within this document).
- The phyCORE<sup>®</sup>-LPC2294 should now be properly connected to a host PC via the Development Board and, if available, Keil ULINK. You are now ready to use the Keil ARM7/μVision3 tools to establish communication between the host-PC and target hardware.

### 2.3 Downloading Example Code with Philips LPC2000 Flash Utility

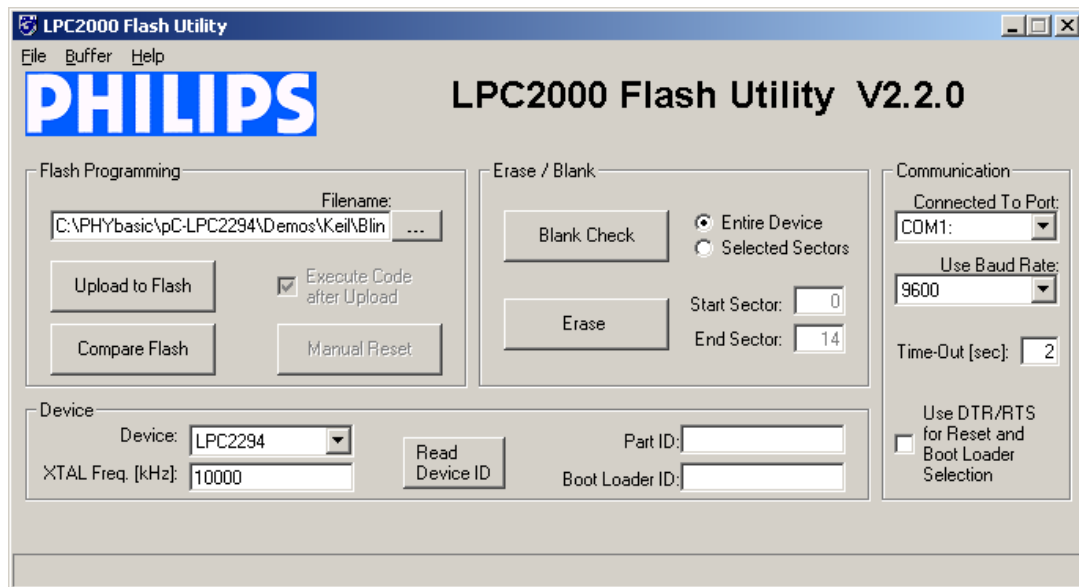
The Philips LPC2000 Flash Utility should have been installed during the initial setup procedure as described in *section 2.1*. If not, you can manually install it using the *setup.exe* file located in the folder *\Software\Philips\LPC2000 Flash Utility*.


Among other tasks, the LPC2000 Flash Utility program downloads user code in Intel *\*.hex* file format from a host-PC to on-chip Flash on an LPC229x device populating a PHYTEC SBC via an RS-232 connection.

**Note:**

Successful completion of this section requires that any user code residing in external Flash be erased. This is because, upon reset, the user code in external Flash will execute rather than user code in internal Flash. *For instruction on erasing the external Flash refer to the instructions for **Erasing Flash** in section 2.4.1.* Typically PHYTEC Rapid Development Kits are shipped with a blank external Flash.

- Start the LPC2000 Flash Utility by selecting it from within the *Programs / LCP2000 Flash Utility* program group.
- The LPC2000 Flash Utility GUI will now appear. Here you can Upload to Flash<sup>1</sup>, compare Flash, erase the Flash, check that the chip is erased, specify the connection properties, and read the device ID.



- For Flash programming, click on the browse button  and navigate to the *Blinky.hex* file to be downloaded within the default path  
*C:\PHYBasic\pC-LPC2294\Demos\Keil\Blinky\iFlash\Blinky.hex*.
- Be sure the correct *Device* is selected as **LPC2294** and *XTAL Freq* is set to **10000** kHz.
- Be sure that the correct *Communication Port* is set for your host-PC and select a **9,600** baud rate.
- Click on the *Upload to Flash* button.

<sup>1</sup> : **NOTE:** PHYTEC typically refers to the process of loading a machine readable file from a host-PC into a Flash device populating a PHYTEC Single Board Computer as "download". In this section we will use the term "Upload to Flash" as defined within the Philips LPC2000 Flash Utility for reasons of being consistent with their terminology. In all other sections of this manual we will use the term "Flash download".

- The following window will appear:



- Put the target hardware in Bootloader mode by simultaneously pressing the **Boot (S\_1)** and **Reset (S\_2)** buttons on the Development Board, first release the **Reset** button and, about 3 seconds later, the **Boot** button.
- Now click on *OK*.
- You will see the file upload progress in the lower bar of the LPC2000 Flash Utility window. When successful the bar will indicate *File Upload Successfully Completed*.
- Upon completion of file upload to on-chip Flash, press the Reset button S\_2 on the Development Board.
- Successful download of the blinky demo will result in the LED D3 on the Development Board blinking on and off at equal intervals.



## 2.4 Downloading Example Code with $\mu$ Vision3

The  $\mu$ Vision3 evaluation software development tool chain should have been installed during the install of the PHYTEC Rapid Development Kit Software from the Spectrum CD, as described in *section 2.1*.

You can also manually install the ARM7/ $\mu$ Vision3 tools by executing ***KARM220.EXE*** from within the `\Software\Keil\Arm` directory of your PHYTEC Spectrum CD. Follow the instructions displayed by the setup program for manual installation.

**Note:**

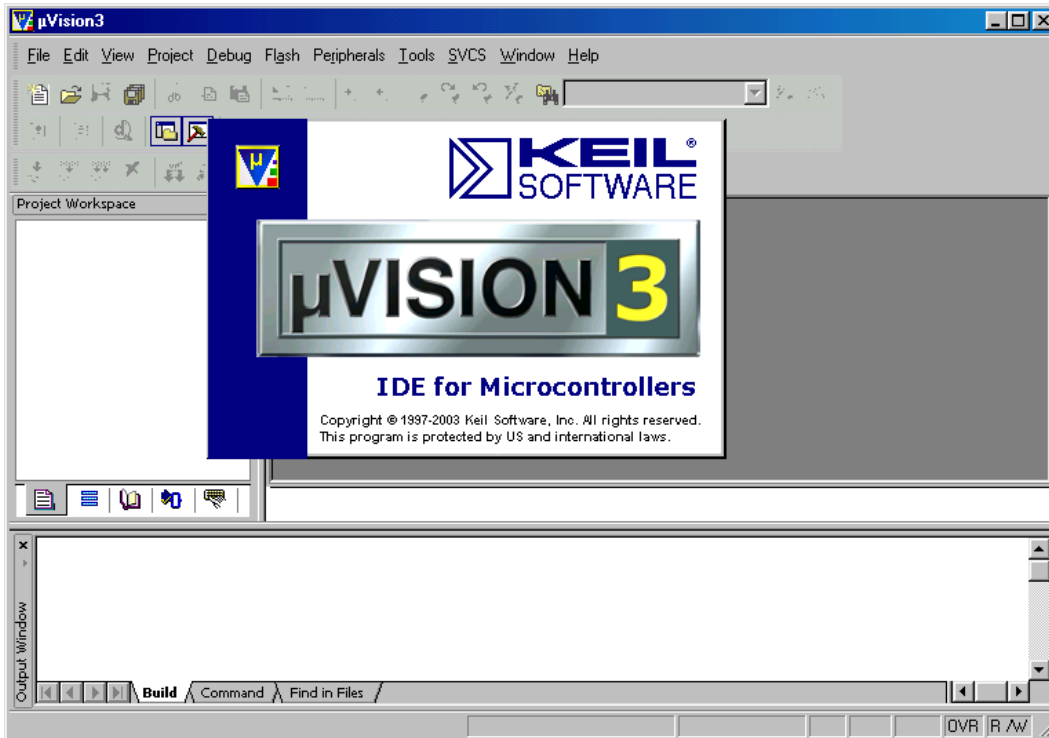
It is recommended to use the Keil tool chain provided on the accompanying Spectrum CD in order to complete this QuickStart Instruction successfully. Use of a different version could lead to possible version conflicts, resulting in functional problems<sup>1</sup>.

Start the tool chain by selecting Keil  $\mu$ Vision3 from within the programs group: `Start\Programs\Keil  $\mu$ Vision3` or by double-clicking on the Keil  $\mu$ Vision3 icon on your desktop.

---

<sup>1</sup>: **Note:** If you have a full version of the Keil ARM7 tools already installed on your host-PC we recommend installing this evaluation version on a different desktop or laptop computer. If this is not possible we recommend to use the same version of the Keil DK-ARM tools that we use in this QuickStart manual. PHYTEC can not guarantee successful completion of these QuickStart instructions if a different version of the Keil DK-ARM tools is used.

After you start  $\mu$ Vision3, the window shown below appears. From this window you can create projects, edit files, configure tools, assemble, link and start the debugger. Close all projects that might be open by selecting *Project / Close Project*.

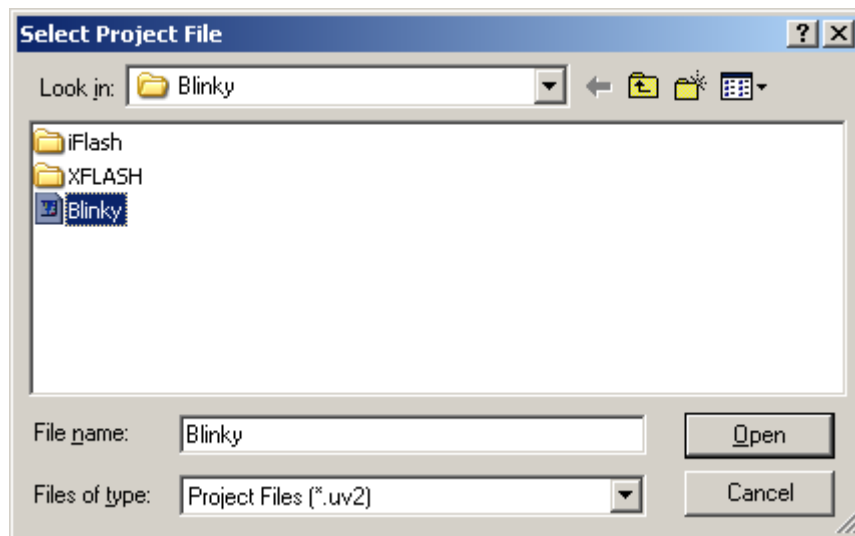


### 2.4.1 "Blinky"

The "Blinky" example downloads a program to external Flash that, when executed, manipulates the LED D3 on the phyCORE<sup>®</sup> Development Board HD200.

- Open the *Blinky* project from the  $\mu$ Vision3 menu *Project / Open Project*.
- Browse to *C:\PHYBasic\pC- LPC2294\Demos\Keil\Blinky*.

- Select the *Blinky* project.




- Click *Open*.
- In the **Select Target** pull-down menu be sure that the *phyCORE-LPC2294 XFLASH* target is selected.

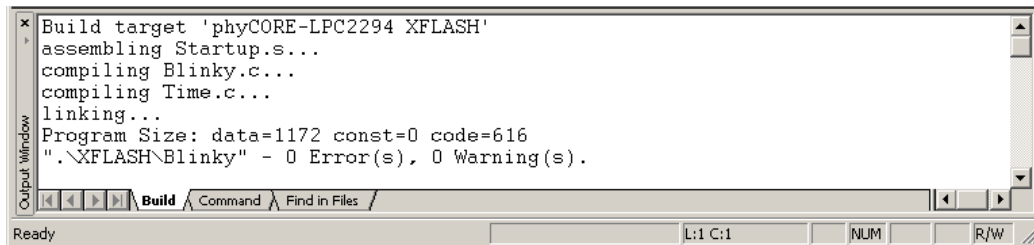
```

01 /* ***** */
02 /* This file is part of the uVision/ARM development tools */
03 /* Copyright KEIL ELEKTRONIK GmbH 2002-2004 */
04 /* ***** */
05 /* */
06 /* Blinky.c: LED Flasher */
07 /* */
08 /* ***** */
09
10 #include <LPC22XX.H> /* LPC22XX Peripheral Registers */
11
12
13 extern long volatile timeval;
14
15 void wait (void)
16 {
17     unsigned long i; /* wait function */
18
19     i = timeval;
20     while ((timeval - i) != 10); /* wait 100ms */
21 }
22
23
24 extern void init_timer (void);
25
26 int main (void)
27 {
28     init_timer(); /* Initialize Timer */
29
30     IOIPR0 = 0x00000100; /* P0.8 defined as Output */
31
32     while (1) /* Loop forever */
33     {
34         IOCLR0 = 0x00000100; /* Turn LED On (P0.8 = 0) */
35         wait(); /* Wait */
36         IOSET0 = 0x00000100; /* Turn LED Off (P0.8 = 1) */
37         wait(); /* Wait */
38     }
39 }
40

```

## Build the Project

- Build the target by either selecting the *Build Target* icon  on the build toolbar or in the main menu bar select *Project / Build target*.
- If any source file of the project contains any errors, they will be shown in the **Output Window - Build** tab. Use the editor to correct the error(s) in the source code, save the file and repeat the build.
- If there are no errors, the code is ready to be downloaded into the external Flash memory.




```

Build target 'phyCORE-LPC2294 XFLASH'
assembling Startup.s...
compiling Blinky.c...
compiling Time.c...
linking...
Program Size: data=1172 const=0 code=616
".\XFLASH\Blinky" - 0 Error(s), 0 Warning(s).

```

## Download to Flash

- Download the code into Flash memory by either selecting the *Download to Flash Memory* icon  on the build toolbar or in the main menu bar select *Flash / Download*.
- The individual steps of the Flash download procedure can be viewed at the bottom of the  $\mu$ Vision3 **Output Window - Build** tab.
- Wait until the programming is complete. This is indicated by the "**Verify OK**" message. The download utility will perform a reset and the code will execute without further user interaction.

Successful execution of the program will flash the LED D3 with equal on and off duration.

## Erase Flash

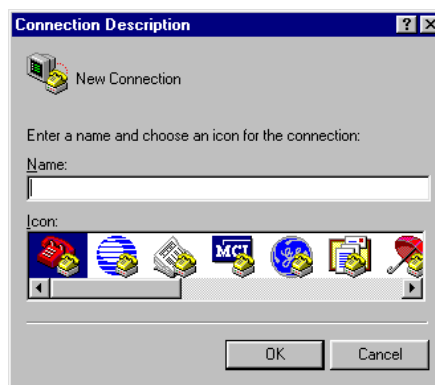
- To erase Flash, select **Flash/Erase** from the uVision toolbar. This requires that a project with an XFLASH target is open.

## 2.4.2 "Hello"

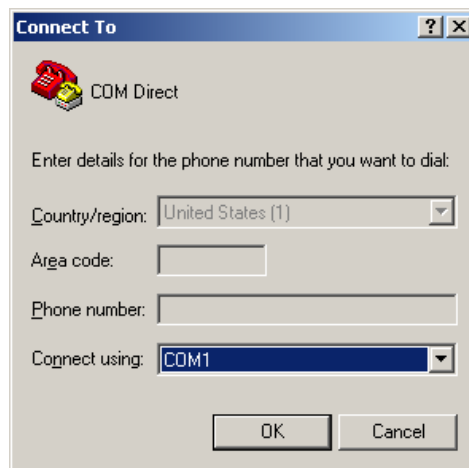
The "Hello" example downloads a program to the external Flash that, when executed, sends a character string from the target hardware back to the host-PC. The character string can be viewed with a terminal emulation program. This example program provides a review of the Flash download procedure using the Keil ULINK and  $\mu$ Vision3.

Monitoring the execution of the Hello demo requires use of a terminal program, such as the HyperTerminal program included within Windows.

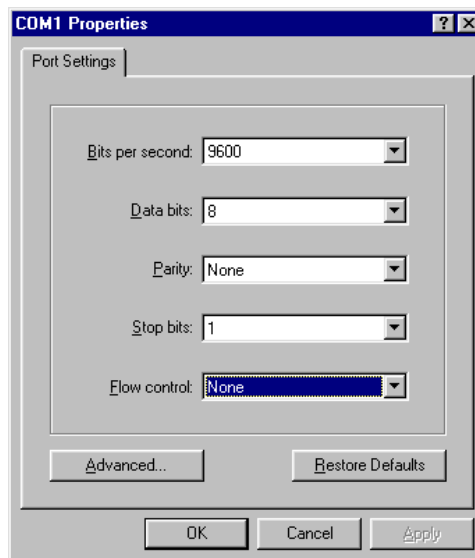
- Start the HyperTerminal program within the *Programs /Accessories/Communications* bar.
- The Connection Description window will now appear. Enter "COM Direct" in the *Name* text field.



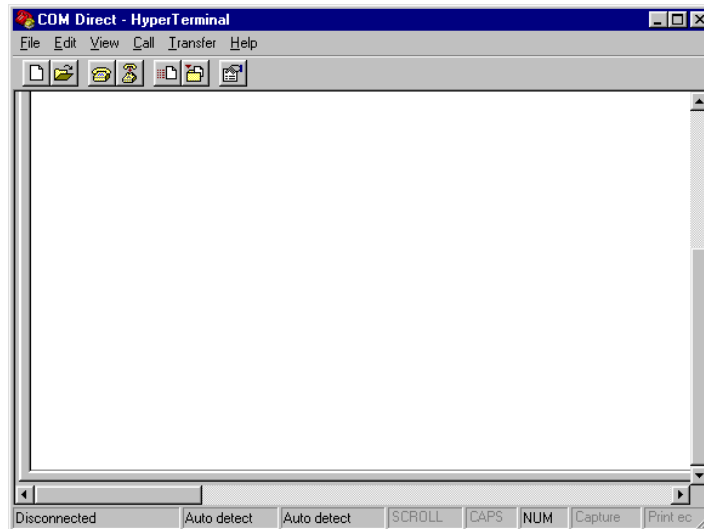
- Next click on *OK*. This creates a new HyperTerminal session named "COM Direct" and advances you to the next HyperTerminal window. Specify *COM1* under the *Connect Using* pull-down menu (be sure to indicate the correct COM setting for your system).



- Click *OK* to advance to the next window (*COM1 Properties*).
- Then set the following COM parameters: Bits per second = 9,600; Data bits = 8; Parity = None; Stop Bits = 1; Flow Control = None.

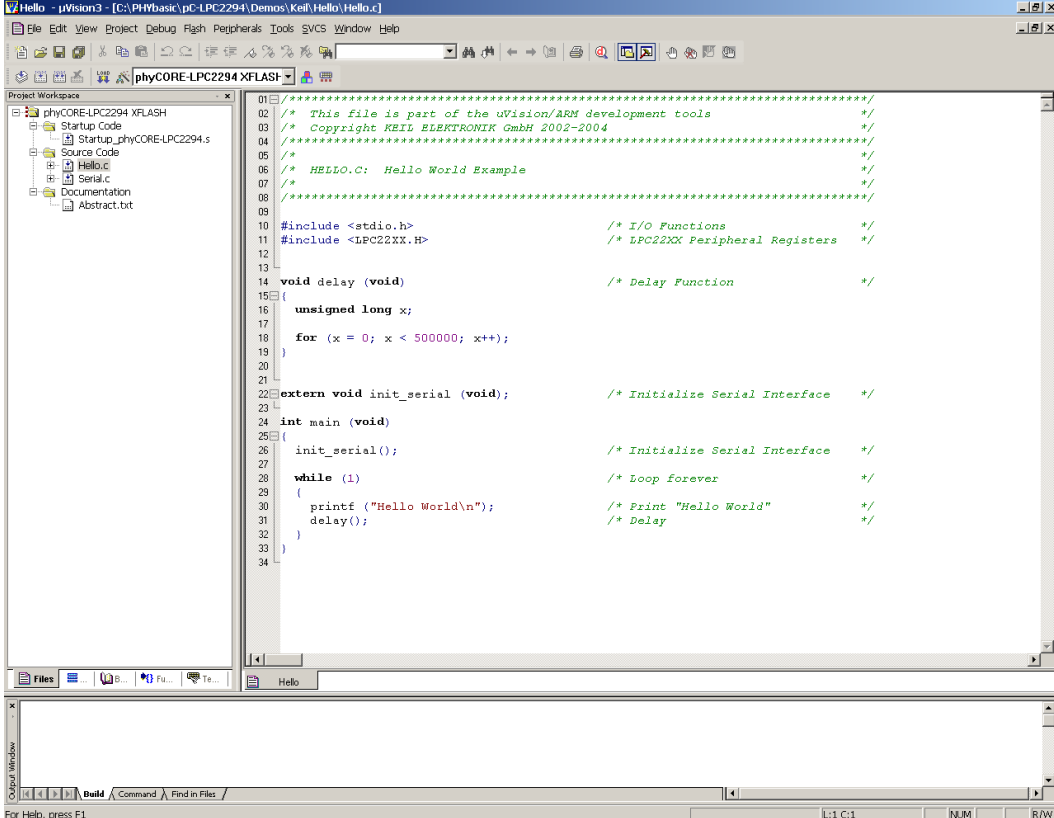


- Selecting *OK* advances you to the *COM Direct–HyperTerminal* monitoring window. Notice the connection status report in the lower left corner of the window.



- Ensure that the target hardware is properly connected to the host-PC via the ULINK and serial cable as well as a power supply.

- Open the **Hello** project from the  $\mu$ Vision3 menu *Project / Open Project*.
- Browse to the correct drive and path for the phyCORE<sup>®</sup>-LPC2294 demo folder (default location **C:\PHYBasic\pC-LPC2294\Demos\Keil>Hello\**) and click *Open*.




```
01 /******  
02 /* This file is part of the uVision/ARM development tools  
03 /* Copyright KEIL ELEKTRONIK GmbH 2002-2004  
04 /******  
05 /*  
06 /* HELLO.C: Hello World Example  
07 /*  
08 /******  
09  
10 #include <stdio.h> /* I/O Functions */  
11 #include <LPC22XX.H> /* LPC22XX Peripheral Registers */  
12  
13  
14 void delay (void) /* Delay Function */  
15 {  
16     unsigned long x;  
17  
18     for (x = 0; x < 500000; x++);  
19 }  
20  
21  
22 extern void init_serial (void); /* Initialize Serial Interface */  
23  
24 int main (void)  
25 {  
26     init_serial(); /* Initialize Serial Interface */  
27  
28     while (1) /* Loop forever */  
29     {  
30         printf ("Hello World\n"); /* Print "Hello World" */  
31         delay(); /* Delay */  
32     }  
33 }  
34
```


- In the **Select Target** pull down menu be sure that the **phyCORE-LPC2294 XFLASH** target is selected.




## Build the Project

- Click on the *Build Target* icon  on the build toolbar to build the target.
- If any source file of the project contains any errors, they will be shown in the **Output Window - Build** tab. Use the editor to correct the error(s) in the source code, save the file and repeat the build.
- If there are no errors, the code is ready to be downloaded into the Flash memory.

## Download to Flash

- Click on the *Download to Flash Memory* icon  in the build toolbar to download the code into Flash memory.
- Wait until the programming is complete. This is indicated by the "**Verify OK**" message. The download utility will perform a reset and the code will execute without further user interaction.
- Successful program execution will send the character string "*Hello World*" from the target hardware to the HyperTerminal window.
- If no output appears in the HyperTerminal window check the power supply, the COM parameters and the RS-232 connection.

The code within the demo application *Hello* initializes the serial port of your phyCORE<sup>®</sup>-LPC2294 to 9600 baud. The initialization values are based on the assumption that the microcontroller runs at a 60 MHz internal clock frequency. If your phyCORE<sup>®</sup>-LPC2294 is equipped with a different speed oscillator, the demo application might transmit using another baud rate. This may lead to incoherent characters appearing in the HyperTerminal window following execution of code.

- Click the disconnect icon  in HyperTerminal toolbar and exit HyperTerminal.



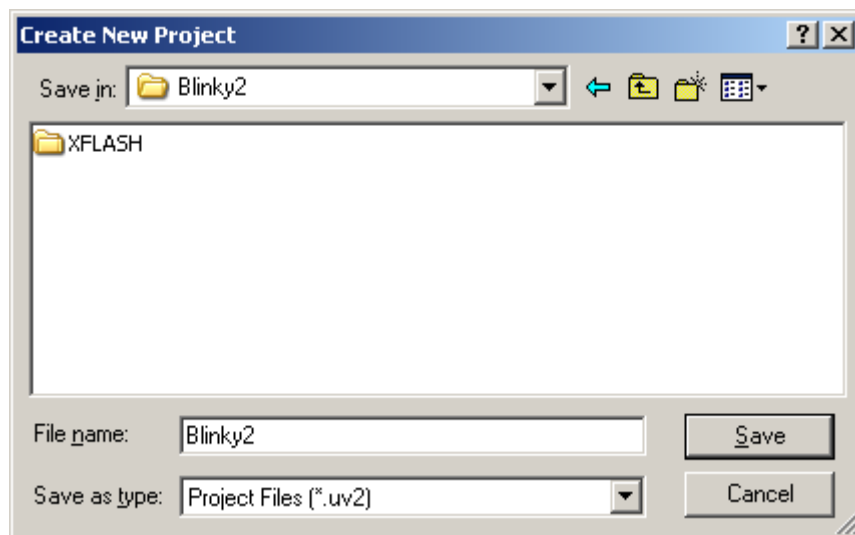
### 3 Getting More Involved

What you will learn with this example:

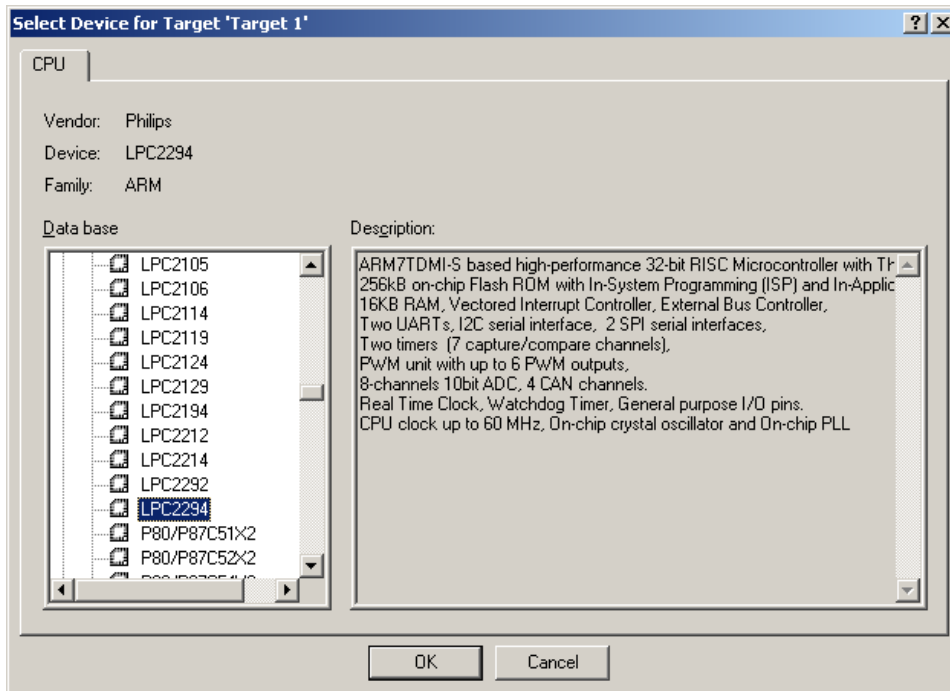
- how to configure the  $\mu$ Vision3 IDE (Integrated Development Environment)
- how to modify the source code from our examples, create a new project and build and download a machine-readable output file to the target hardware

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

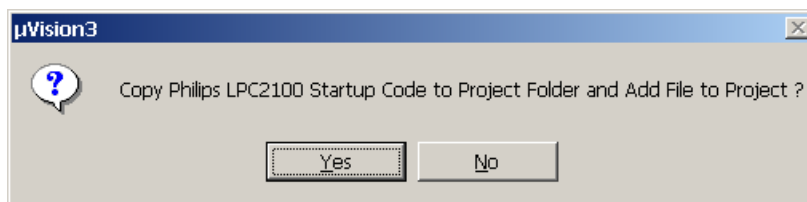
- To create a new project file select from the  $\mu$ Vision3 menu **Project/New Project....** This opens a standard Windows dialog that asks you for the new project file name.
- Change to the project directory created by the installation procedure (default location **C:\PHYBasic\pC-LPC2294\Demos\Keil\Blinky2**).
- In the text field 'File name', enter the file name of the project as **Blinky2.uv2** and click on **Save**.



- The **Select Device for Target 'Target 1'** window will automatically appear. Select **Philips** as manufacturer for the CPU. The phyCORE<sup>®</sup>-LPC2294 is populated with an LPC2294 CPU. Choose the controller type from the list as shown below. This selection sets necessary tool options for the LPC2294 device and simplifies in this way the project configuration.



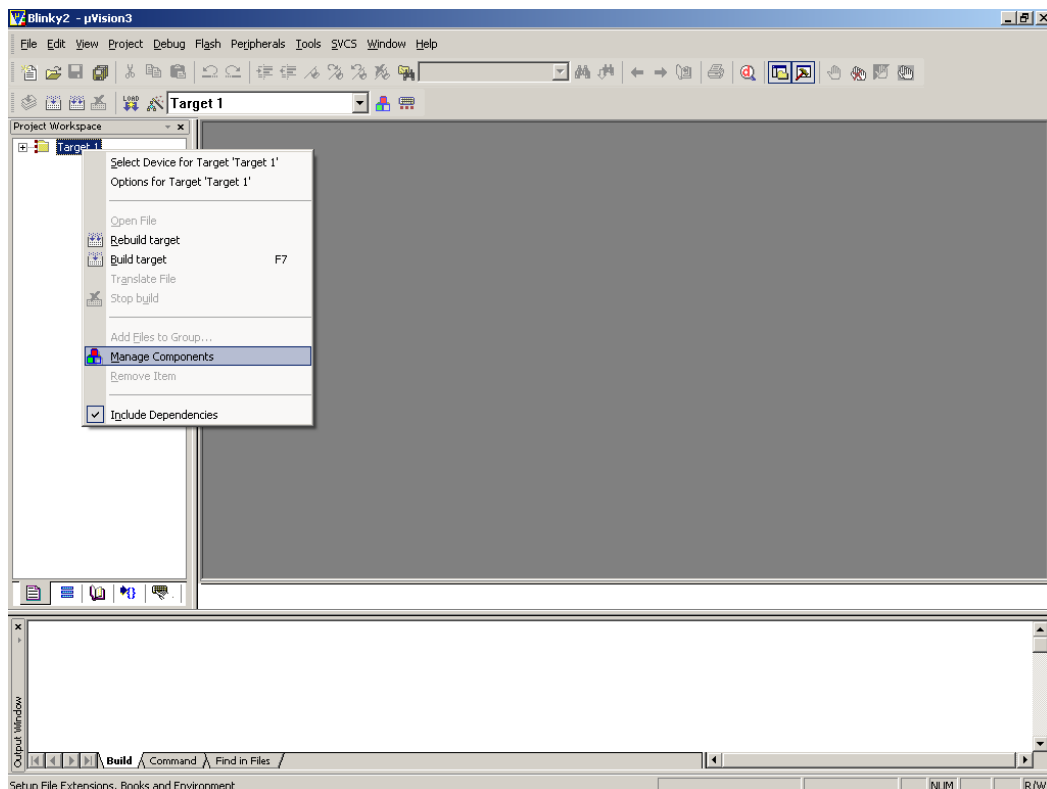
- Click on *OK* to save the settings.
- The uVision3 dialog box "*Copy Philips LPC2100 Startup Code to Project Folder and Add File to Project?*" will appear.



- Click on *No* to not include this default startup code! The default startup code provided by Keil does not match the phyCORE-LPC2294 hardware properties. The correct startup file called *startup\_phyCORE-LPC2294.s* will be added to the project later.

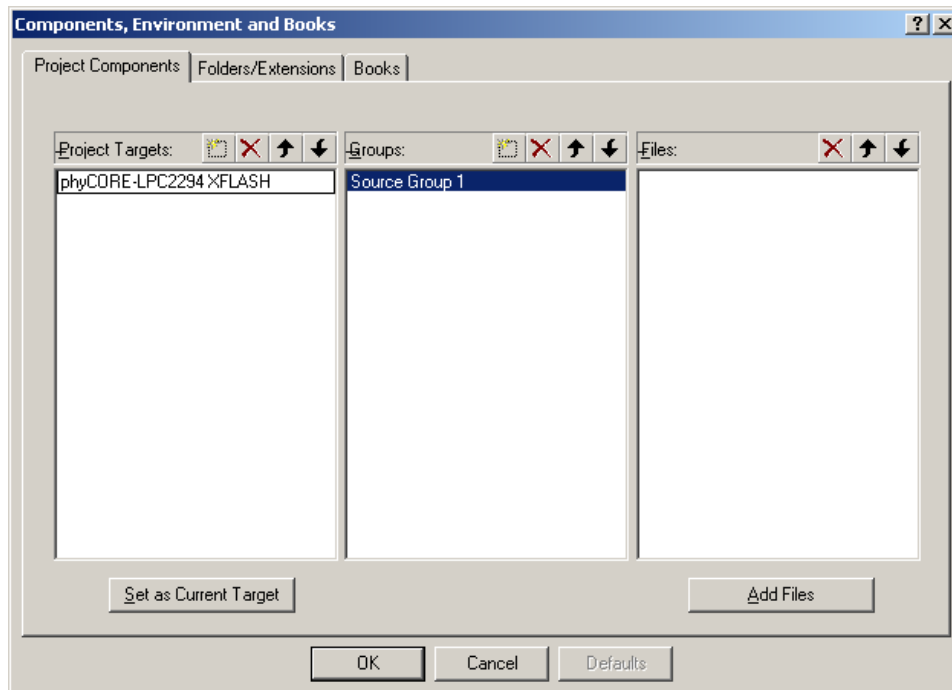
### 3.1.1 Configure the Project Components

- To configure the target click on the  icon in the build toolbar or right-click on the target, 'Target 1' in the **Project Workspace** window and select *Manage Components*.

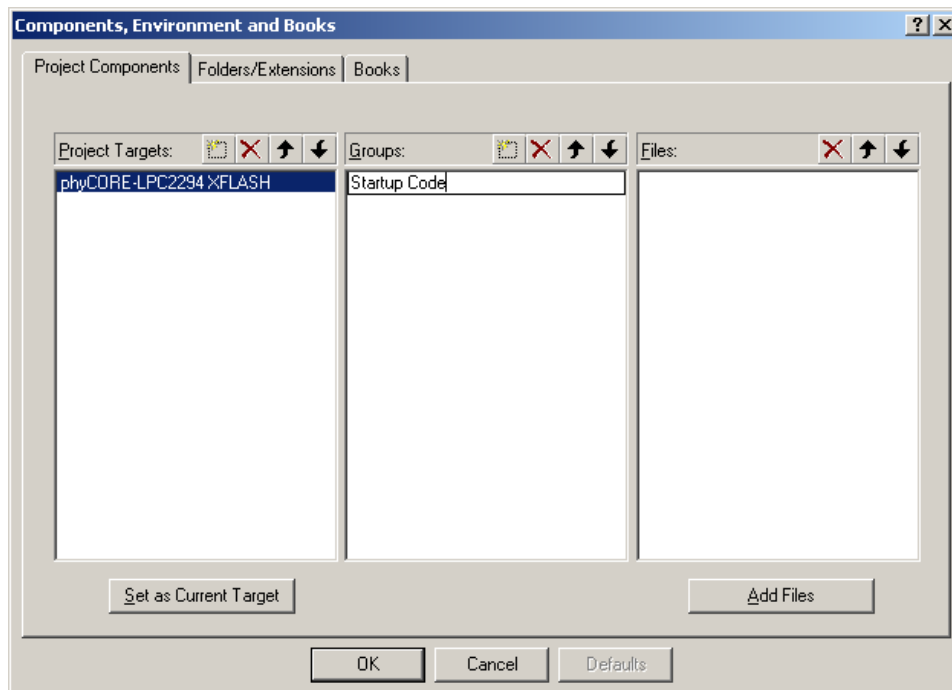



- The **Components, Environments, and Books** window will appear.

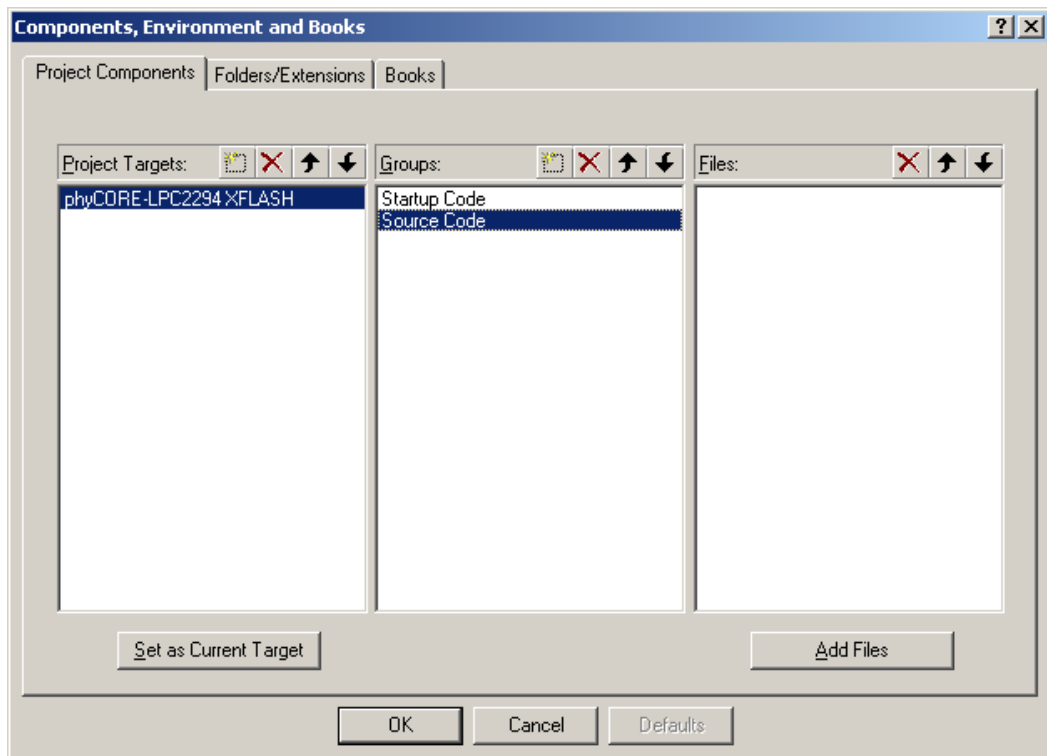
- In the **Project Components** tab, double-click on *Target 1*. Change the name of the target to *phyCORE-LPC2294 XFLASH*.



- Double-click on *Source Group 1*, in the **Groups** window, and change the name of the group to *Startup Code*.

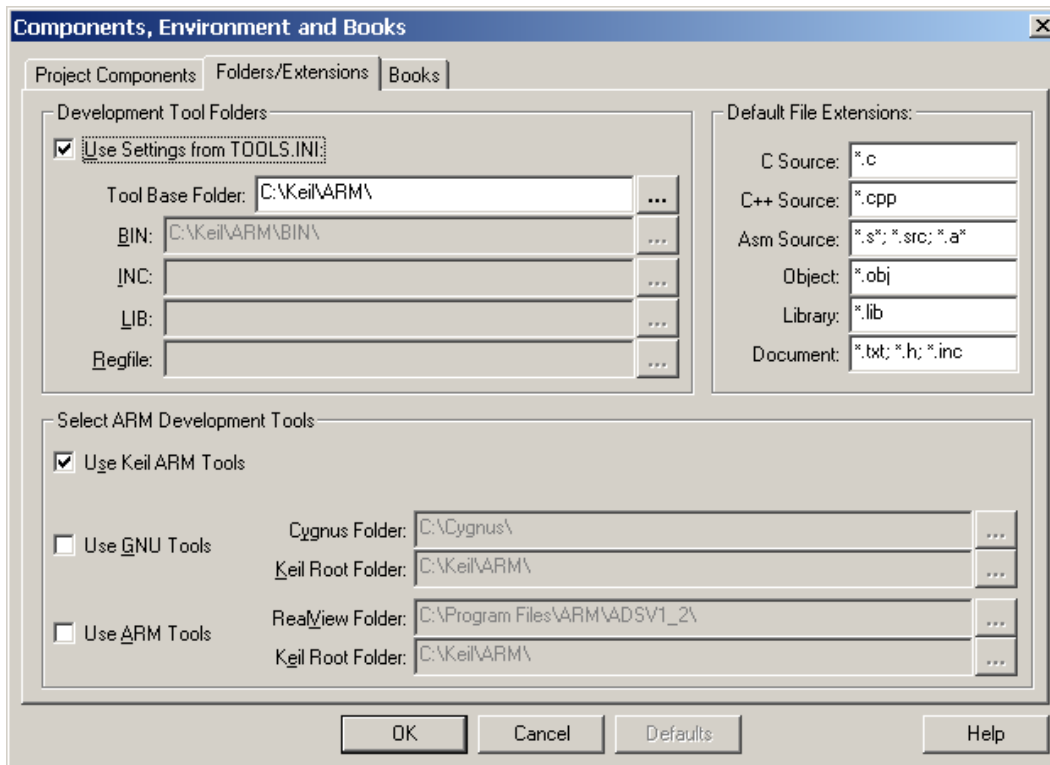


- Add another group by selecting the *New(Insert)* icon  in the **Groups** window. Name the new group *Source Code*.



- Select the **Folders/Extensions** tab.

- Be sure that *Use Settings from TOOLS.INI* is checked and the path for the *Tool Based Folder* is *C:\Keil\ARM\*
- Be sure that *Use Keil ARM Tools ARM Development Tools* is selected.

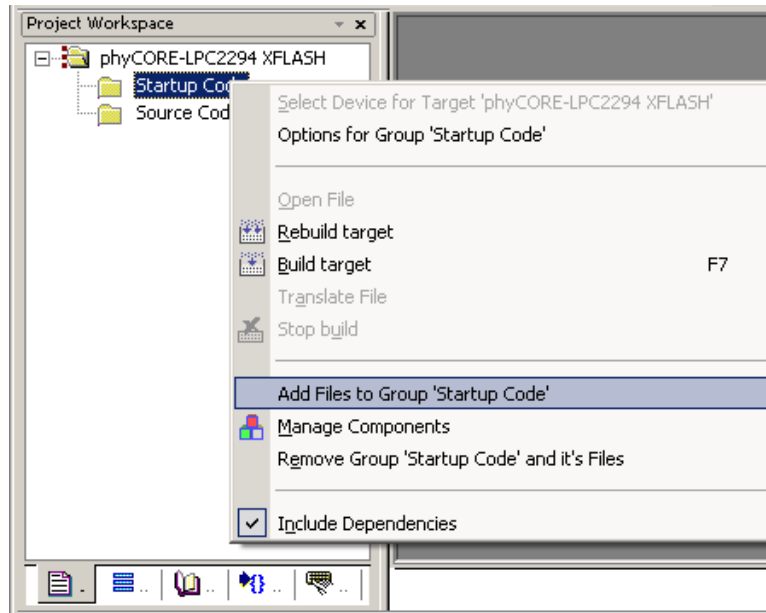


- Click *OK*. This brings you back to the **Project Workspace** window.
- You are now ready to add source files to the project. Make sure the view in the **Project Workspace** window is expanded to see both file groups.

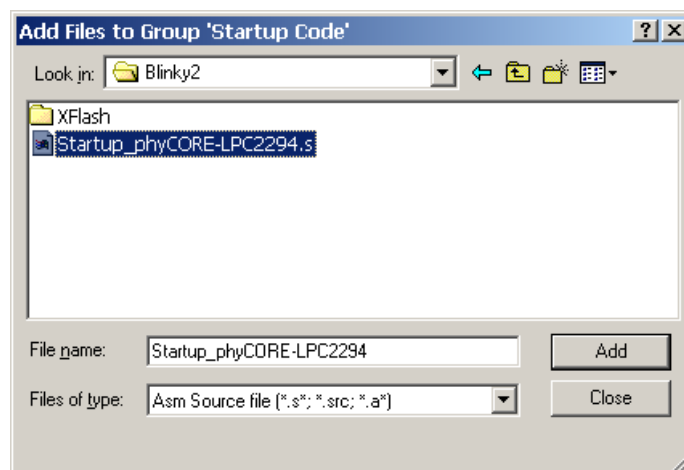


### 3.1.2 Adding Source Files to the Project

- In the **Project Workspace** window - **Files** tab right-click on *Startup Code* and select *Add Files to Group 'Startup Code'*.

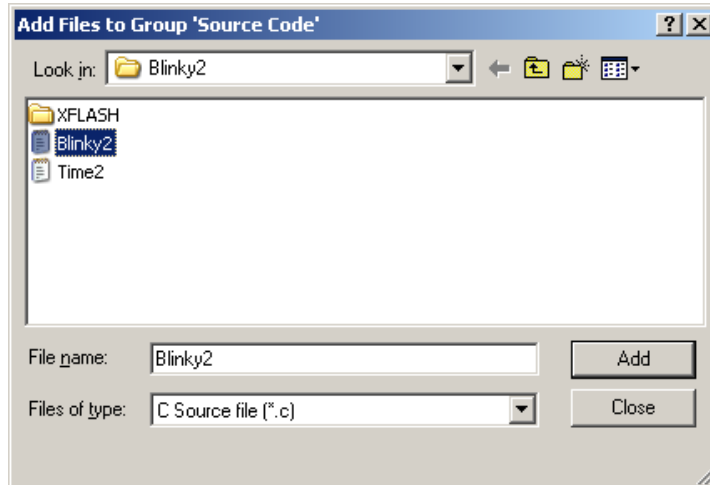


- In the *File of type* pull-down menu, select: "*Asm Source file (\*.s\*; \*.src\*; .a\*)*". Browse to *C:\PHYBasic\pC-LPC2294\Demos\Keil\Blinky2* and select *Startup\_phyCORE-LPC2294.s*.

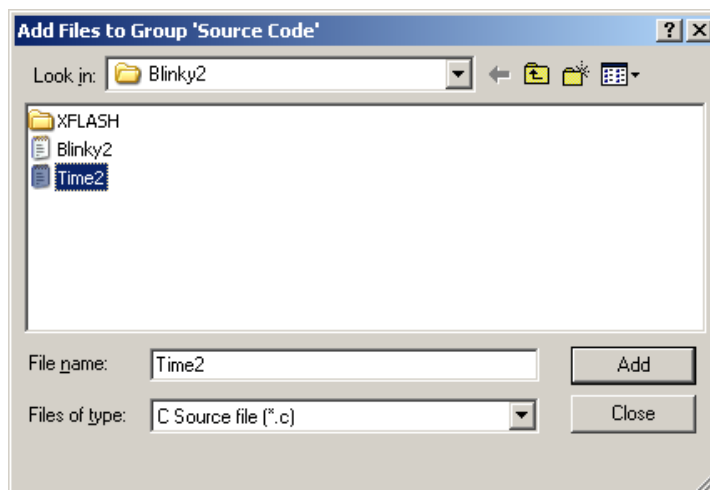


- Click on the *Add* button and then click *Close*.

- In the **Project Workspace** window right-click on the *Source Code* group.
- Select *Add Files to Group 'Source Code'*.
- Browse to *C:\PHYBasic\pC-LPC2294\Demos\Keil\Blinky2* and select *Blinky2.c*.

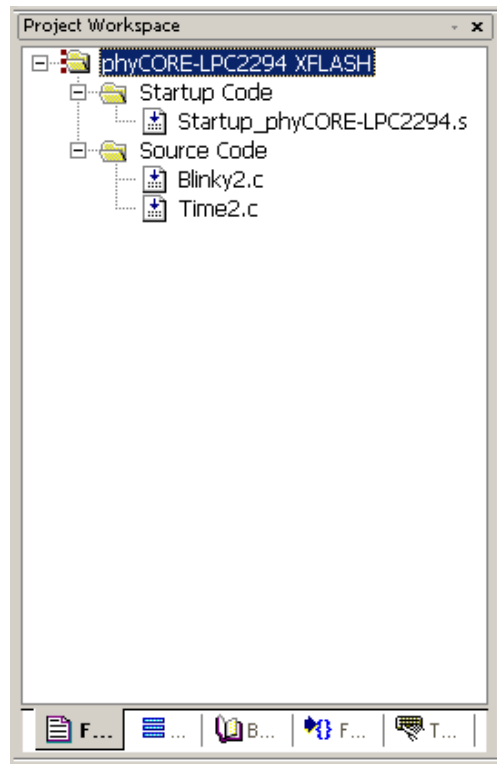


- Click on the *Add* button.
- Select *Time2.c*.



- Click on the *Add* button and then click *Close*.

- The **Project Workspace** window should appear as follows:



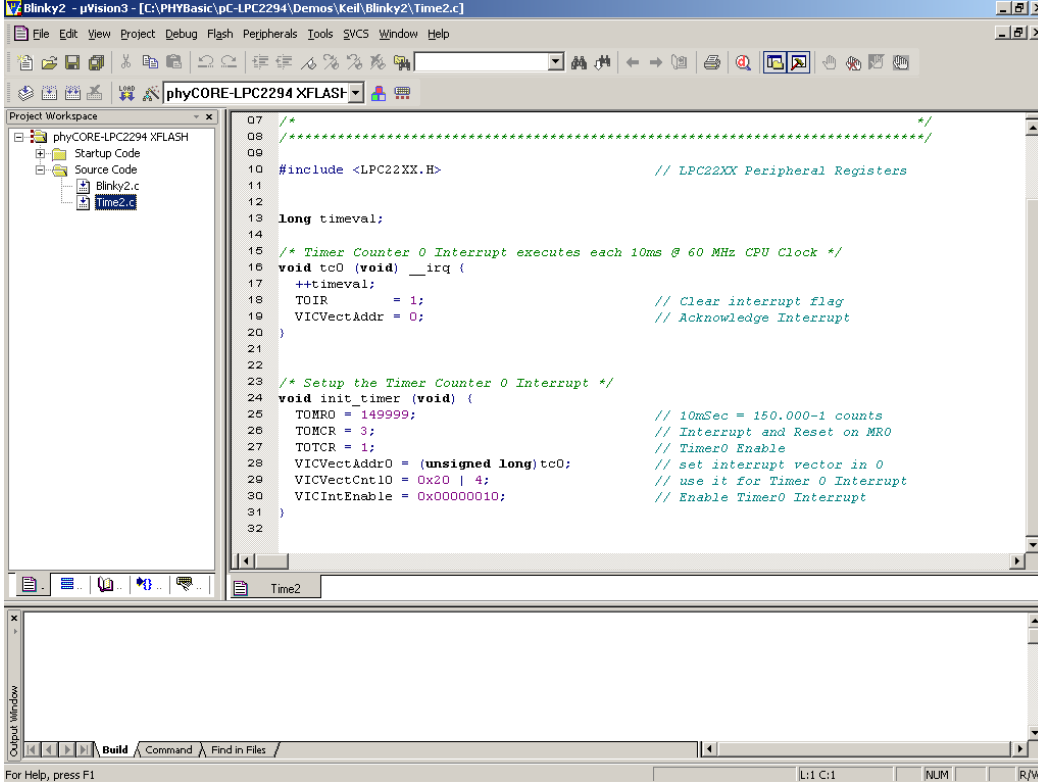
At this point you have created a project called *Blinky2.uv2* and added an existing C source file called *Blinky2.c* and *Time2.c* and an existing assembly source file called *Startup\_phyCORE-LPC2294.s*. The next step is to modify the C source file before building your project. This includes compiling, linking, locating and creating the machine-readable file.

**NOTE:**

Always use the *Startup\_phyCORE-LPC2294.s* file provided by PHYTEC in your application project. This startup file contains the correct controller setting for access to external memory and other on-board components. Using other startup code, e.g. the default Keil startup code that is offered when creating a new project will lead to functional problems and may cause your application code to not execute as desired.

## 3.2 Modifying the Source Code

- Double-click on *Time2.c* in the **Project Workspace** window to open the file in the source code editor.




```

07  /*
08  /**
09  //*****
10  #include <LPC22XX.H>                // LPC22XX Peripheral Registers
11
12
13  long timeval;
14
15  /* Timer Counter 0 Interrupt executes each 10ms @ 60 MHz CPU Clock */
16  void tc0 (void) __irq (
17  ++timeval;
18  TOIR      = 1;                    // Clear interrupt flag
19  VICVectAddr = 0;                // Acknowledge Interrupt
20  }
21
22
23  /* Setup the Timer Counter 0 Interrupt */
24  void init_timer (void) (
25  TOMR0 = 149999;                 // 10mSec = 150.000-1 counts
26  TOMCR = 3;                     // Interrupt and Reset on MRO
27  TOTCR = 1;                     // Timer0 Enable
28  VICVectAddr0 = (unsigned long)tc0; // set interrupt vector in 0
29  VICVectCnt10 = 0x20 | 4;       // use it for Timer 0 Interrupt
30  VICIntEnable = 0x00000010;    // Enable Timer0 Interrupt
31  )
32

```

- Locate the following code section and modify the delay for LED flashes from the original **149999** to **549999**.


```
TOMR0 = 149999;                // 10mSec = 150.000-1 counts
```

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

### 3.3 Setting Options for Flash Target

Keil includes a Make utility that can control compiling and linking source files in several programming languages. Before building your project you must configure the target options. Most of the options are set when specifying the target device for the project.

Enter the changes as indicated below and leave all other options set to their default values.  $\mu$ Vision3 allows you to set various options with mouse clicks and these are all saved in your project *\*.opt* file.

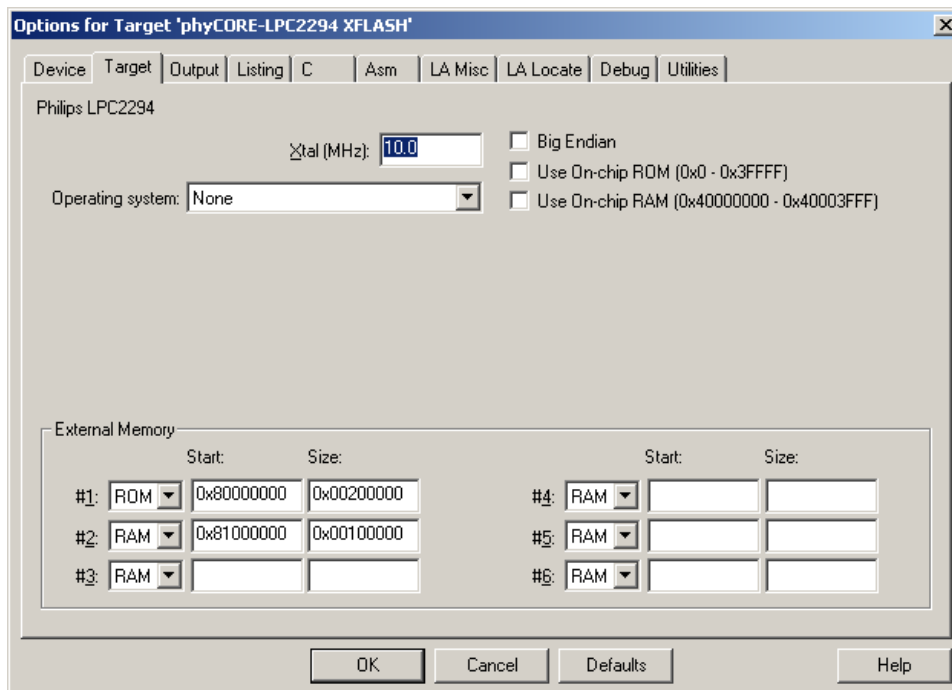
- Configure options for target by selecting the *Options for Target* icon  on the build toolbar or right-click on the *phyCORE-LPC2294 XFLASH* target in the **Project Workspace** window and select *Options for Target 'phyCORE-LPC2294 XFLASH'*.

## Configure the Target Options

- In the **Target** tab be sure that the *Xtal* is set to **10 MHz** and *Use On-chip ROM(0x0-0x3FFFF)* as well as *Use On-chip RAM(0x40000000-0x40003FFF)* are not checked. Set the *External Memory #1* to **ROM** and #2 to **RAM** and set the *Start* and *Size* as follows:

ROM:   **0x8000 0000** (Start)       **0x0020 0000** (Size)

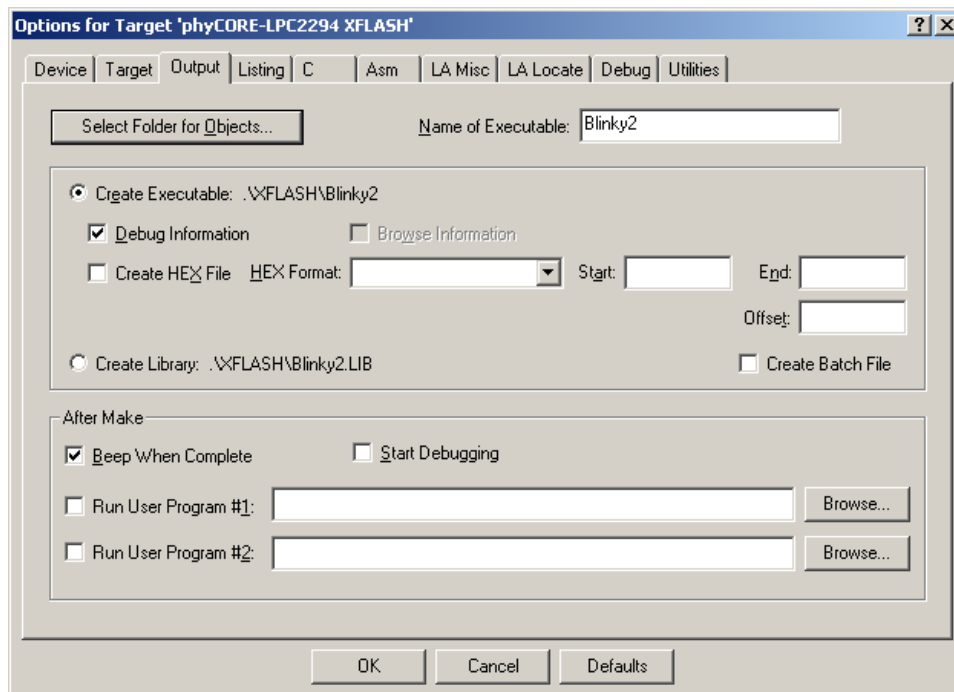
RAM:   **0x8100 0000** (Start)       **0x0010 0000** (Size)



It is necessary to configure the External Memory *Start* and *Size* settings so that the user code and data does not exceed the physical size of the memory. The phyCORE®-LPC2294 standard version features 2 MByte of external Flash and 1 MByte of external SRAM.

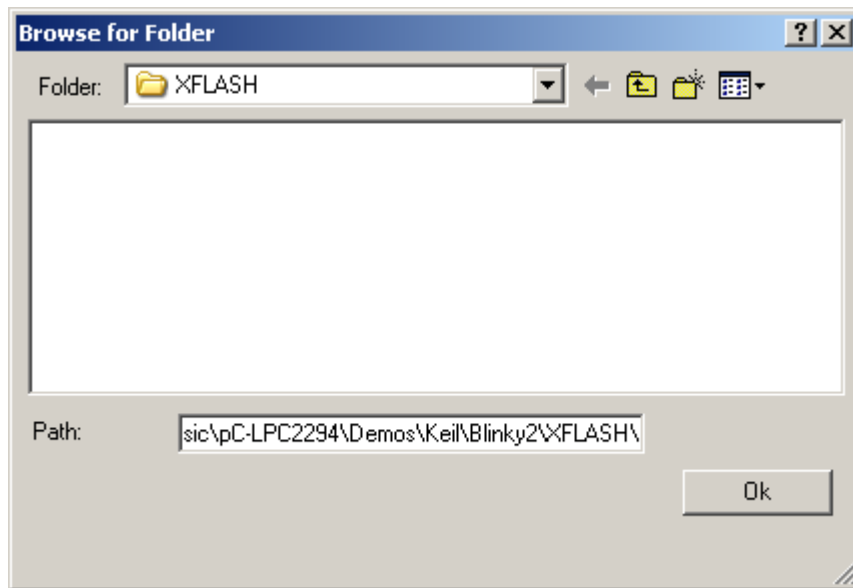
## Configure the Output Options

- In the **Output** tab be sure that *Create Executable*, *Debug Information* and *Beep When Complete* options are selected.



- Click on the *Select Folder for Objects* button.

- Browse to the folder:  
***C:\PHYBasic\pC-LPC2294\Demos\Keil\Blinky2\XFLASH.***

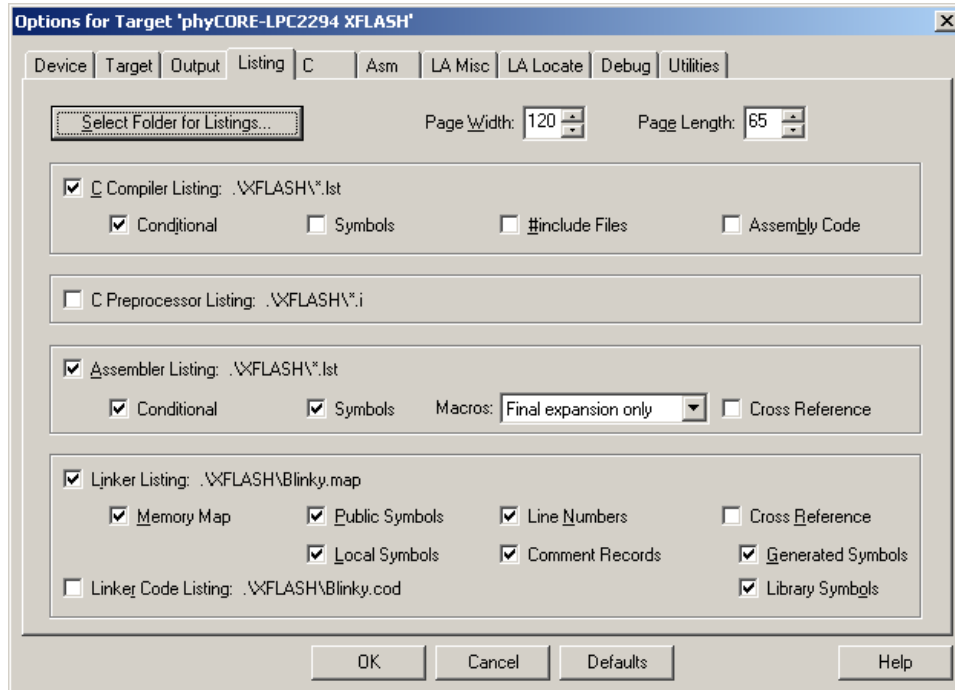


- Click *Ok*. This takes you back to the *Options for Target* window.

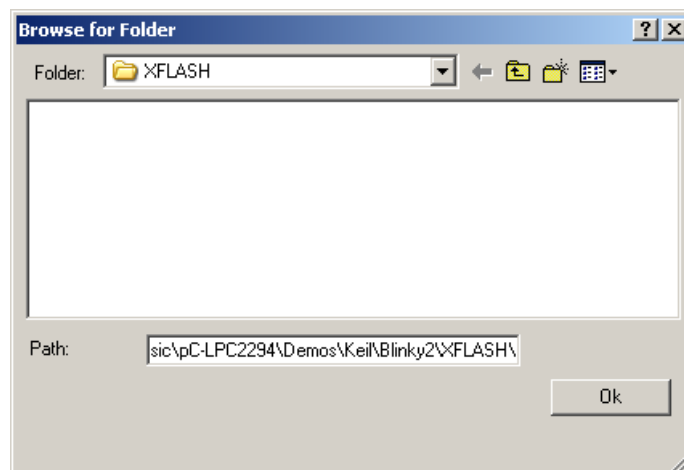


## Configure the Listing Options

- In the **Listing** tab, be sure that the default settings are set as follows.



- Click on the *Select Folder for Listings* button.
- Browse to the folder:  
***C:\PHYBasic\pC- LPC2294\Demos\Keil\Blinky2\XFLASH***



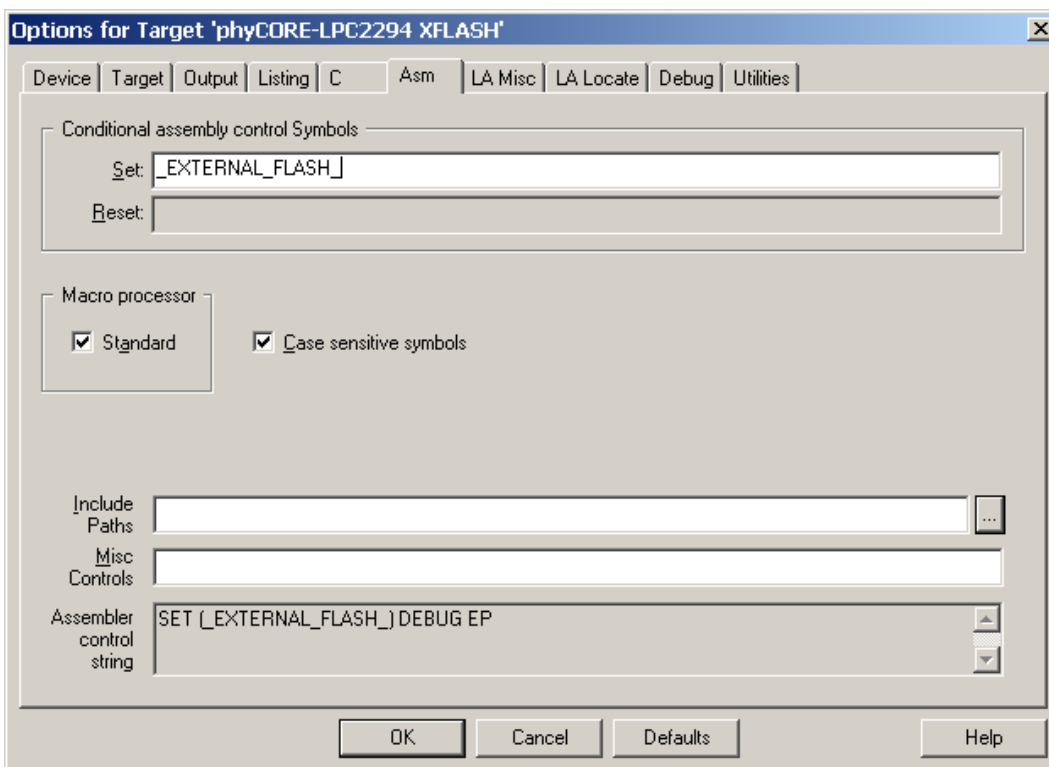
- Click *OK*. This takes you back to the *Options for Target* window.

## Configure the Asm Options

- Change to the **Asm** tab. In the *Conditional assembly control Symbols*, *Set* field, type: **`_EXTERNAL_FLASH_`**. This will set the LPC2294 MEMMAP register to *User External Memory Mode*, where the interrupt vectors are re-mapped to external memory.

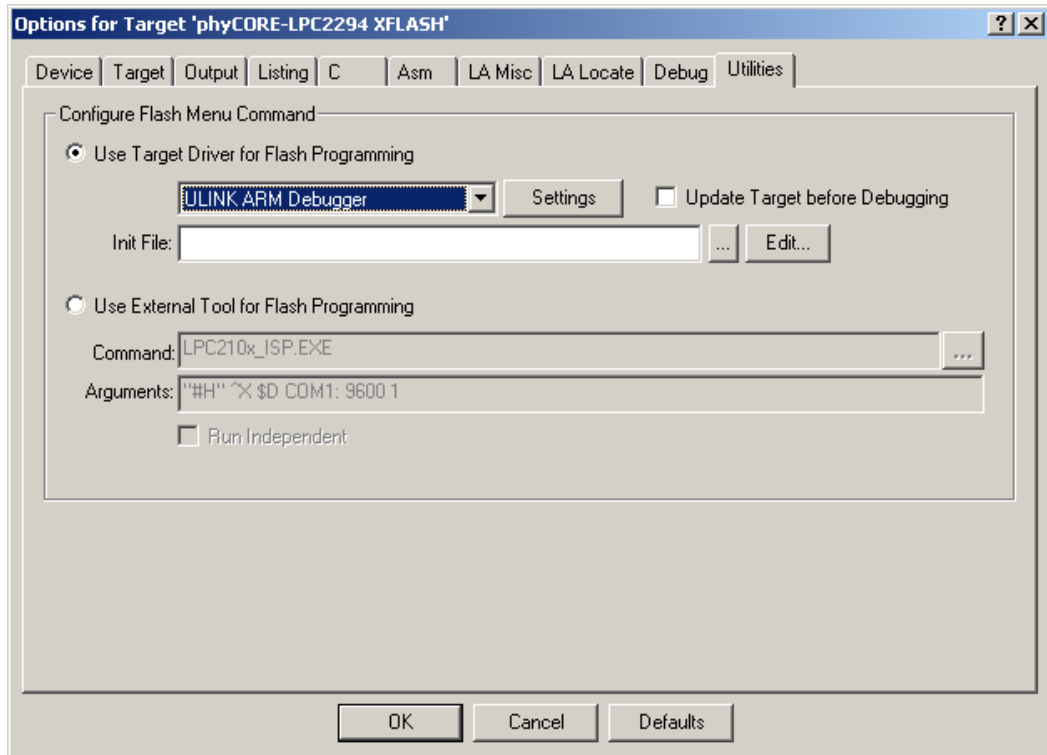
### Note:

Please refer to the Philips LPC2294 User Manual section "Memory Mapping Control" for more details.



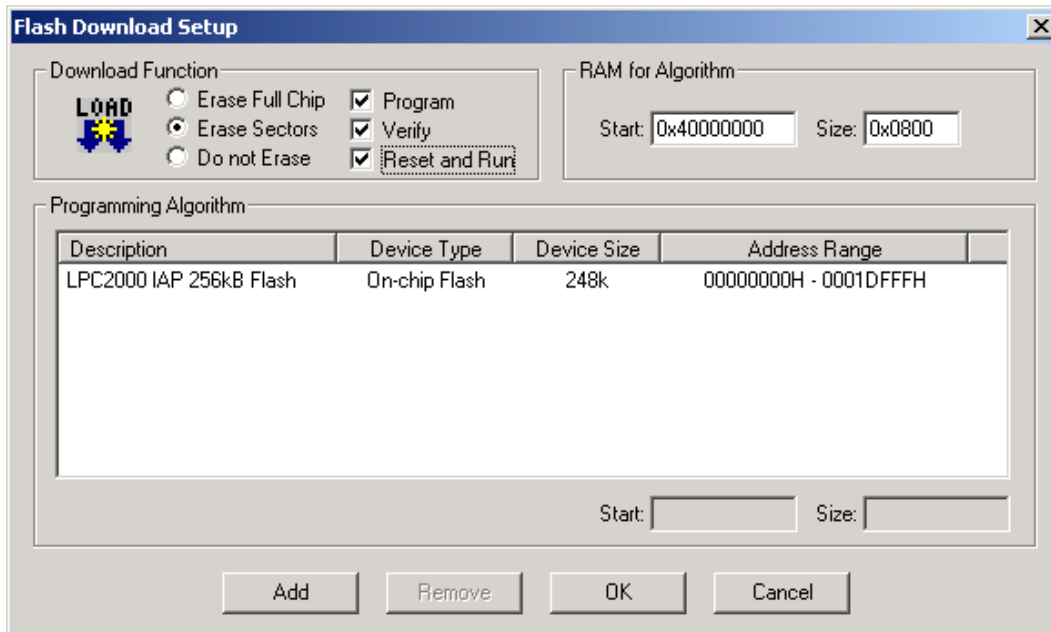
## Configure the Utilities Options

- Select the **Utilities** tab.
- Select the *Use Target Driver for Flash Programming* option.
- In the pull-down menu select *ULINK ARM Debugger*.



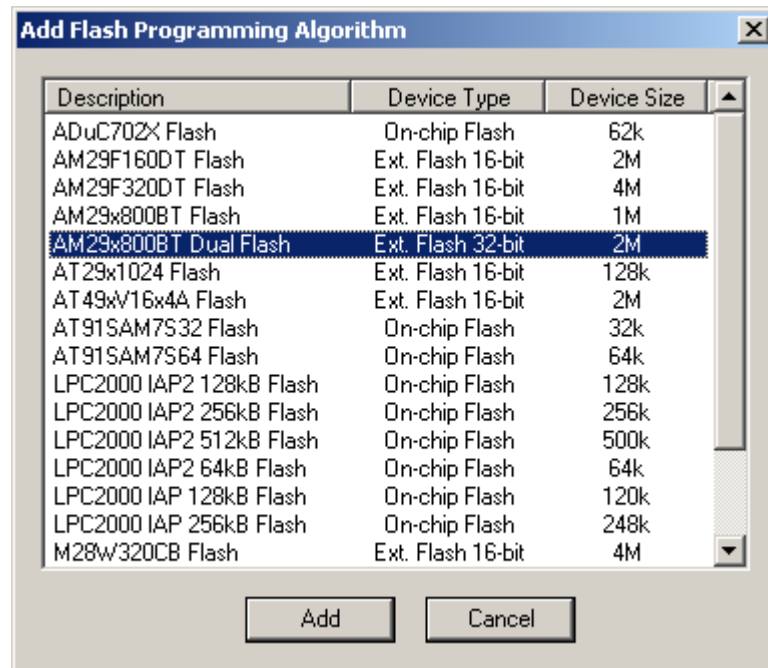
- Click the *Settings* button.

- In the **Flash Download Setup** window be sure that the following *Download Function* options are selected: *Program*, *Verify*, and *Reset and Run*. Checking the *Reset and Run* box will execute the downloaded *Blinky2* code at the end of the Flash programming sequence without having to push the Reset button (S\_2) on the Development Board.



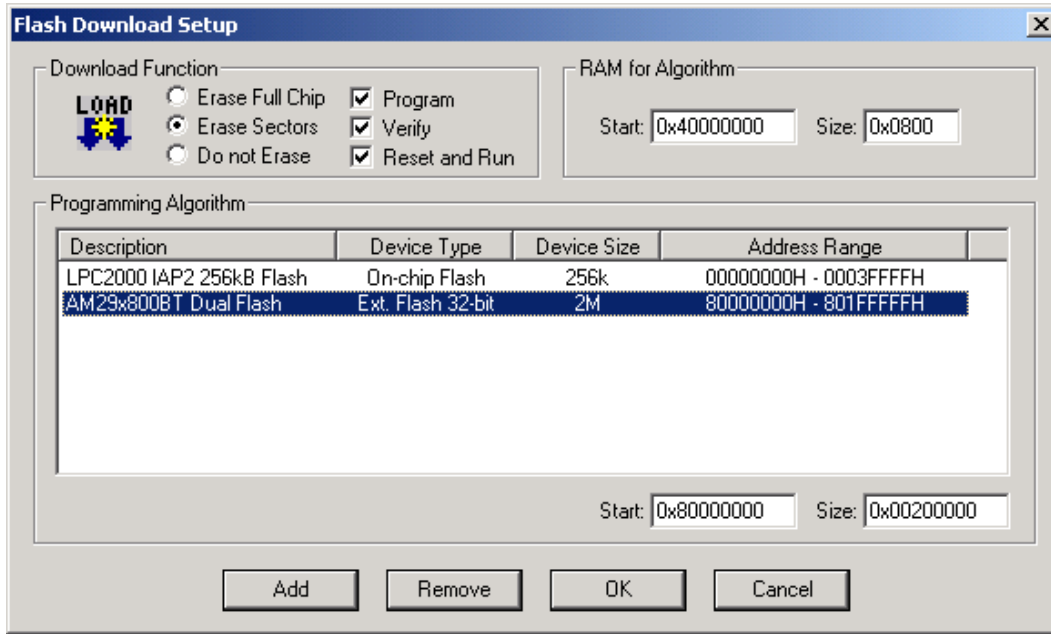
- Click on *Add* to add a *Programming Algorithm*.


- In the **Add Programming Algorithm** window, select *AM29x800BT Dual Flash* and click *Add*. The AM29DL800BT device is the external Flash memory populating the phyCORE®-LPC2294 module.



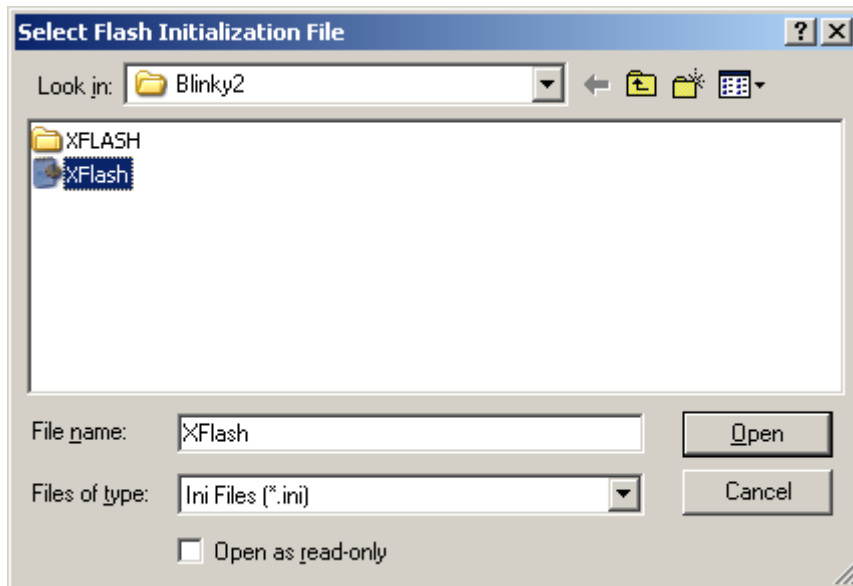
- Click on *Add*.

- In the *RAM for Algorithm* fields, set the *Start* address to **0x40000000** and address *Size* to **0x0800** as shown below.
- In the *Programming Algorithm* window click on the *AM29x800BT Dual Flash* device and set the *Start* address to **0x80000000** and *Size* to **0x00200000** as shown below and click *OK*.



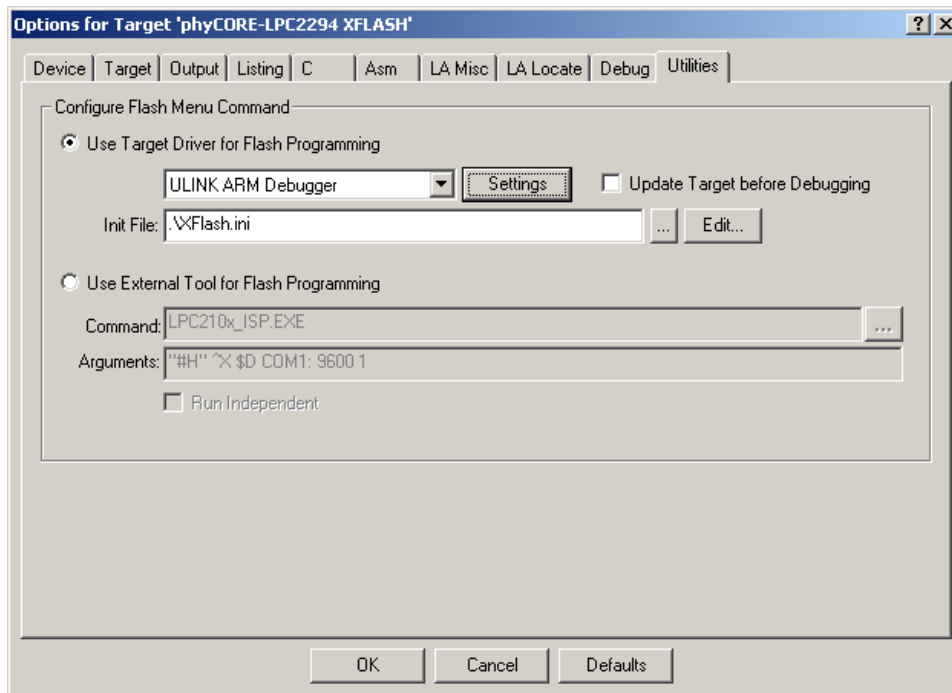
- Back in the **Utilities** tab, select the browse button  in the *Init File:* line to add the Flash initialization file.

- In the **Select Flash Initialization File** pop-up window browse to *C:\PHYBasic\pC-LPC2294\Demos\Keil\Blinky2* and select the *XFlash.ini* file.



- Click the *Open* button.


The **Utilities** tab should appear as follows:



- Click *Ok*.
- In the main  $\mu$ Vision3 menu select *File / Save All*.

### 3.4 Building the Project

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


- Build the desired target by either selecting the build icon  on the build toolbar or in the main menu select *Project / Build target*.

If any source file of the project contains any errors, they will be shown in the **Output Window - Build** tab. Use the editor to correct the error(s) in the source code and save the file and repeat the build.

If there are no errors, the code is ready to be downloaded into the Flash memory.



### 3.5 Downloading Code into Flash Memory

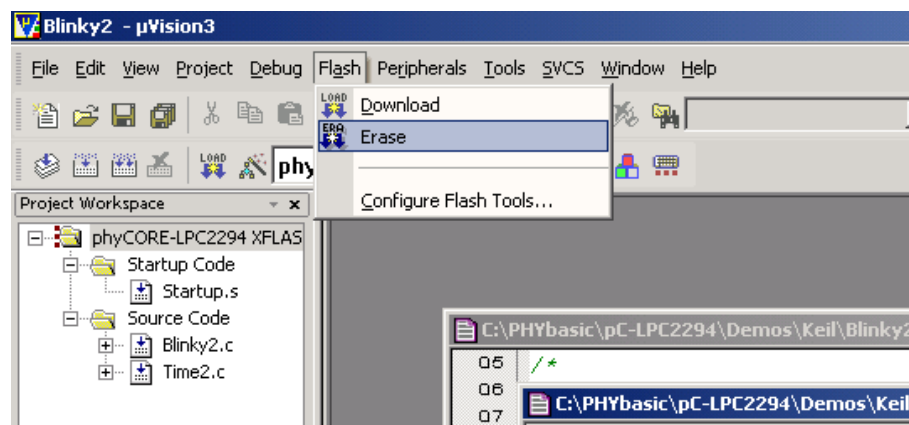
- In the **Select Target** pull down menu be sure that the *phyCORE-LPC2294 XFLASH* target is selected.
- Download the code into Flash memory by either selecting the *Download to Flash Memory* icon  on the build toolbar or in the main menu select *Flash / Download*.
- Wait until the programming is complete. This is indicated by the "**Verify OK**" message. The download utility will perform a reset and the code will execute without further user interaction.

Successful execution of the program will flash the LED D3 with equal on and off duration but with longer cycles compared to the original *Blinky* demo.

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

Before starting with the following debugging section we need to erase the external Flash.

- In the main  $\mu$ Vision3 menu select *Flash /Erase*.



- The Flash erase status will be displayed at the bottom of the  $\mu$ Vision3 window and take a few seconds.



## 4 Debugging

This Debugging section provides a basic introduction to the debug functions included in the Keil ARM7/ $\mu$ Vision3 evaluation tool chain. Using an existing example, the more important features are described. For a more detailed description of the debugging features, *please refer to the appropriate manuals provided by Keil.*

The  $\mu$ Vision3 Debugger offers two operating modes that can be selected in the *Project/Options for Target phyCORE-LPC2292/94* dialog:

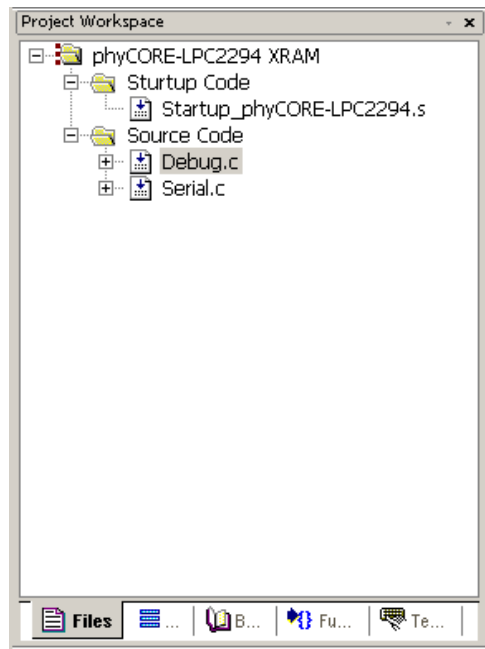
- The **Simulator** allows PC-based simulation of most features of the LPC2294 microcontroller without actually having target hardware. You can test and debug your embedded application before the hardware is ready.  $\mu$ Vision3 simulates a wide variety of peripherals, including external I/O and timers. The peripheral set is configured when you select a CPU from the device database for your target.
- USB-JTAG debugging interface adapters such as the **Keil ULINK**, allow target-based debugging. With the ULINK interface you may connect directly to the target hardware using the JTAG interface. Debugging on the target hardware also enables the testing of peripheral components of the application and real-time program execution.


The following examples utilize the **ULINK ARM Debugger** environment.

## 4.1 Creating a Debug Project and Preparing the Debugger

### 4.1.1 Creating a New Project


- Start the Keil  $\mu$ Vision3 environment and close all projects that might be open.
- Open the **Project** menu and create a **New Project** called *Debug.uv2* within the existing project directory *C:\PHYBasic\pC-LPC2294\Demos\Keil\Debug* (default location) on your hard drive. Select the *Philips LPC2294* in the CPU Vendor Data base list.
- Rename the target of your project within the **Project Workspace** window - **Files** tab into *phyCORE-LPC2294 XRAM*.
- Rename the file group *Source Group 1* as *Startup Code* and add one additional file group named *Source Code*.
- Add *Startup\_phyCORE-LPC2294.s* to the file group *Startup Code*, (In the *File of type* pull-down menu, select: "*Asm Source file (\*.s\*; \*.src\*; .a\*)*" to see this file).
- Add the files *Debug.c* and *Serial.c* to the file group *Source Code* from within the **Project Workspace**.
- Your project should now appear as follows:



- 
- Go to **Project / Components, Environment, Books...** from the menu or select the  icon on the toolbar. Select the **Folders/Extensions** tab and be sure that *Use Keil ARM Tools* under *Select ARM Development Tools* box is selected.
  - Save the project.
  - Double-click on the *Debug.c* file to open the source code window.

At this point you have created a project called *Debug.uv2*, consisting of the C source files called *Debug.c* and *Serial.c* and the assembler file *Startup\_phyCORE-LPC2294.s*.

#### 4.1.2 Setting Options for Target

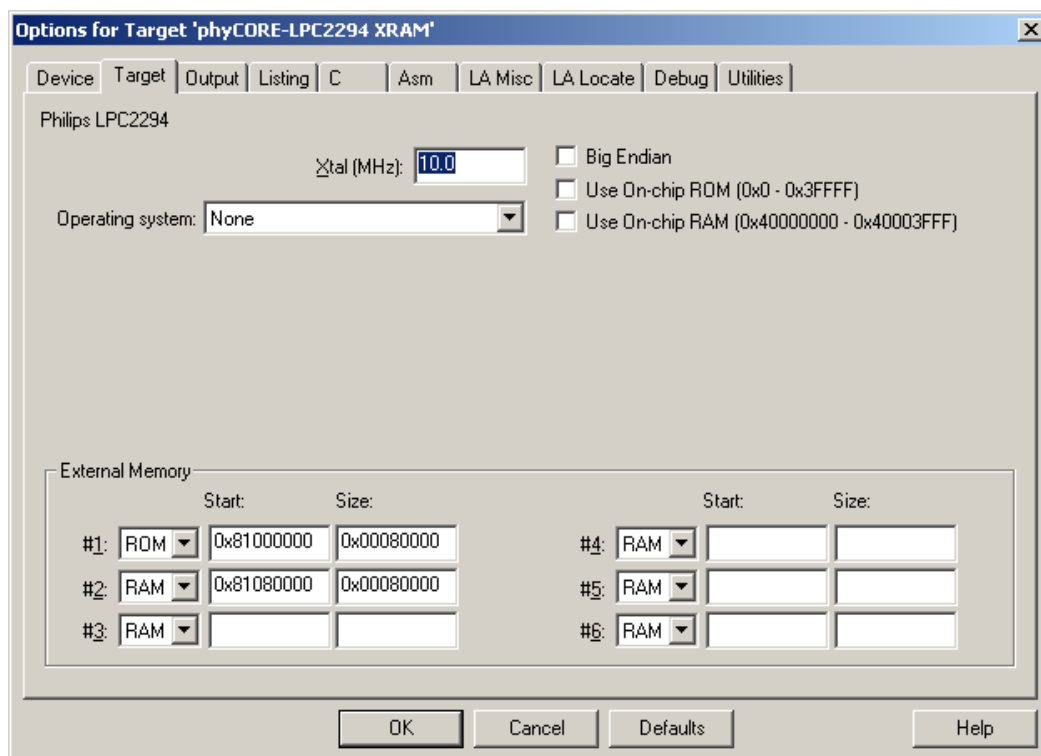
- In the **Select Target** pull-down menu make sure the *phyCORE-LPC2294 XRAM* target is selected.
- Configure options for target by selecting the *Options for Target* icon  on the build toolbar or right-click on the *phyCORE-LPC2294 XRAM* target in the **Project Workspace** window and select *Options for Target 'phyCORE-LPC2294 XRAM'*.

## Configure the Target Options

- In the **Target** tab be sure that *Xtal* is set to **10 MHz**, and *Use On-chip ROM (0x0-0x3FFFF)* as well as *Use On-chip RAM (0x40000000-0x40003FFF)* are not checked. Set the *External Memory #1* to **ROM** and *#2* to **RAM** and set the *Start* and *Size* as follows:

ROM:    **0x8100 0000** (Start)       **0x0008 0000** (Size)

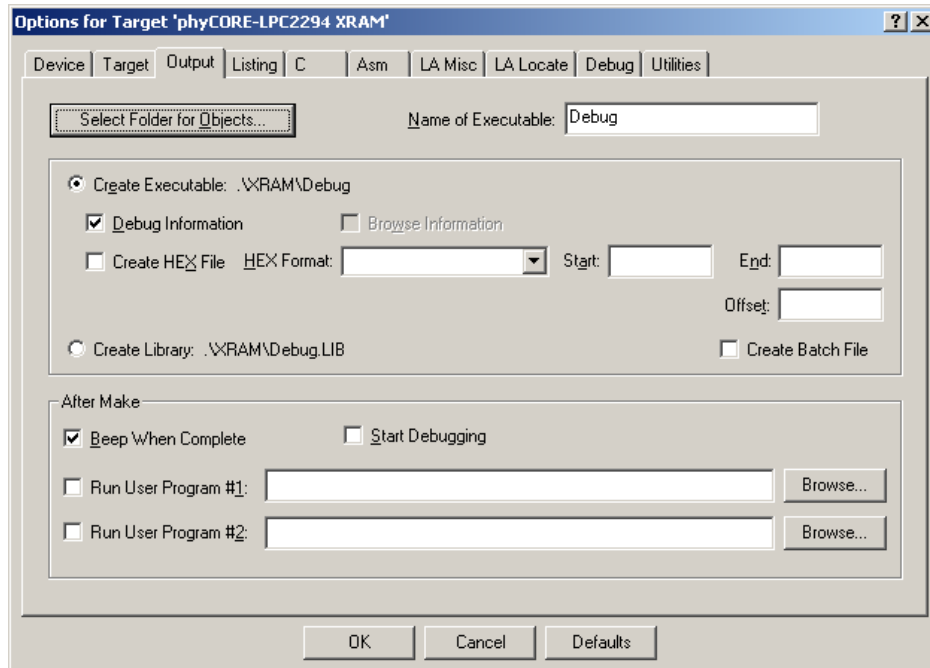
RAM:    **0x8108 0000** (Start)       **0x0008 0000** (Size)



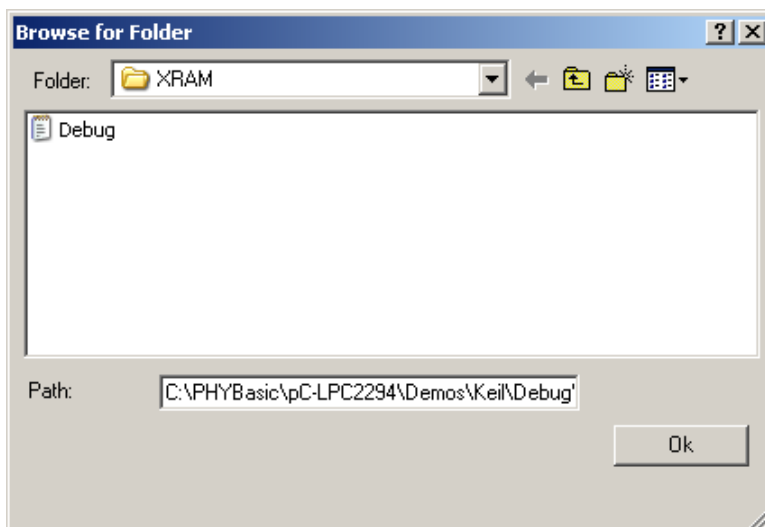
It is necessary to configure the External Memory *Start* and *Size* settings so that the combined user code and data does not exceed the physical size of the RAM. The phyCORE<sup>®</sup>-LPC2294 standard version features 1 MByte of external SRAM.

## Configure the Output Options

- In the **Output** tab be sure that *Create Executable*, *Debug Information* and *Beep When Complete* options are selected.



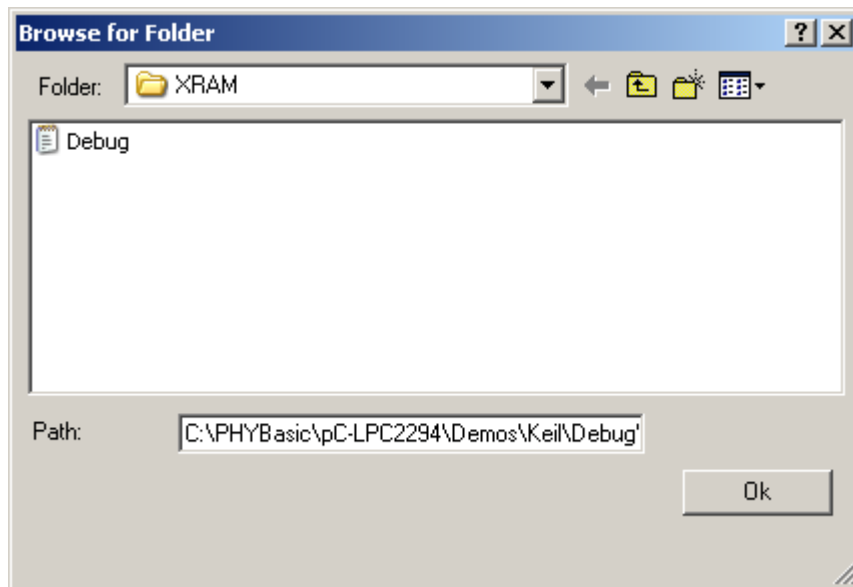
- Click on the *Select Folder for Objects* button.
- Browse to the folder:  
*C:\PHYBasic\pC-LPC2294\Demos\Keil\Debug\XRAM.*



- Click *Ok*.

## Configure the Listing Options

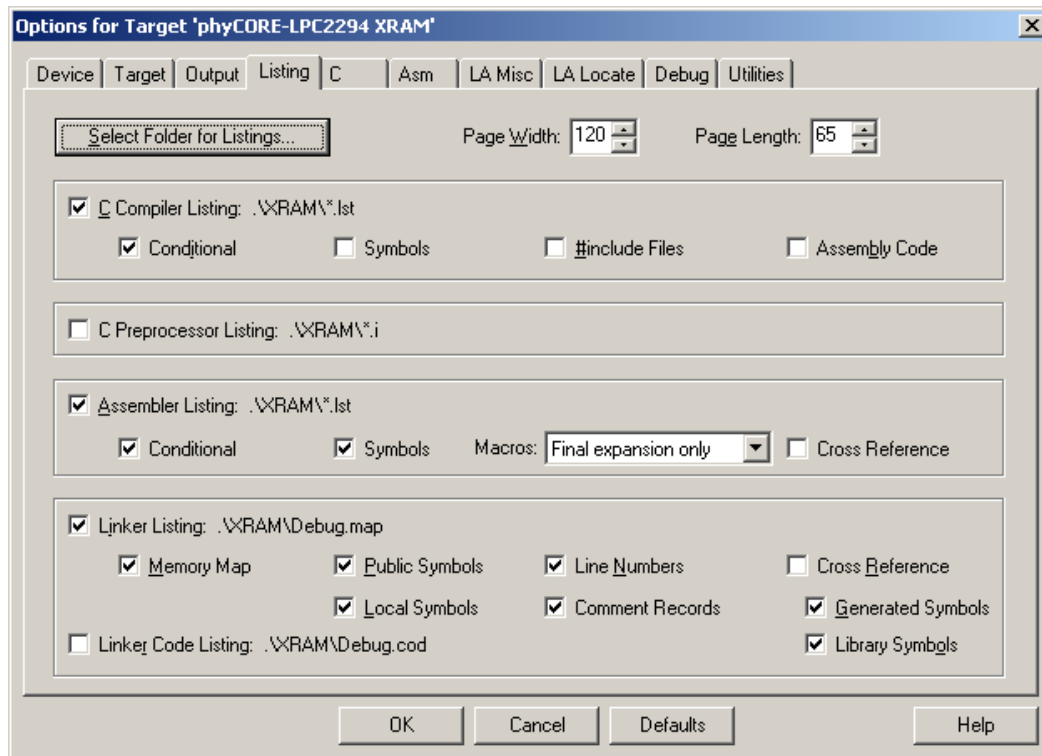
- In the **Listing** tab, leave the default settings.
- Click on the *Select Folder for Listings* button.
- Browse to the folder:  
***C:\PHYBasic\pC-LPC2294\Demos\Keil\Debug\XRAM.***



- Click *Ok*.



- The **Listing** tab should appear as follows:

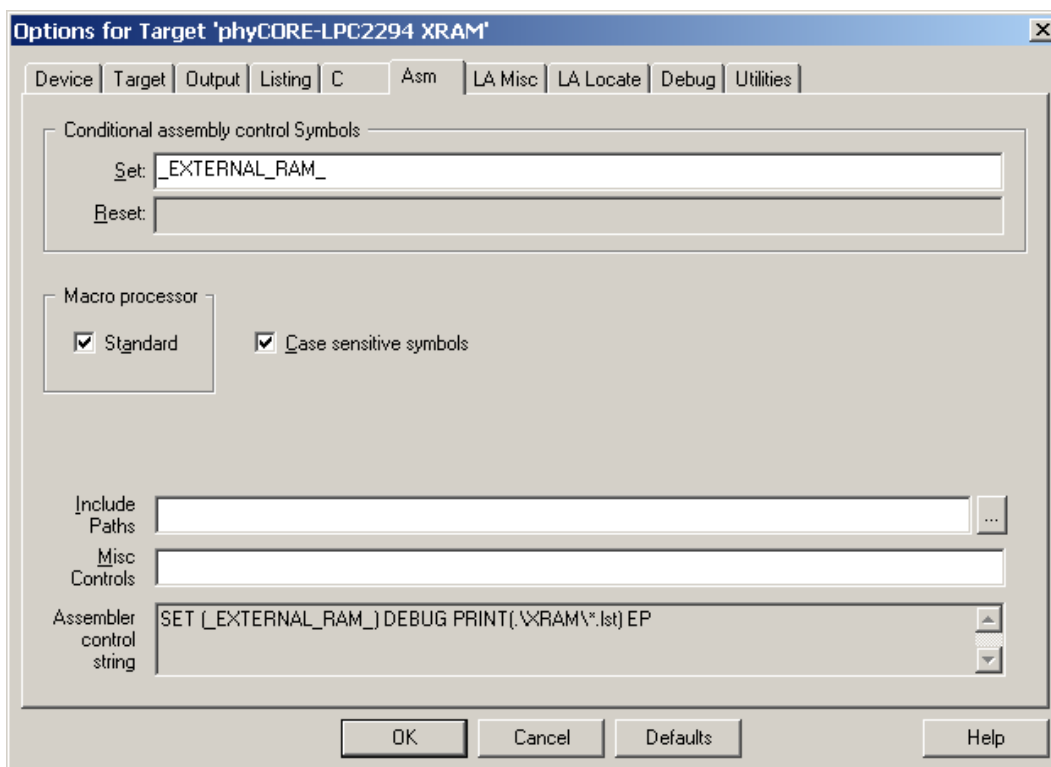


## Configure the Asm Options

- Change to the **Asm** tab. In the *Conditional assembly control Symbols*, *Set* field, type: **`_EXTERNAL_RAM_`**. This will set the LPC2294 MEMMAP register to *User RAM Mode*, where the interrupt vectors are re-mapped to static RAM.

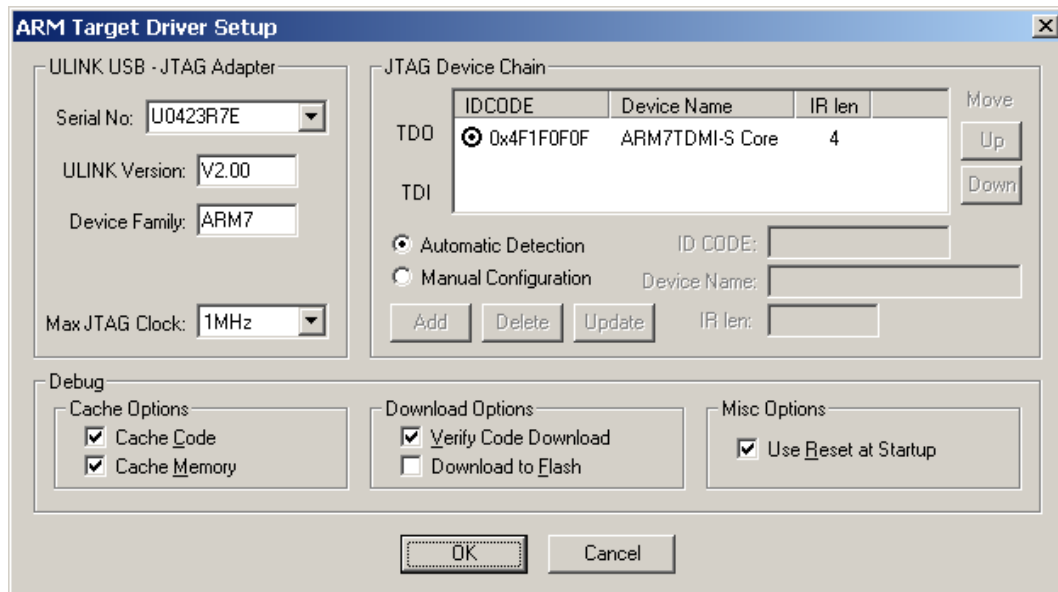
### Note:


Please refer to the Philips LPC2294 User Manual section "Memory Mapping Control" for more details.



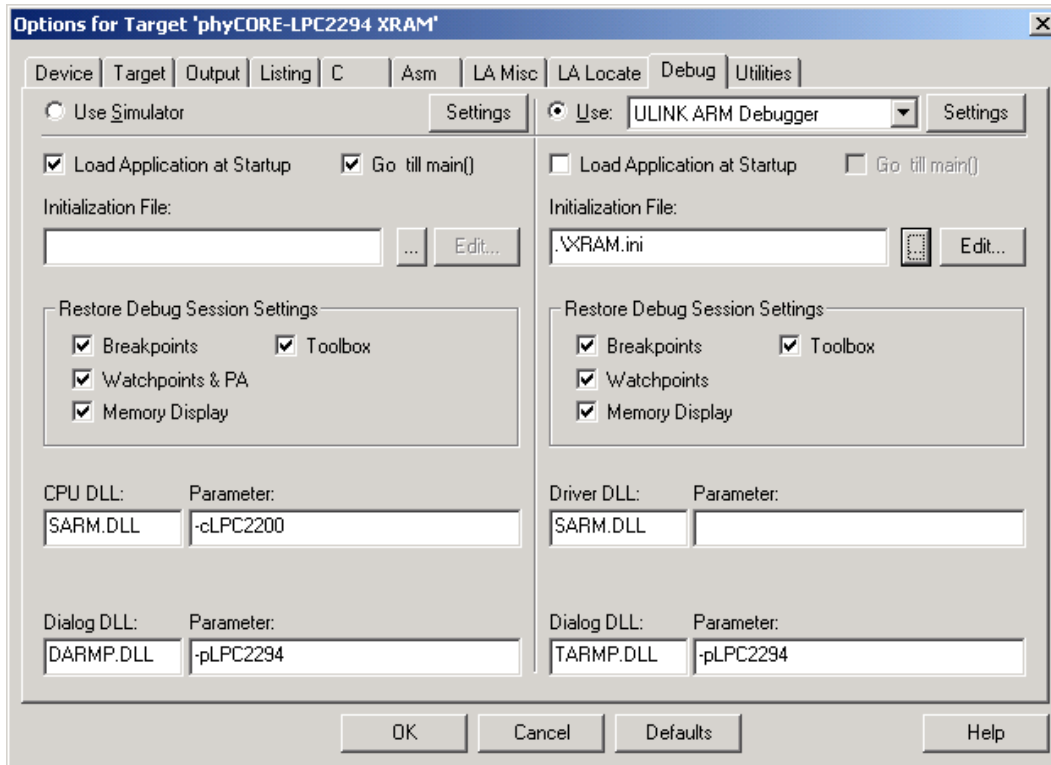
## Configure the Debug Options

- In the **Debug** tab, check the *Use: ULINK ARM Debugger* option.
- Configure the *ARM Target Driver Setup* by clicking the *Settings* button. The *ARM Target Driver Setup* settings should be as follows:



- Click OK to return to the *Options for Target* window.
- Be sure that the *Load Application at Startup*, under the ULINK ARM Debugger setting, is **NOT** checked. Loading the application code and the go till main function are executed by the *XRAM.ini* file (see below).
- Add the correct *Initialization File* by clicking the browse button  and select *XRAM.ini*.

- The *Debug* tab should now appear as follows:




- Click *OK* to save all the settings.
- In the main  $\mu$ Vision3 menu select *File / Save All*.

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

- Build the target by either selecting the build icon  on the build toolbar or in the main menu select *Project / Build target*.

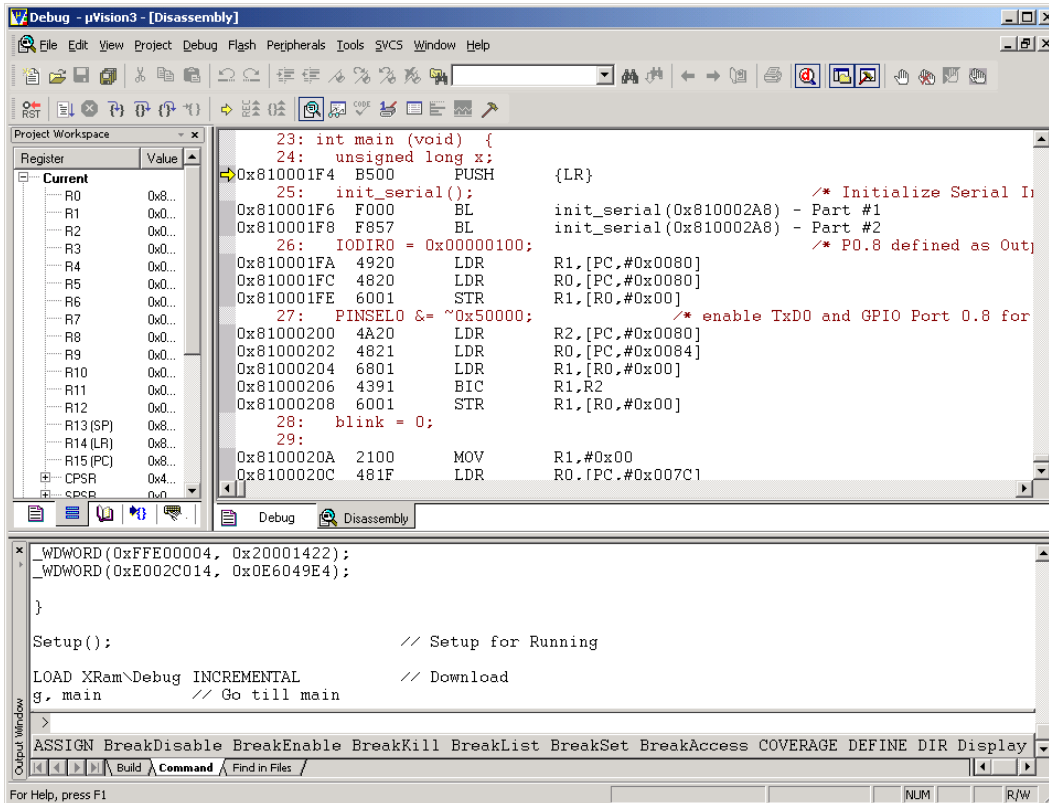
If there are no errors, the code is ready to be downloaded into the external SRAM for further debugging steps. Before starting the debugger first open HyperTerminal again using the same settings as described in *section 2.4.2*. This allows you to monitor the `printf` outputs over the RS-232 port.

## 4.2 Starting the Debugger

- To start the ARM7/ $\mu$ Vision3 debug environment, click on the debugger icon  on the  $\mu$ Vision3 toolbar.
- You will see a blue status bar from left to right at the bottom of your screen indicating the download process of the debug program.

If a problem occurs during data transfer, an error message will be displayed. If this should occur, make sure the target hardware is properly connected to a power supply and the host-PC using the Keil ULINK device (*refer to section 2.2*).

If the data transfer was successful, a screen similar to the one shown below will appear. The **Project** window changes to the **Register** page and the **Disassembly** window becomes active. The debug toolbar is also displayed. In the lower part of the debug screen you will see the **Command** and **Watch** windows.



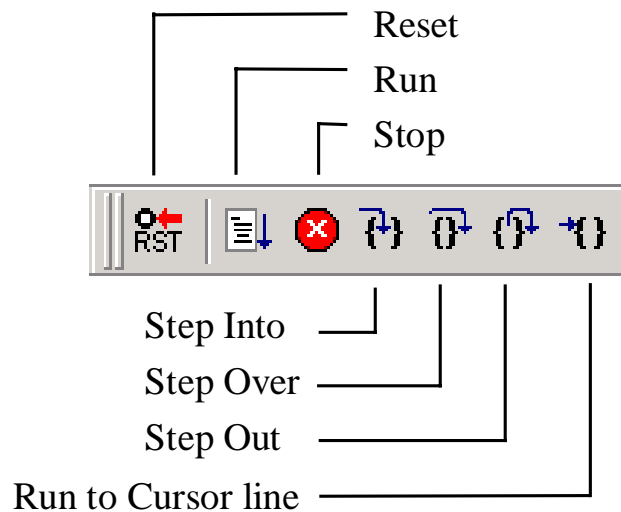
You may need to open, resize and /or move some windows to make your screen look similar to the screen capture. You can open inactive windows by choosing the desired window from the **View** pull-down menu.




- Change to the *Debug.c* source window by clicking on the corresponding tab.


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


### 4.3 Keil $\mu$ Vision3 Debug Features


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





- The first button on the debugger toolbar is the *Reset*  button.  
The *Reset* command sets the program counter to 0.
- The button to the right of the *Reset* button starts the *Run*  command.  
Clicking this button runs the program without active debug functions. To stop program execution at a desired point, a breakpoint can be placed before the *Run* button is pushed.
- The next button on the debugger toolbar is the *Stop*  button.  
The *Stop* button interrupts and stops the running program at an undetermined location.

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


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

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

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

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

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


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

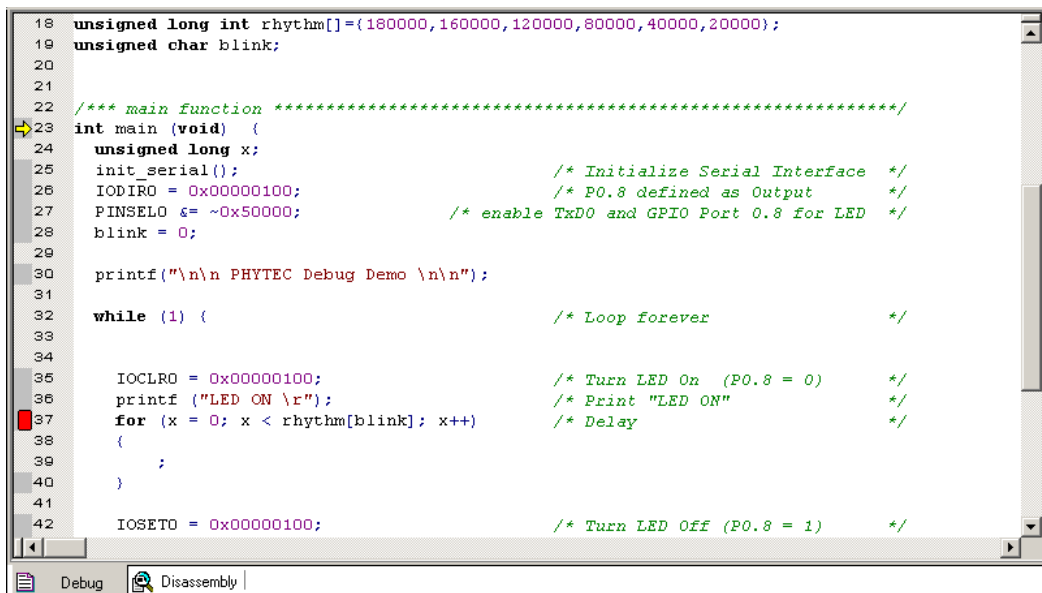
The *Run to Cursor line* command executes the program to the current cursor position within the code window. This allows use of the cursor line as a temporary breakpoint.



## 4.4 Using the Keil $\mu$ Vision3 Debug Features

### 4.4.1 Breakpoints



- Activate the *Debug.c* file by clicking on the *Debug* tab.
- Click in the source code, line 37, at  
for (x = 0; x < rhythm[blink]; x++)
- Click on *Insert/Remove Breakpoint*  to set a breakpoint here. The red marker on the left-hand side of the selected line indicates the breakpoint. You can also set a breakpoint by double-clicking in the desired code line.




```

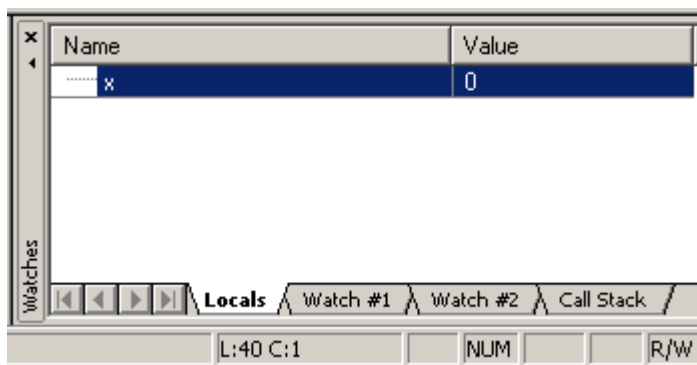
18 unsigned long int rhythm[]={180000,160000,120000,80000,40000,20000};
19 unsigned char blink;
20
21
22 /** main function *****/
23 int main (void) {
24     unsigned long x;
25     init_serial(); /* Initialize Serial Interface */
26     IODIRO = 0x00000100; /* P0.8 defined as Output */
27     PINSEL0 &= ~0x50000; /* enable TxD0 and GPIO Port 0.8 for LED */
28     blink = 0;
29
30     printf("\n\n PHYTEC Debug Demo \n\n");
31
32     while (1) { /* Loop forever */
33
34
35         IOCLR0 = 0x00000100; /* Turn LED On (P0.8 = 0) */
36         printf ("LED ON \r"); /* Print "LED ON" */
37         for (x = 0; x < rhythm[blink]; x++) /* Delay */
38         {
39             ;
40         }
41
42         IOSET0 = 0x00000100; /* Turn LED Off (P0.8 = 1) */

```

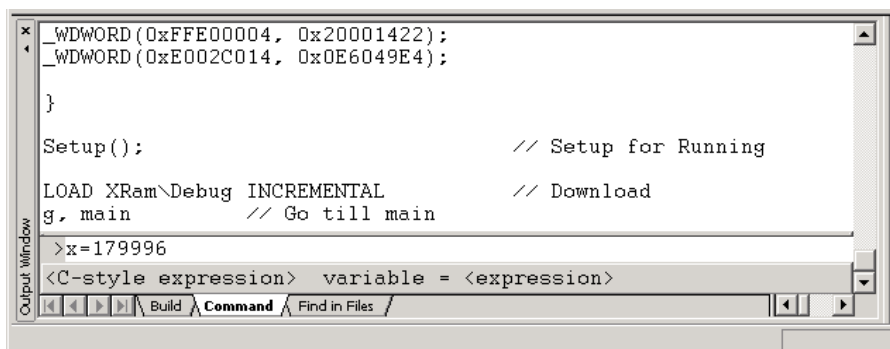
- Click on the *Run*  icon and the program will run and stop at the breakpoint.
- Notice that the LED (D3) on the Development Board now illuminates.
- Also notice the output message in HyperTerminal that comes from the `printf` statement in line # 30 and #36.
- Click again on *Insert/Remove Breakpoint*  to remove the breakpoint.



#### 4.4.2 Single Stepping and Watch Window

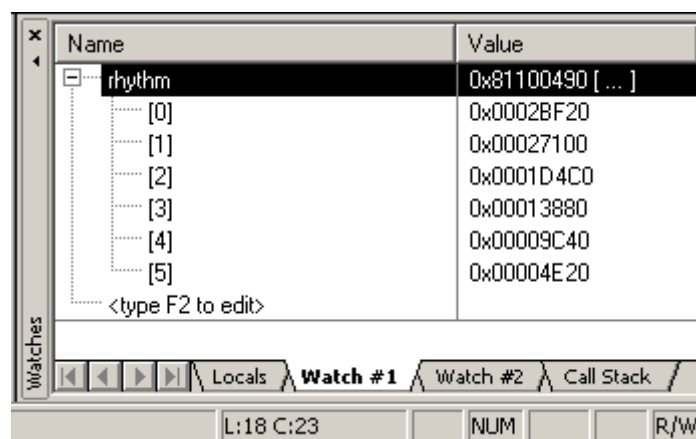
- Open the **Watch and Call Stack Window** from the **View** pull-down menu.
- Click on the **Step Into**  icon to enter the `for{ }` loop.
- The **Watch** window – **Locals** tab automatically shows the value of the local variable `x`. Change the number base from hexadecimal to decimal by right-clicking on the variable.





- Click **Step Over** several times and watch the value of `x` count up.
- As you can see in the source code, the `for{ }` loop will end if `x` becomes equal to the first element of the constant field `rhythm[ ]` which has the value of 180,000. To leave the `wait` function, change the value of `x` by typing `x=179996` in the command line and press <Enter>. Now repeat clicking on **Step Over** until you leave the `wait` function.




- Click in the source code, line 44, at `blink++` and choose **Run to Cursor line** from the debug toolbar. Your program will be executed until it reaches this line.
- Notice that the LED D3 on the Development Board is off now and a new output message appeared in your HyperTerminal window.
- As a last example, the constant "`rhythm[]`" will be evaluated. Go to source code line #18 where the constant "`rhythm[]`" is declared. Right-click on "`rhythm[]`" and choose the "**ADD rhythm to watch window**" -> #1 option. Select the "**Watch #1**" tab at the bottom of the watch window. The constant is shown with its address and a small  sign in front which indicates that "`rhythm[]`" is an array with a group of array elements. Click the  sign to expand the view and to see all array elements of "`rhythm[]`".



## 4.5 Running, Stopping and Resetting




- To run your program without stopping at any time, delete all breakpoints by clicking on the  button.
- Click the **Run**  button.

The LED now blinks at changing on/off intervalls.

You can use the **Stop**  button to stop program execution at any time.

## 4.6 Changing Target Settings for the "Executable Version"

After successfully debugging the program, next change the project and the target settings in order to create an executable file that can then be downloaded to and executed out of the Flash memory on the phyCORE®-LPC2294.

- Make sure the program execution is stopped.
- Exit the current debug session by selecting **Debug/Start/Stop Debug Session**.
- We recommend adding a new target to the debug project. This allows you to use the same C source files but different target settings to be used for debugging (XRAM) and Flash download (XFLASH).
- Click on the  icon in the build toolbar.
- Add another target by selecting the *New(Insert)* icon  in the **Project targets** window. Name the new target *phyCORE-LPC2294 XFLASH*.
- Click *OK* to return to the  $\mu$ Vision3 window.
- In the **Select Target** pull down menu select *phyCORE-LPC2294 XFLASH*.
- Configure options for target by selecting the *Options for Target* icon  on the build toolbar.
- Set all target options as described in *section 3.3* and build the project.
- Download the created **Debug** file to the external Flash memory. For general download procedure information refer to *section 3.5*.
- Start the HyperTerminal program as described in *section 2.4.2*.
- The **Debug** code will start automatically at the end of the download.

Now you can watch your final debug example execute. The HyperTerminal will display the status of the LED.

**Document:** phyCORE<sup>®</sup>-LPC2294 QuickStart Instructions  
**Document number:** L-659e\_3, June 2005

---

**How would you improve this manual?**

---

---

---

---

---

**Did you find any mistakes in this manual?** \_\_\_\_\_ page

---

---

---

**Submitted by:**

Customer number: \_\_\_\_\_

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

---

**Return to:**

PHYTEC America LLC  
203 Parfitt Way SW, Suite G100  
Bainbridge Island, WA 98110  
Fax : (206) 780-9135

---

Published by

**PHYTEC**

---

© PHYTEC Messtechnik GmbH 2005

Ordering No. L-659e\_3  
Printed in Germany