phyCORE®-LPC3180

QuickStart Instructions

Using Keil's ULINK and the Keil ARM/µVision3 Software Development Tool Chain

Note: The PHYTEC Spectrum CD includes the electronic version of the English phyCORE-LPC3180 Hardware Manual

Edition: November 2006

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 America LLC assumes no responsibility for any inaccuracies. PHYTEC America LLC neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. PHYTEC America LLC reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages, which might result.

Additionally, PHYTEC America LLC offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. PHYTEC America LLC 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 sales@phytec.com
Technical Support:	+49 (6131) 9221-31 support@phytec.de	1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Site:	http://www.phytec.de	http://www.phytec.com

1st Edition: November 2006

1 Introduction to the Rapid Development Kit 1
1.1 Rapid Development Kit Documentation 1
1.2 Overview of this QuickStart Instruction 1
2 System Description 2
2.1 Hardware Description 2
2.2 Host System Requirements 2
2.3 The PHYTEC phyCORE®-LPC3180 3
2.4 Keil ARM/µVision3 Software Development Tool Chain 3
3 Getting Started 4
3.1 Installing Rapid Development Kit Software 4
3.1.1 Installing Demos and Documentation from Spectrum CD 4
3.1.2 Installing Keil uVision3 6
3.1.3 Installing NAND Flash Algorithm 6
3.2 Interfacing the phyCORE®-LPC3180 to a Host-PC 7
3.3 Downloading Blinky Example Code with µVision3 10
3.3.1 Build the Project 12
3.3.2 Start the Debugger and Download to Internal RAM 13
4 Debugging 15
4.1 Keil µVision3 Debug Features 15
4.2 Debugging Modes 16
4.3 Starting the Debugger 18
5 Flash Programming 19
5.1 Loading the Secondary Boot Loader 19
5.2 Loading the Blinky Executable to Flash 21
6 Getting More Involved 22
6.1 Creating a New Project 22
6.2 Adding Source Files to the Project 28
6.3 Setting Options for Target 30
6.3.1 Configure the Target Options 30
6.3.2 Configure the Output Options 31
6.3.3 Configure the Listing Options 32
6.3.4 Configure the C/C++ Options 33
6.3.5 Configure the Asm Options 33
6.3.6 Configure the Linker Options 34
6.3.7 Configure the Debug Options 35
6.3.8 Configure the Utilities Options 37
6.4 Building the Project 40

Index of Figures

Figure 1: Default Jumper Settings of the phyCORE Carrier Board with phyCORE-LPC3180 7

Figure 2: JTAG Edge Connector on the phyCORE-LPC3180 (Bottom View) 8

Figure 3: ULINK Connected to the phyCORE-LPC3180 Rapid Development Kit 9

Figure 4: Power Connector 9

1 Introduction to the Rapid Development Kit

The phyCORE-LPC3180 SBC module is designed to be plugged into the appropiate PHYTEC Carrier Board. The phyCORE-LPC3180 Carrier Board contains the I/O connectors as well as any other interface circuitry not provided on the phyCORE module itself. The phyCORE module, combined with the PHYTEC Carrier Board, provides a platform to jump start embedded designs and propel concept to prototype and finished product.

Each PHYTEC Rapid Development Kit contains an SBC module mounted on an applicable Carrier Board, cables, power supply, printed schematics, applicable evaluation software development tool CDs, and the PHYTEC Spectrum CD. The PHYTEC Spectrum CD-ROM provides this QuickStart guide, complete electronic documentation and demo programs.

1.1 Rapid Development Kit Documentation

This "Rapid Development Kit" (RDK) includes the following documentation on the enclosed "PHYTEC Spectrum for ARM7/9 CD-ROM":

- the PHYTEC phyCORE-LPC3180 Hardware Manual
- LPC3180 controller User's Manuals and Data Sheets
- this QuickStart Instruction

1.2 Overview of this QuickStart Instruction

This QuickStart Instruction gives a general "Rapid Development Kit" description, as well as software installation hints and example programs enabling quick out-of-the box sart-up of the phyCORE-LPC3180 in conjunction with the Keil ULINK and ARM9/µVision3 software tools. It is structured as follows:

1) The *"Getting Started"* section describes how to interface the phyCORE-LPC3180 target hardware to a host PC and uses the ***Blinky*** example project to demonstrate the download of user code to the internal RAM using the Keil ULINK and ARM7/µVision3 software tools.

2) The *"Debugging"* section introduces the main debugging features using a new target within the ***Blinky*** example project.

3) The *"Flash Programming"* section shows how to download a secondary boot loader as well as the ***Blinky*** executable into the external NAND Flash

4) The *"Getting More Involved"* section provides step-by-step instructions on how to create a new project, add existing source files, configure all necessary tool options and generate an executable for the phyCORE-LPC3180 using the Keil tools.

2 System Description

2.1 Hardware Description

The following PHYTEC hardware components are included in the phyCORE-LPC3180 Rapid Development Kit (KPCM-031-KEIL) and are necessary for completing the instructions in this application note:

- the PHYTEC phyCORE-LPC3180 (PCM-031)
- the phyCORE-LPC3180 Carrier Board (PCM-976)
- AC adapter supplying 5 VDC /min. 500 mA, center positive
- RS-232 DB9 serial cable (not needed for QuickStart)
- USB A to B cable
- the Keil ULINK JTAG-USB adapter, **only** included in the Keil Rapid Development Kit version[1]
- the PHYTEC Spectrum CD for ARM
- the Keil ARM Development Evaluation Tools CD

2.2 Host System Requirements

- Operating System: Windows 98, Windows NT Version 4, Windows 2000, Windows XP
- CPU architecture: Any x86 32-bit or 64-bit (x64: AMD64 or Intel EM64T) processor
- 30MB Free Hard Disk Space
- 128MB of RAM

For more information and example updates, please refer to the following sources:

http://www.phytec.com - or - http://www.phytec.de
support@phytec.com - or - support@phytec.de

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

[1]: The Keil ULINK is included in the Rapid Development Kit version with the part number KPCM-031-KEIL.

2.3 The PHYTEC phyCORE®-LPC3180

The phyCORE-LPC3180 supports the industry's first 90nm ARM-9 based microcontroller. The new 32-bit MCU high-performance, low power LCP3180 ARM926EJ-S device from NXP Semiconductors (founded by Philips) is the only ARM9 microcontroller that provides a vector floating-point co-processor and integrated USB OTG, as well as the ability to operate in ultra-low-power mode down to 0.9V. The on-board MMU supports many embedded operating systems. Other chip-level features include 7 UARTs, SPI, I^2C, a real-time clock with a separate power domain, and NAND Flash and DDR memory controllers.

Please refer to the phyCORE-LPC3180 Hardware Manual for specific information on board-level features, jumper configuration, memory mapping, pin layout, and carrier board features.

2.4 Keil ARM/µVision3 Software Development Tool Chain

Keil Software development tools for the ARM7, ARM9, and Cortex-M3 microcontrollers are easy to learn and use, yet powerful enough for the most demanding embedded applications. The Keil ARM compiler supports all ARM-compatible devices including the NXP Semiconductors LPC3180 device. For a complete list of supported ARM derivatives go to:

http://www.keil.com/arm/chips.asp

µVision3, 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 µVision3 features including. For more information on Keil ARM/µVision3 tools visit their website at:

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

3 Getting Started

What you will learn with this Getting Started example:

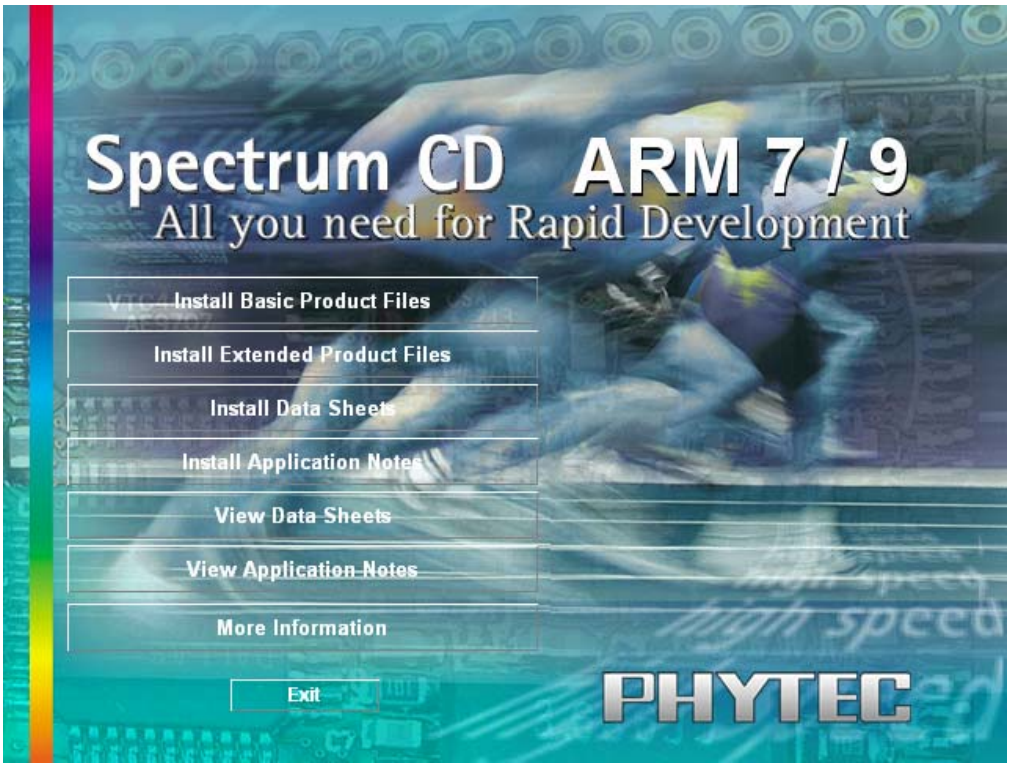
- installing Rapid Development Kit software
- interfacing the phyCORE-LPC3180, mounted on the Carrier Board, to a host-PC using the Keil ULINK
- downloading example user code to the LPC3180 internal RAM memory.

3.1 Installing Rapid Development Kit Software

3.1.1 Installing Demos and Documentation from Spectrum CD

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 demos and documentation required for the Rapid Development Kit. Otherwise the setup program ***start.exe*** can be manually executed from the root directory of the PHYTEC Spectrum CD.

The following window appears:



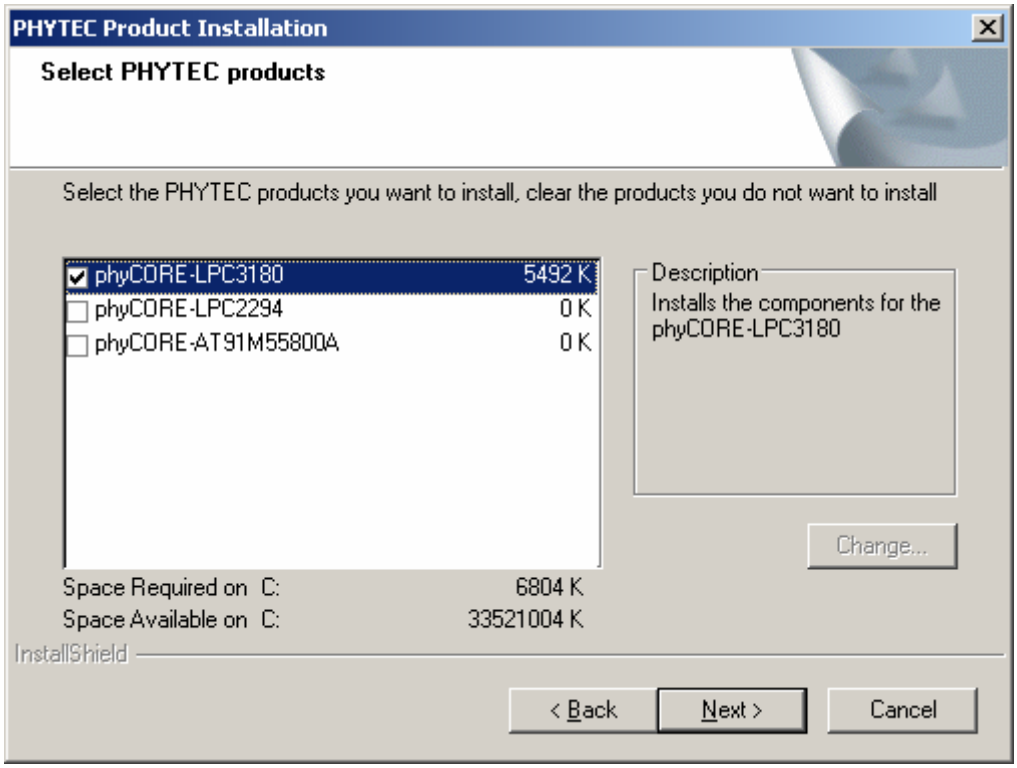
- Choose *Install Basic Product Files* Button.

- 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.

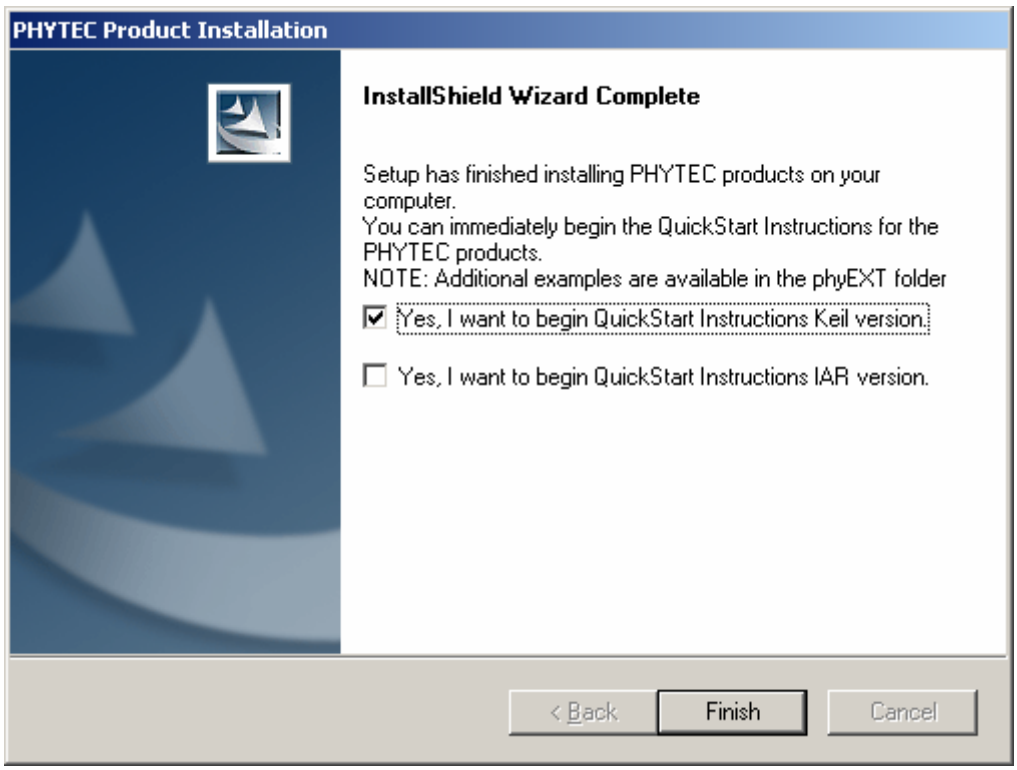


- In the next dialog you must choose whether to copy the selected documentation as *.***pdf** files to your hard drive or to install a link to the file on the Spectrum CD.

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.

- Adobe's Acrobat Reader V7.0 is offered for installation during the next setup step. PHYTEC PDF manuals require use of Acrobat Reader V7.0 or higher.

- At the end of the CD installation you can open the QuickStart instructions manual for the Keil ARM/µVision3 tools. Check the applicable box and click on *Finish*.



3.1.2 Installing Keil uVision3

When you insert the Keil uVision3 CD into the CD-ROM drive of your host-PC, the Keil uVision3 CD should automatically launch a setup program that installs the required software. Otherwise the setup program (*setup.exe*) can be manually executed from the root folder of the CD.

- Install the Keil ARM Evaluation tools, from the enclosed Keil uVision3 CD, following the steps indicated in the install procedure.

- Alternatively, the Keil ARM Evaluation tools can be installed from:

  https://www.keil.com/demo/eval/arm.htm

3.1.3 Installing NAND Flash Algorithm

- Copy the NAND Flash Algorithm files from…
  C:\PHYBasic\pC-LPC3180\pC-LPC3180\Demos\Keil\Flash

- to…
  C:\Keil\ARM\Flash

3.2 Interfacing the phyCORE®-LPC3180 to a Host-PC

Connecting the phyCORE-LPC3180, mounted on the phyCORE Carrier Board, to your computer is simple.

- Ensure proper jumper settings on the phyCORE Carrier Board as shown in *Figure 1*.

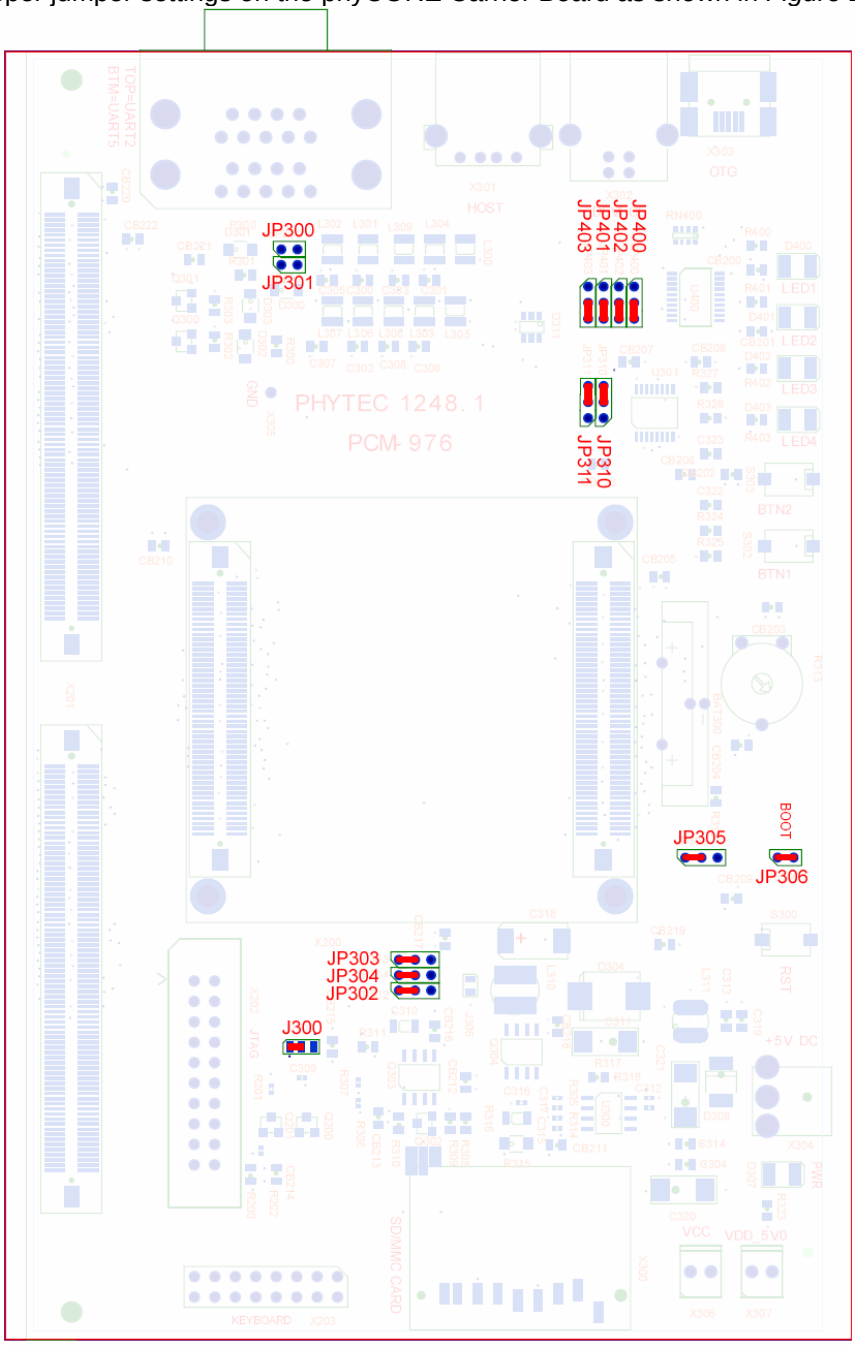


Figure 1: Default Jumper Settings of the phyCORE Carrier Board with phyCORE-LPC3180

- The ULINK JTAG adapter may come with various flat-band cables. In order to connect this device to the phyCORE-LPC3180 module you need to install a flat-band cable with a 2.0 mm pitch connector. If such a cable is not already installed on your ULINK open the enclosure and connect the 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. A 2.54 mm pitch ribbon connector can also be connected to X202 on the Carrier Board directly from the ULINK.
- Connect the 2.0 mm cable connector onto the JTAG pin header rows on the phyCORE module. Make sure that pin 1 on the ULINK cable (black wire) is correctly connected to pin 1 on JTAG connector (located on the under side of the PCB) of the phyCORE-LPC3180 *(refer to Figure 2)*.

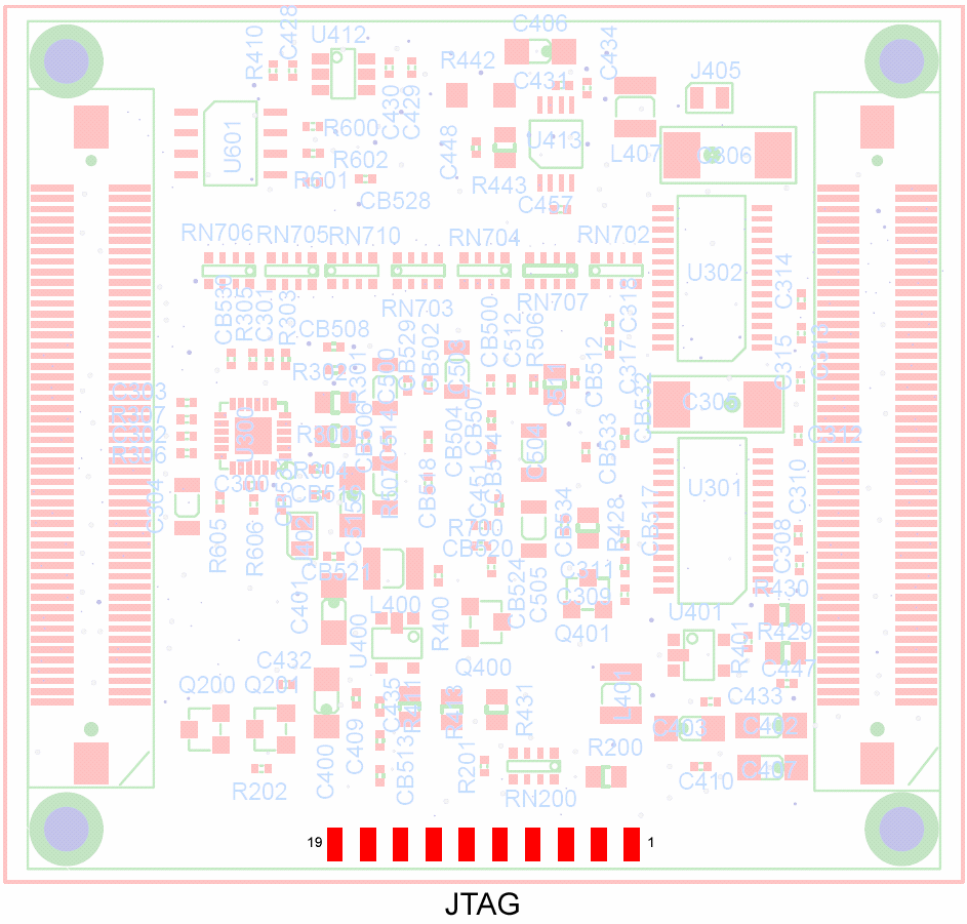


Figure 2: JTAG Edge Connector on the phyCORE-LPC3180 (Bottom View)

- Mount the phyCORE module, pins-down, onto the Carrier Board's receptacle footprint (X200) as shown in *Figure 3* below. Ensure that pin 1 of the module matches pin 1 of the receptacle on the Carrier Board. The JTAG PCB edge connector on the phyCORE-LPC3180 module will point towards the DB-9 connectors on Carrier Board.

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

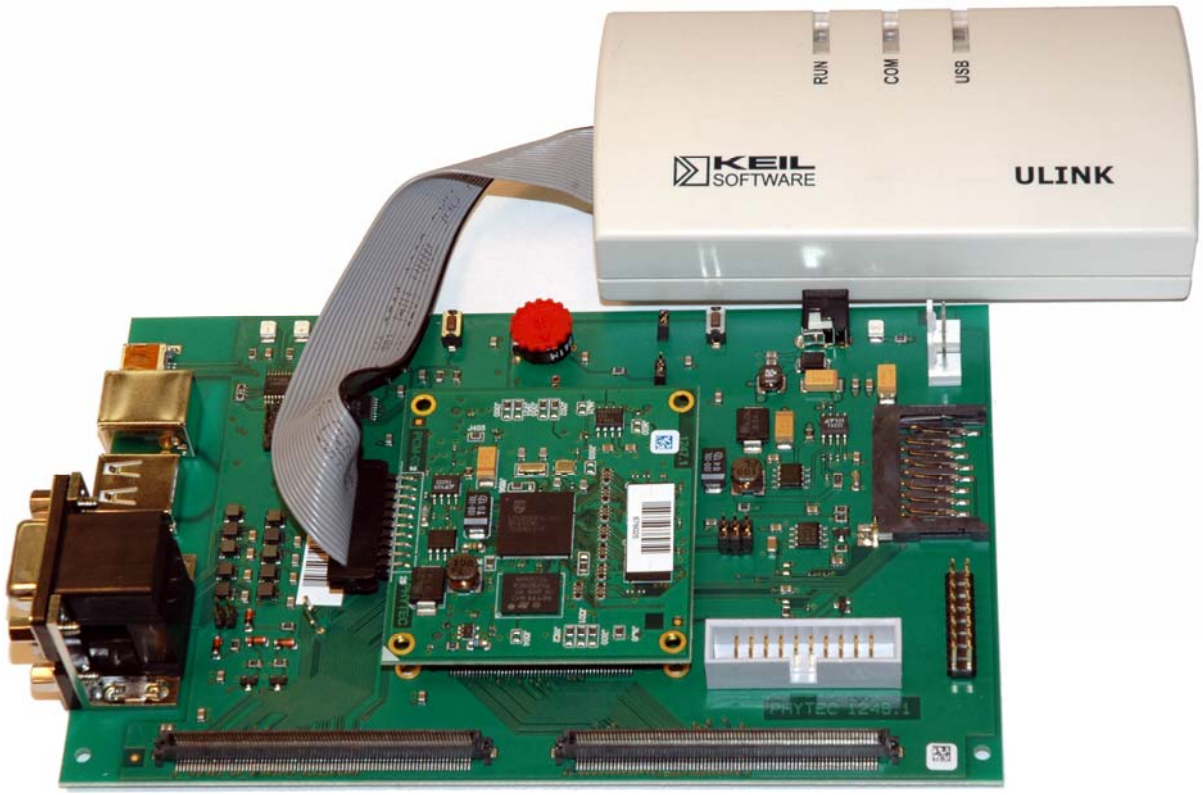


Figure 3: ULINK Connected to the phyCORE-LPC3180 Rapid Development Kit

- Connect the USB end of the ULINK JTAG adapter to the USB port of your host-PC using the included USB cable.
- Connect the included 5V power adapter to the power socket X304 on the phyCORE Carrier Board (*refer to Figure 4 for the correct polarity*).

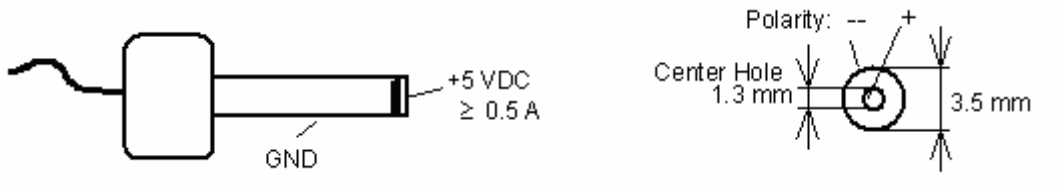


Figure 4: Power Connector

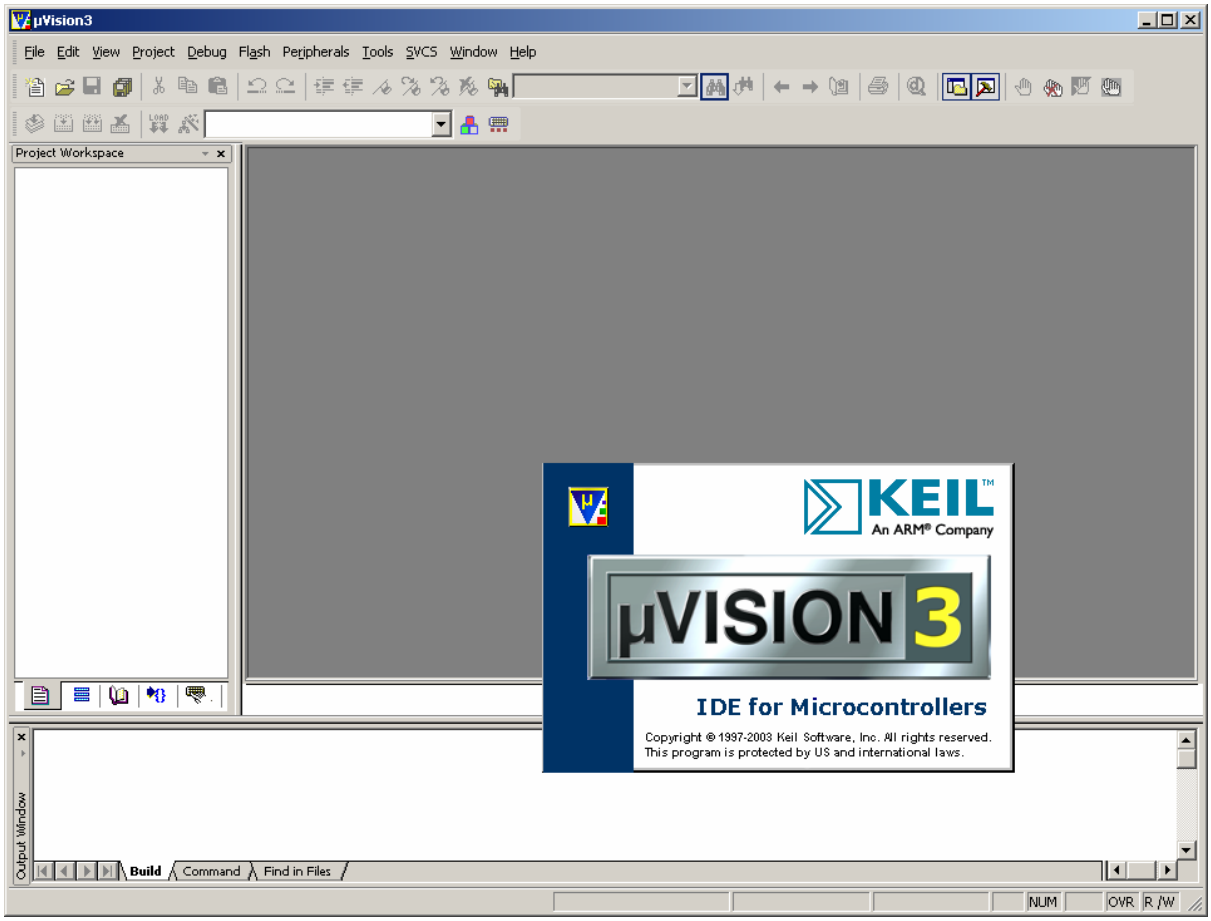
- The red power LED D307, located next to the power socket at X304, should illuminate. This indicates that proper voltage is supplied to the phyCORE module/Carrier Board combination (which is also referred to as "target hardware" within this document).
- The phyCORE module/Carrier Board combination should now be properly connected to a host PC via the Keil ULINK. You are now ready to use the Keil ARM/µVision3 tools to establish communication between the host-PC and target hardware.

3.3 Downloading Blinky Example Code with µVision3

The µVision3 evaluation software development tool chain should have been installed as described in *section 3.1.2*.

- Start the tool chain by selecting Keil µVision3 from within the programs group: *Start\Programs\Keil µVision3* or by double-clicking on the Keil µVision3 icon on your desktop.

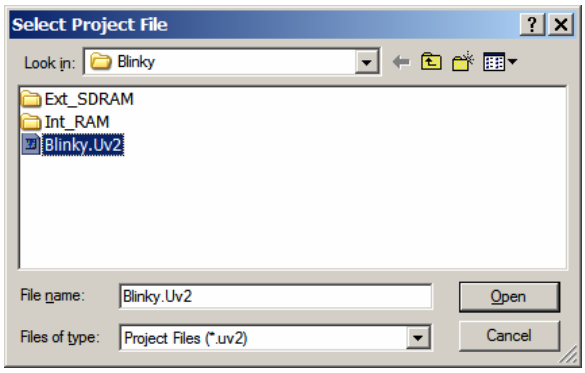
After you start µVision3, the window shown below appears. From this window you can create projects, edit files, configure tools, assemble, link and start the debugger. Close all projects that might be open by selecting *Project / Close Project*.



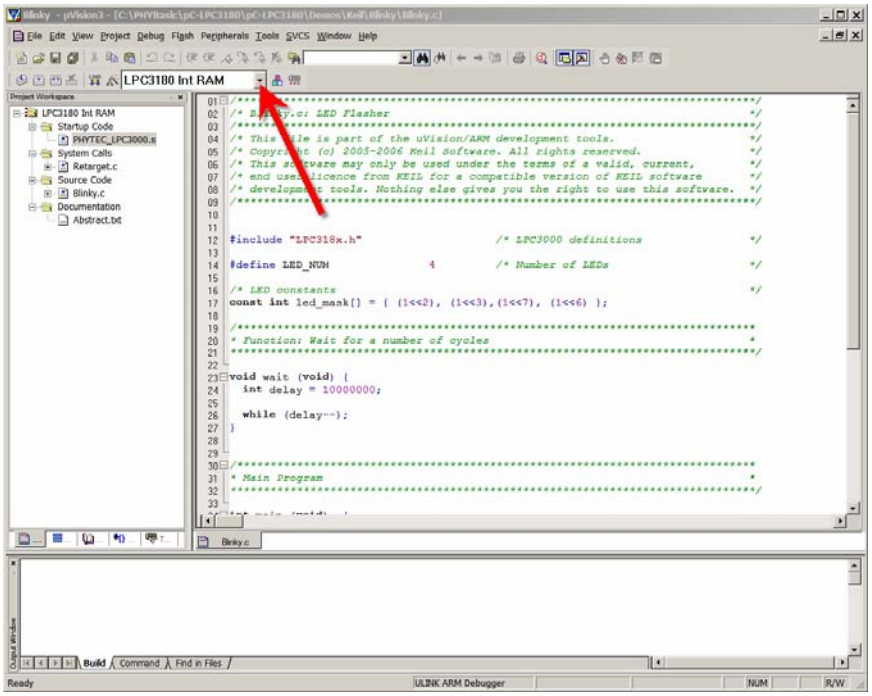
The **Blinky** project, when executed, manipulates the 4 user LEDs on the phyCORE Carrier Board. The **Blinky** example contains two targets: *LPC3180 Int RAM* and *LPC3180 Ext Memory.*

- Open the **Blinky** project from the µVision3 menu *Project / Open Project.*
- Browse to ***C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky***

- Select the **Blinky** project and Click *Open*.



- In the **Select Target** pull-down menu be sure that the *LPC3180 Int RAM* target is selected.



If the ***Blinky.c*** source file is not already visible in the editor window you can open the file by double-clicking on the file name in the project workspace.

3.3.1 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 internal RAM.

3.3.2 Start the Debugger and Download to Internal RAM

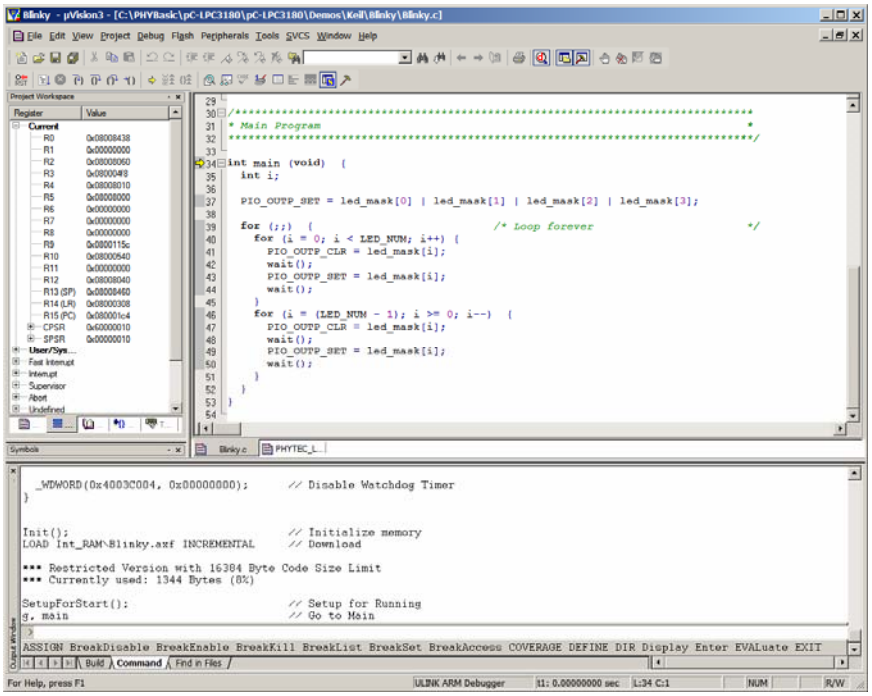
- To start the ARM/µVision3 debug environment, click on the debugger icon  on the µVision3 toolbar or select from the tool menu under Debug.

- While using the ARM/µVision3 Evaluation Tools, the following warning will appear. Please select *OK* and continue.

- 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.

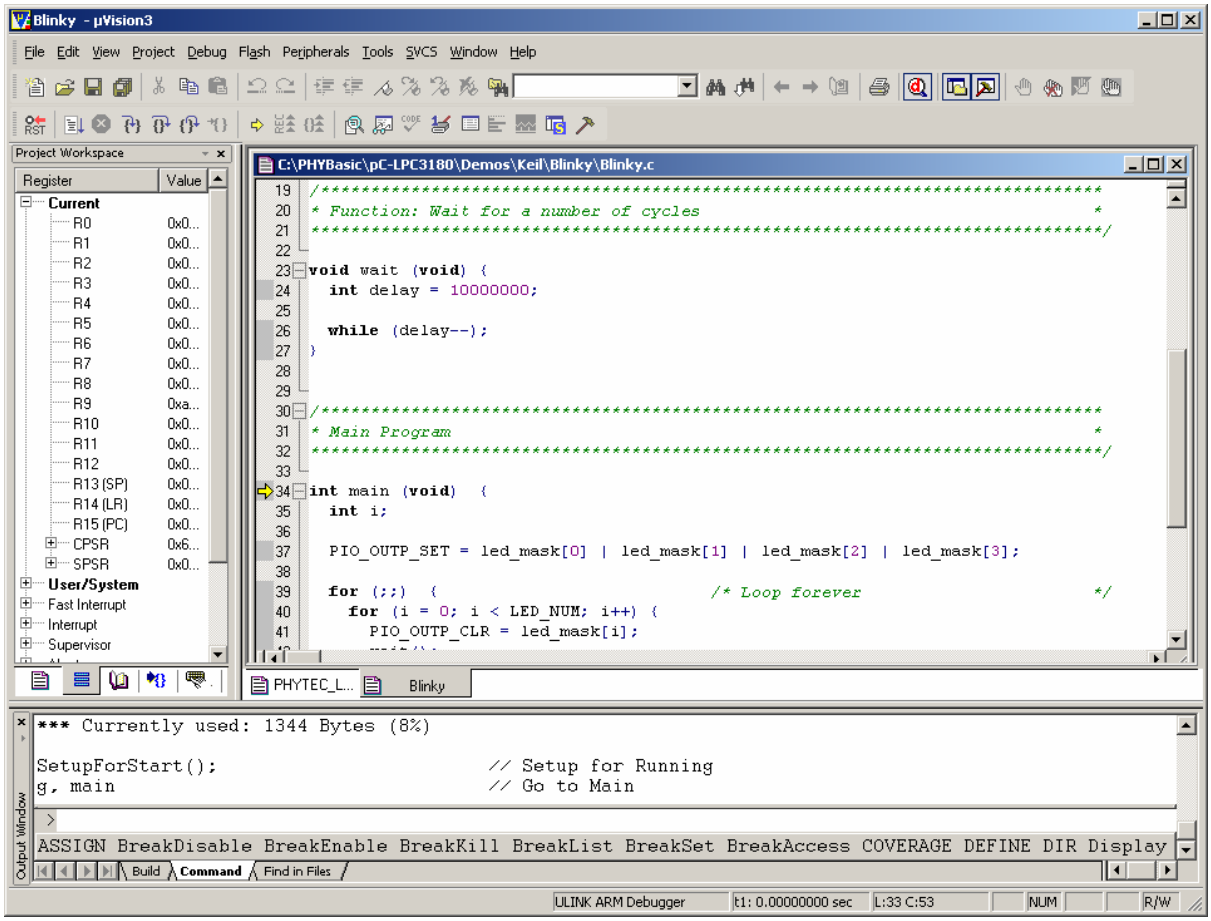
If a problem occurs during data transfer, an error message will be displayed. If "No JTAG Devices Found" error should occur, make sure the target hardware is properly connected to a power supply and the host-PC using the Keil ULINK device (*refer to section 3.2*).

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

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.



- Click on the *Run*  icon and the program will start executing.

Successful execution of the program will flash the LEDs (D400, D401, D402, D403) on the Carrier Board.

4 Debugging

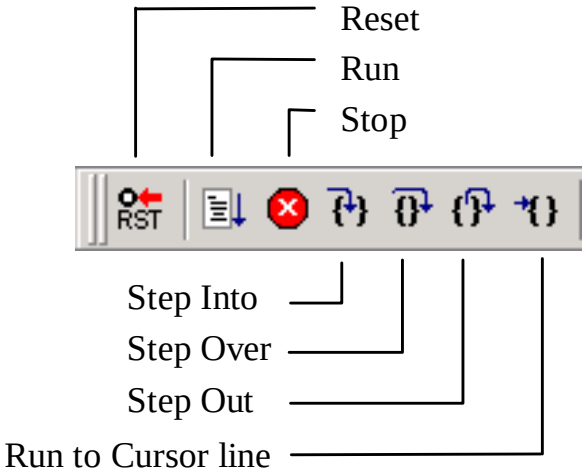
What you will learn with this example:

- familiarize yourself with simple debug functions provided by the Keil µVision3 debug environment
- downloading example user code in the phyCORE-LPC3180 external SDRAM memory

This Debugging section provides a basic introduction to the debug functions included in the Keil ARM/µVision3 evaluation tool chain. For a more detailed description of the debugging features, please refer to the appropriate documentation provided by Keil.

4.1 Keil µVision3 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 debugfunctions. 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 ocation.

- 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.

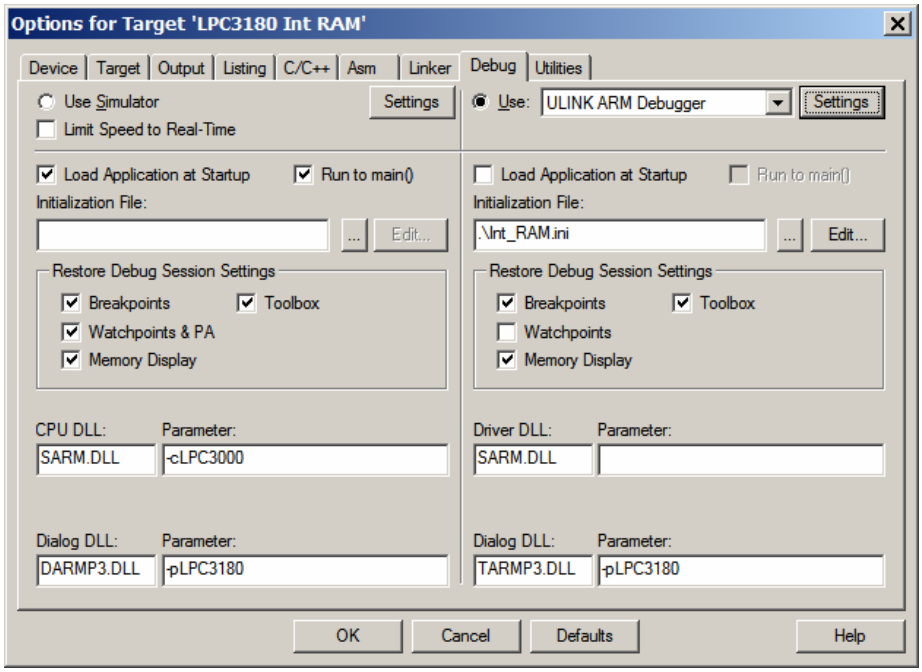
- The last button on the debugger toolbar performs the *Run to Cursor line* command.

  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.

4.2 Debugging Modes

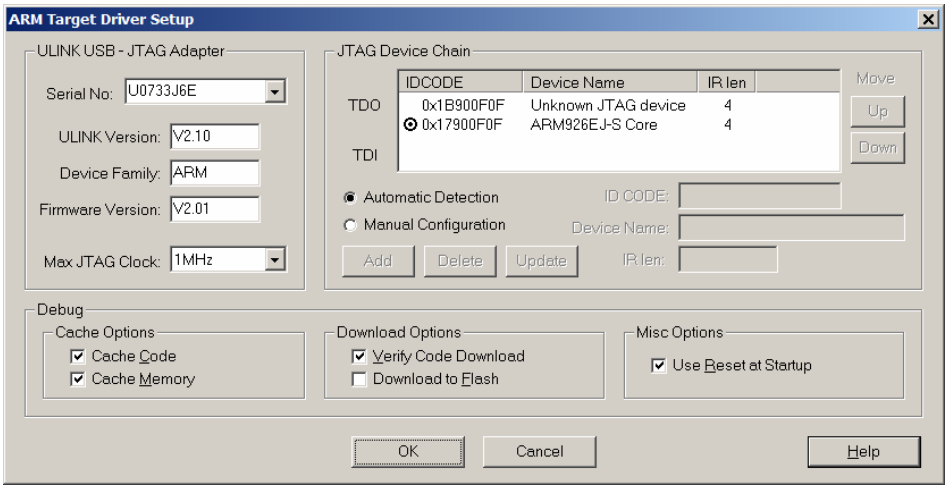
The µVision3 Debugger offers two operating modes that can be selected in the ***Project / Options for Target phyCORE-LPC3180 / Debug*** dialog:

- The **Simulator** allows PC-based simulation of most features of the LPC3180 microcontroller without actually having target hardware. You can test and debug your embedded application before the hardware is ready. µVision3 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.



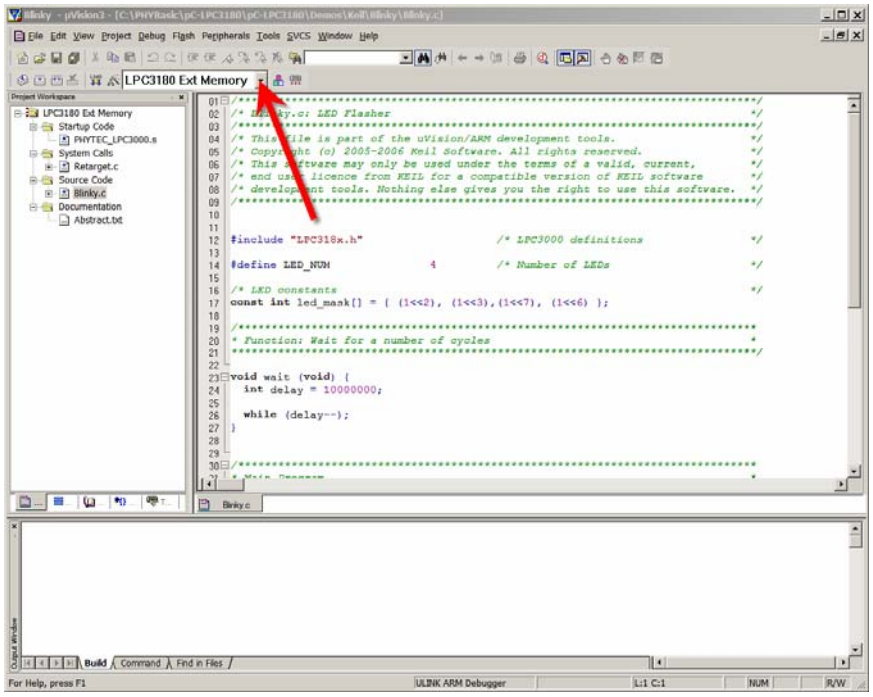
The **Blinky** demo in this section utilizes the **ULINK ARM Debugger** environment.

- The **ULINK ARM Debugger** settings can be viewed by selecting the *Settings* button.



4.3 Starting the Debugger

- Open the **Blinky** project as described in *section 3.3*.
- In the **Select Target** pull-down menu be sure that the *LPC3180 Ext RAM* target is selected.



- To download the code into external SDRAM and start the ARM/µVision3 debug environment, click on the debugger icon  on the µVision3 toolbar.
- Click on the *Run*  con and the program will run.

Successful execution of the program will flash the LEDs (D400, D401, D402 and D403) on the phyCORE-LPC3180 Carrier Board.

5 Flash Programming

What you will learn with this example:

- downloading the Keil secondary boot loader for the LPC3180.
- downloading example user code to the external Flash memory using ARM/µVision3 tools

5.1 Loading the Secondary Boot Loader

To better understand the reasons for the need of a secondary boot loader on the LPC3180 a concise explanation of the boot process is presentedbelow.

After a reset the LPC3180 executes the on-chip bootstrap software located in the on-chip boot ROM. This software is responsible for reading code out of NAND Flash and loading it in internal SRAM and executing it. Because the SRAM is limited to 64kB on the LPC3180, it is not possible to execute code out of internal RAM which exceeds 64kB in size. Furthermore, despite the presence of 64kB of IRAM in the LPC3180 the bootstrap software will only copy 15.5kB from the NAND Flash into IRAM for execution.

To get around this limitation code must be executed from external SDRAM which is much larger in size than internal SRAM. In order to execute code out of SDRAM three basic steps must be followed:

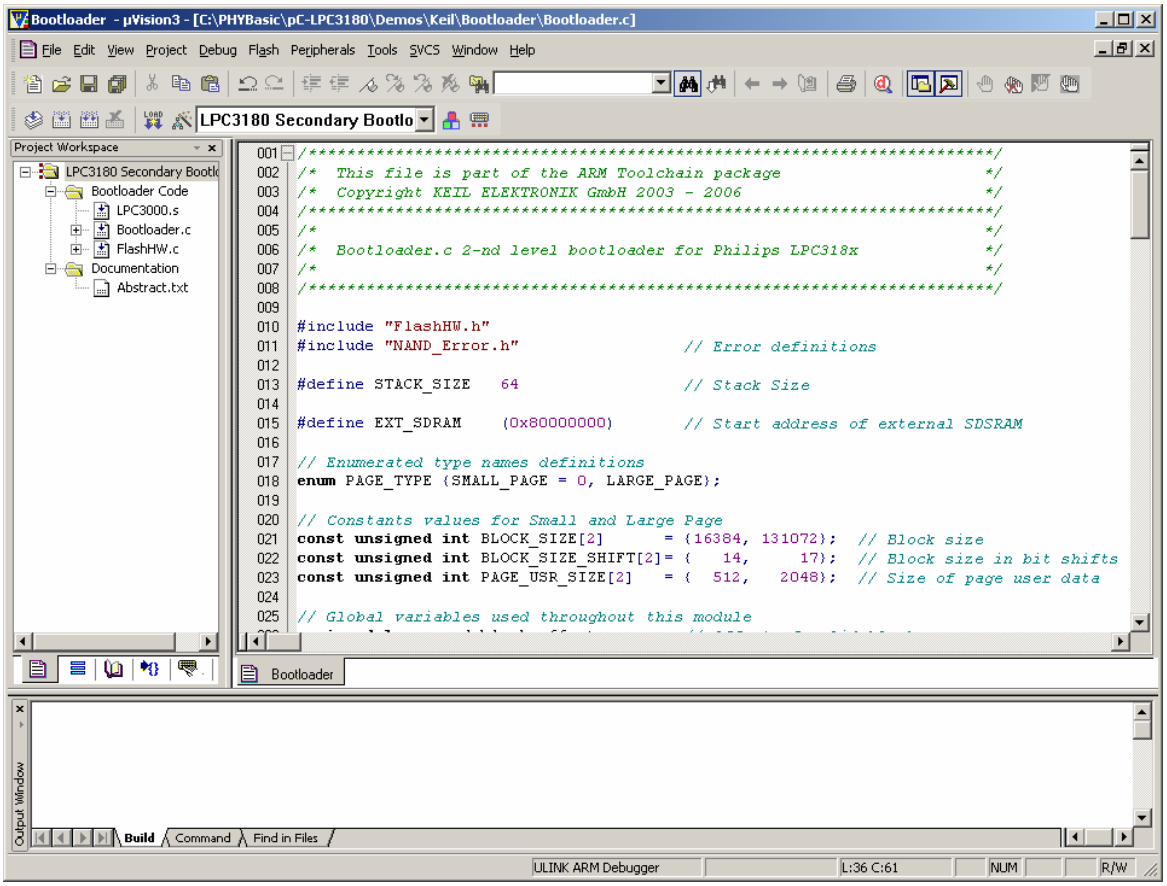
(1) the SDRAM interface must be initialized, and
(2) the code must be copied from NAND Flash into SDRAM, and
(3) execution must be transferred to SDRAM.

The LPC3180 bootstrap software is not capable of initialization SDRAM, copying code from NAND Flash to SDRAM, and executing it. For this reason a secondary boot loader must be present on the LPC3180 target. The secondary boot loader must be located in block 0 of the NAND Flash from which the LPC3180 bootstrap software will load into IRAM and execute. This secondary boot loader will then initialized the SDRAM, copy application code from NAND Flash into SDRAM, and then transfer execution to SDRAM.

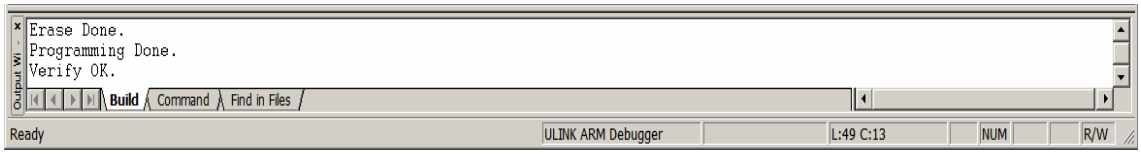
For a more detailed description of the boot process, please refer to the phyCORE-LPC3180 Hardware Manual.

Note: Bad block skipping is supported by the secondary boot loader provided with this Rapid Development Kit.

- Open the **Bootloader** project from the µVision3 menu *Project / Open Project.*
- Browse to ***C:\PHYBasic\pC-LPC3180\Demos\Keil\Bootloader***.



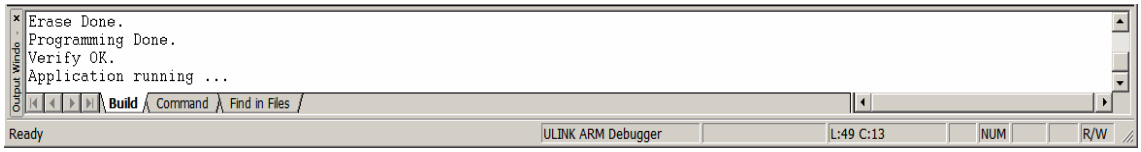
- Download the code into NAND Flash memory by either selecting the *Download to Flash Memory* icon  on the build toolbar or in the main menu bar select *Flash / Download.*
- The individual steps of the Flash download procedure can be viewed at the bottom of the µVision3 **Output Window - Build** tab.
- 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.



5.2 Loading the Blinky Executable to Flash

- Open the **Blinky** project as described in *section 3.3*.
- In the **Select Target** pull-down menu be sure that the *LPC3180 Ext Memory* 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 bar select *Flash / Download.*
- The individual steps of the Flash download procedure can be viewed at the bottom of the µVision3 **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 LEDs (D400, D401, D402 and D403) on the Carrier Board.

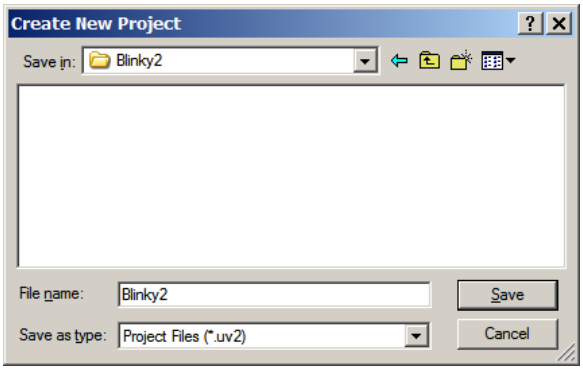
6 Getting More Involved

What you will learn with this example:

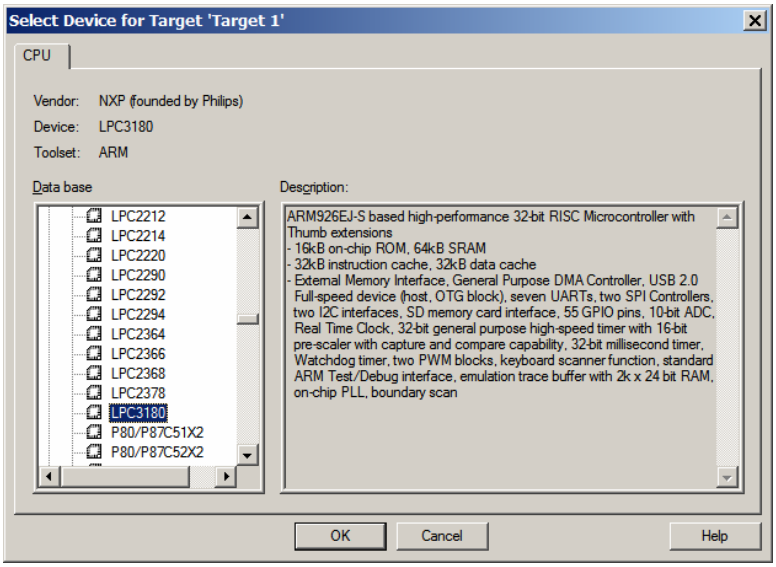
- how to create a new project,
- how to configure the µVision3 IDE (Integrated Development Environment).

6.1 Creating a New Project

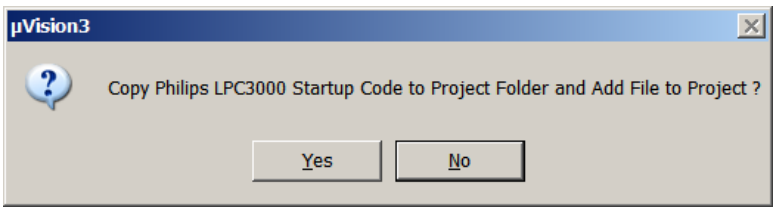
- Start the Keil µVision3 environment.
- Open the **Project** menu and create a **New Project** called ***Blinky2.uv2*** within the existing project directory ***C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky2*** (default location) on your hard drive.



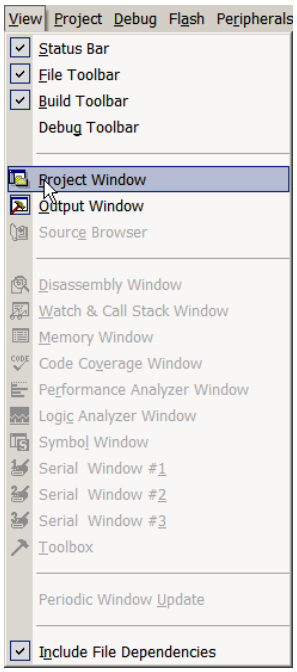
- Select the *NXP LPC3180* in the CPU vendor database list and select *OK*.



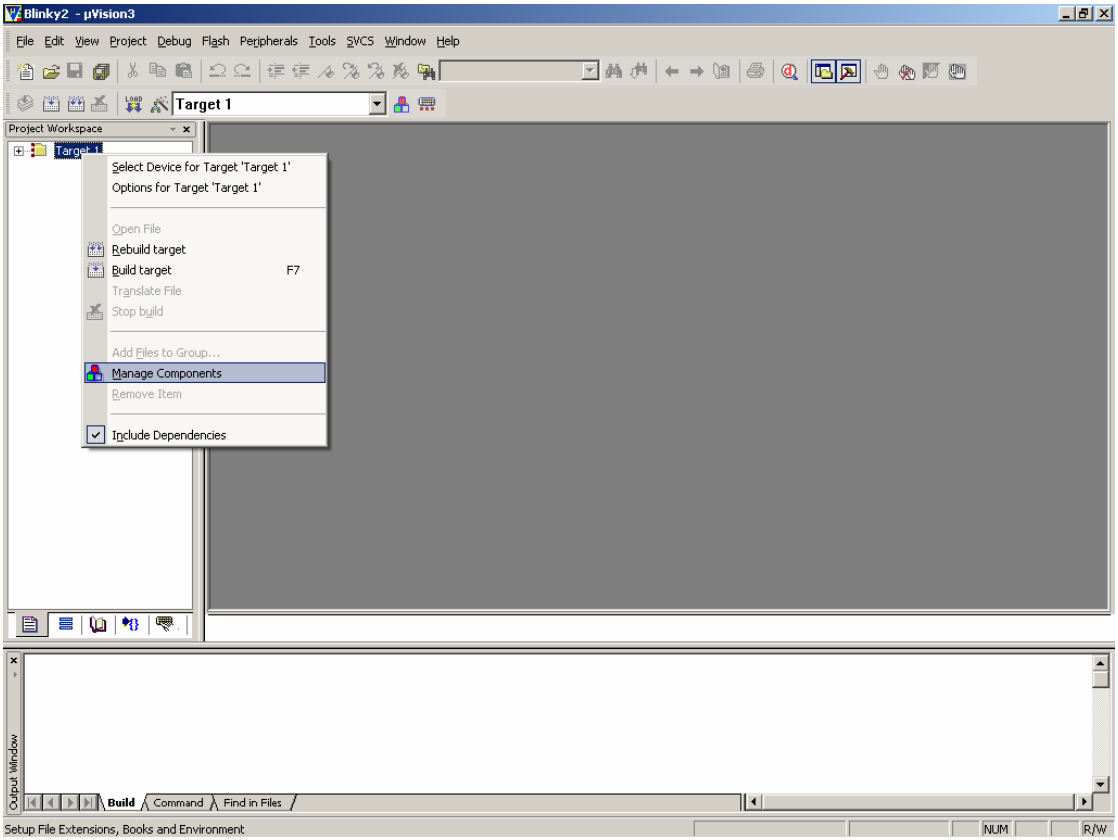
- When the prompt to "Copy Philips LPC3000 Startup Code to the Project Folder and Add File to Project?" appears select *No.*



- Open the *Project Window* view by selecting ***View / Project Window***

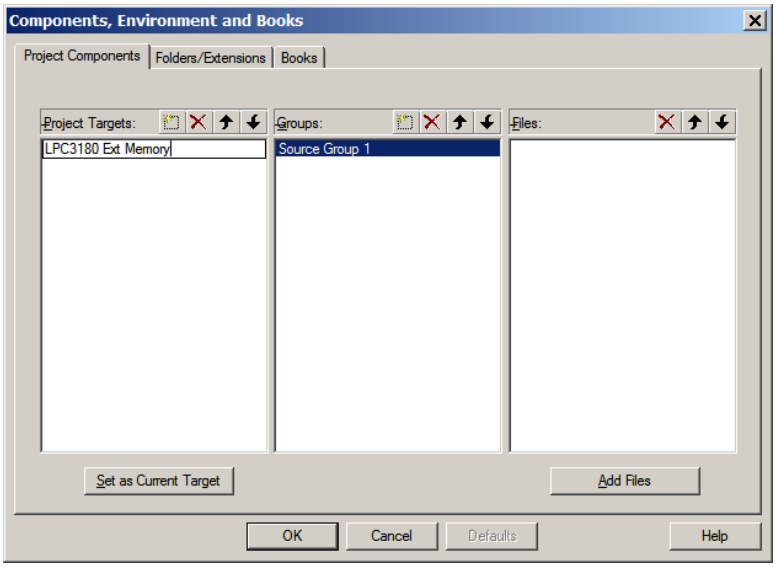


- To configure the target click on the  icon in the build toolbar or right-click on the target, *'Target 1'* in the **Project Workspace** window and select *Manage Components*.

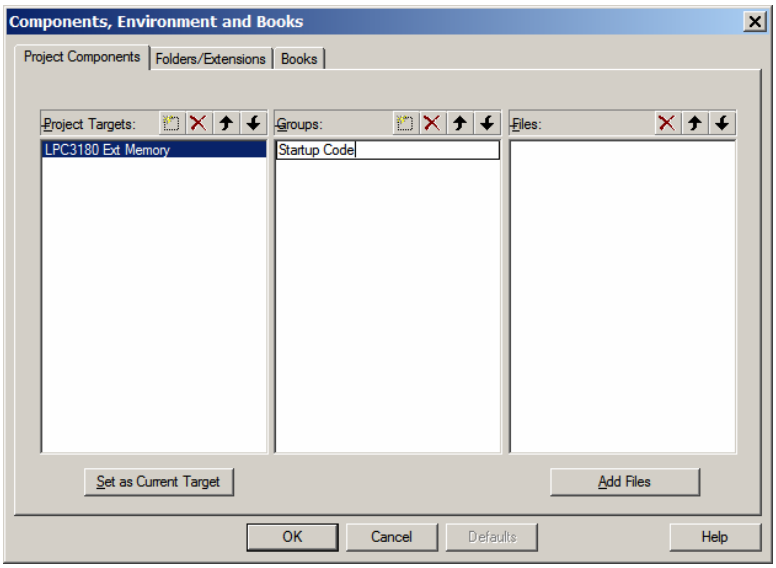


- The **Components, Environments, and Books** window will appear.

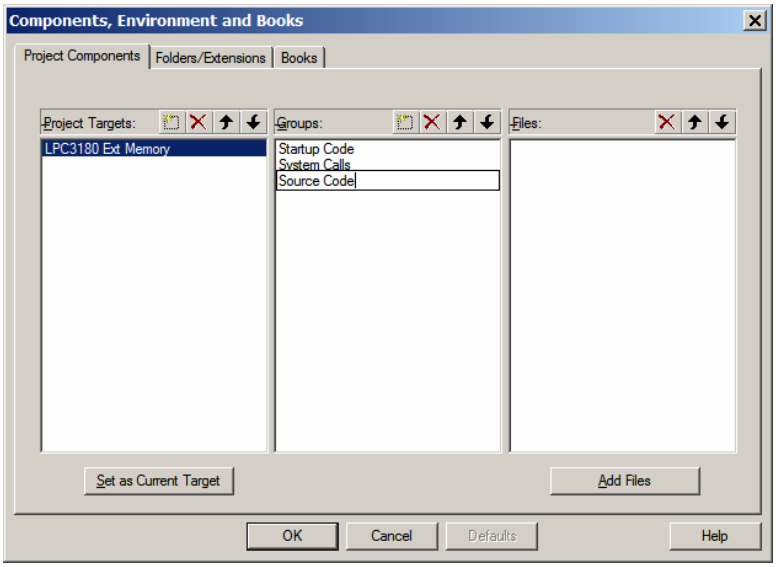
- In the **Project Components** tab, double-click on *Target 1*. Change the name of the target to *LPC3180 Ext Memory.*



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

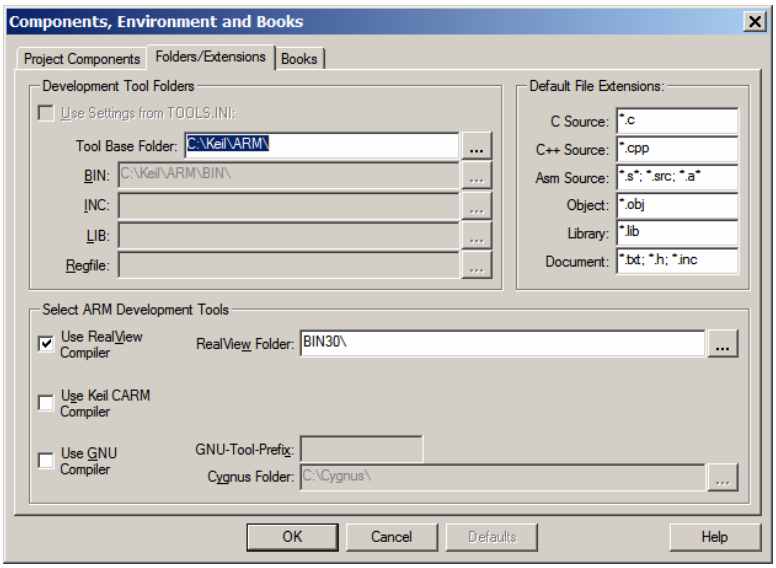


- Add more groups by selecting the *New(Insert)* icon  in the **Groups** window. Name the new groups *System Calls* and *Source Code.*

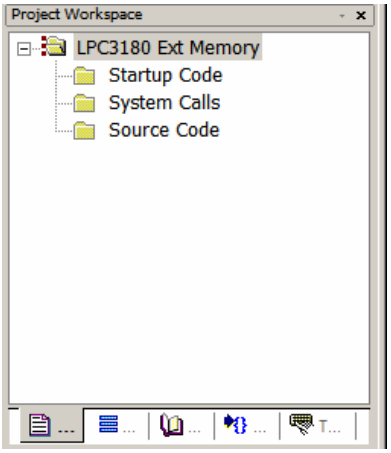


- Select the **Folders/Extensions** tab.

- Be sure that *Use RealView Compiler* is selected under Select ARM Development Tools.

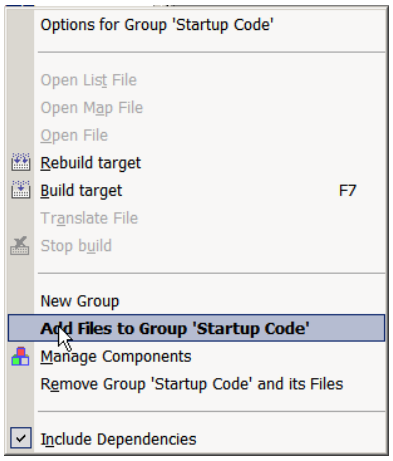


- Click *OK*. This brings you back to the **Project Workspace** window.
- You are now ready to add source files to the project. Make sure the view in the **Project Workspace** window is expanded to see the file groups.

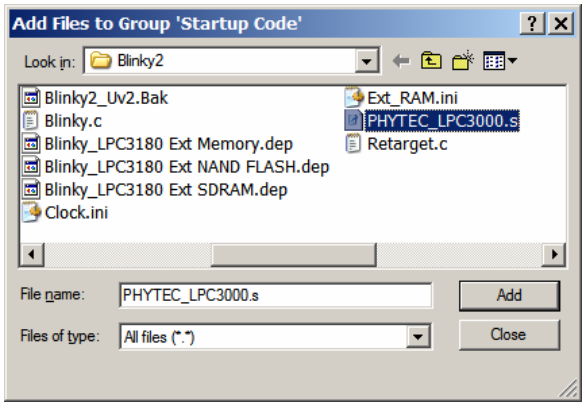


6.2 Adding Source Files to the Project

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



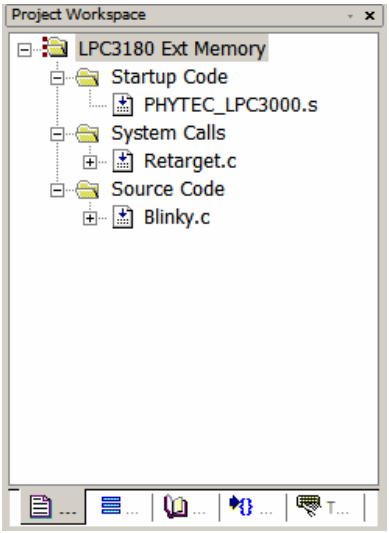
- In the *File of type* pull-down menu, select: "***Asm Source file (*.s*; *.src*; .a*)***". Browse to ***C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky2*** and select ***PHYTEC_LPC3000.s***.
- Click on the *Add* button and then click *Close*.



Note: Always use the ***PHYTEC_LPC3000.s*** file provided by PHYTEC in your application project if you are using the Keil Evaluation Version. The default Keil ***LPC3000.s*** startup file uses scatter loading functionality which is disabled in the Evaluation version of the Keil tool chain. Using other startup code, e.g. the default Keil startup code that is offered when creating a new project will lead to compiler errors.

- In the **Project Workspace** window right-click on the *System Calls* group.
- Select *Add Files to Group 'System Calls'.*
- Browse to ***C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky2*** and select ***Retarget.c***.
- 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-LPC3180\Demos\Keil\Blinky2*** and select ***Blinky.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 C source and system files called ***Blinky.c*** and ***Retarget.c*** and an existing assembly source file called ***PHYTEC_LPC3000.s***. The next step is to configure the *Project Options* for the Target *'LPC3180 Ext Memory'*.

6.3 Setting Options for Target

Configure the necessary target options by selecting the *Options for Target* icon  on the build toolbar or right-click on the *LPC3180 Ext Memory* target in the *Project Workspace* window and select *Options for the Target 'LPC3180 Ext Memory'*.

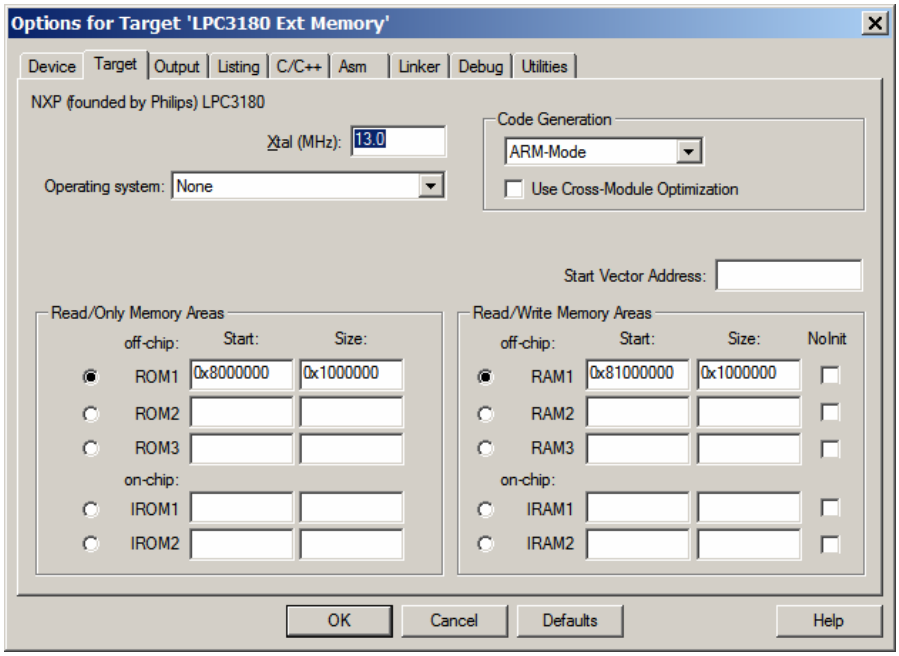
6.3.1 Configure the Target Options

- In the **Target** tab be sure that *Xtal* is set to **13 MHz,** and *Read/Only Memory Areas off-chip* ***ROM1*** *and* Read/Write Memory Areas *off-chip* ***RAM1*** *are checked.* Set the Memory Areas *Start* and *Size* as follows:

  ROM1: **0x8000000** (Start) **0x1000000** (Size, 16 MB)

  RAM1: **0x81000000** (Start) **0x1000000** (Size, 16 MB)

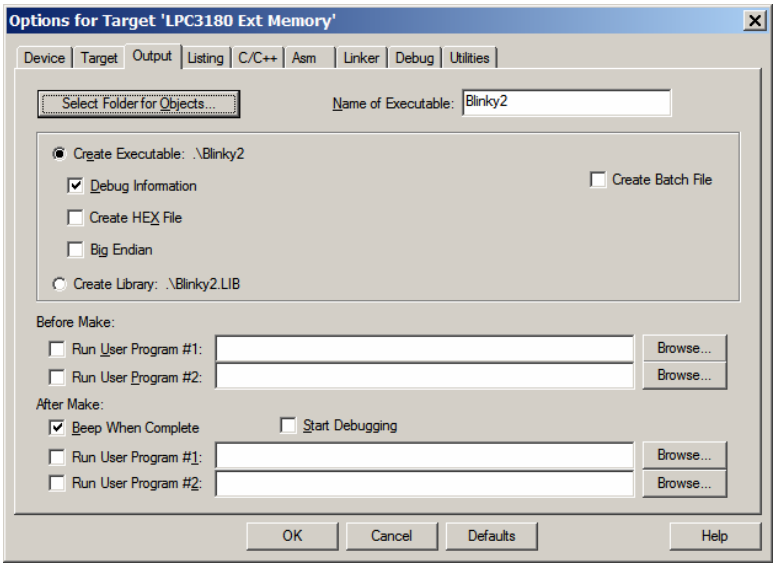
  The **Target** tab should appear as follows:



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 SDRAM. The phyCORE-LPC3180 standard version features 32 MByte of external SDRAM.

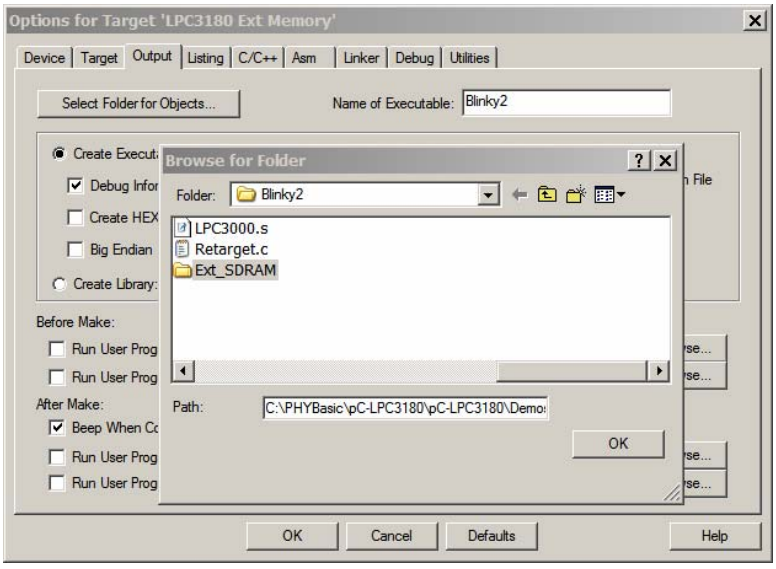
6.3.2 Configure the Output Options

- 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 and brows to. ***C:\PHYBasic\pC-LPC3180\pC-LPC3180\Demos\Keil\Blinky2***.

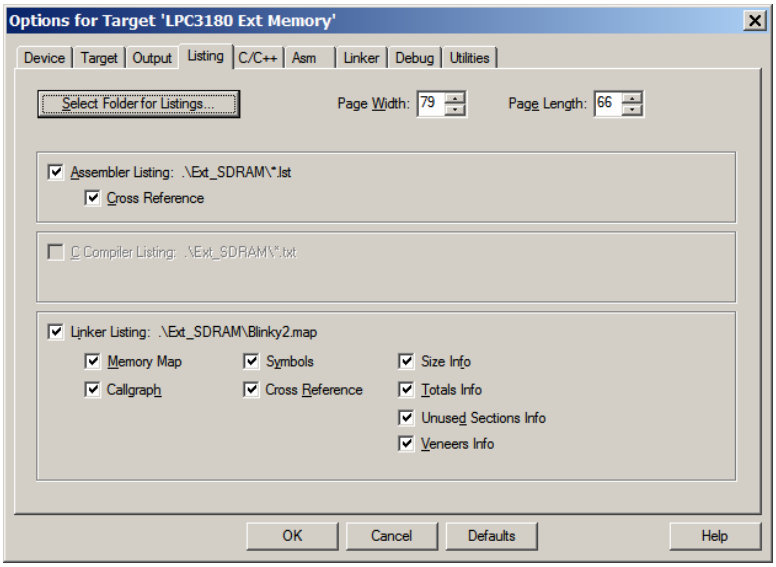
- Click on the *Create a new folder*  button and name the folder ***Ext_SDRAM*****,** double-click the ***Ext_SDRAM*** folder to open it and select *OK*.



6.3.3 Configure the Listing Options

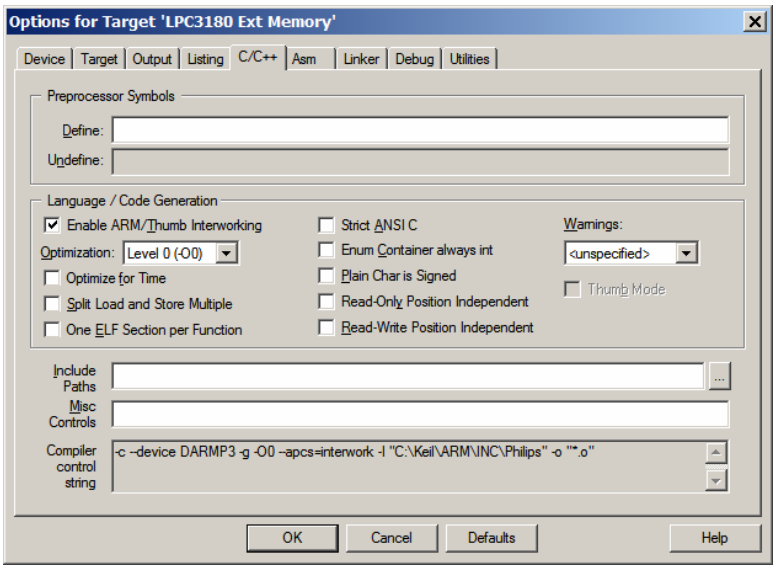
- In the **Listing** tab, leave the default settings.
- Click on the *Select Folder for Listings* button.
- Browse to the folder:
  C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky2\Ext_SDRAM.
- Click *OK*.

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



6.3.4 Configure the C/C++ Options

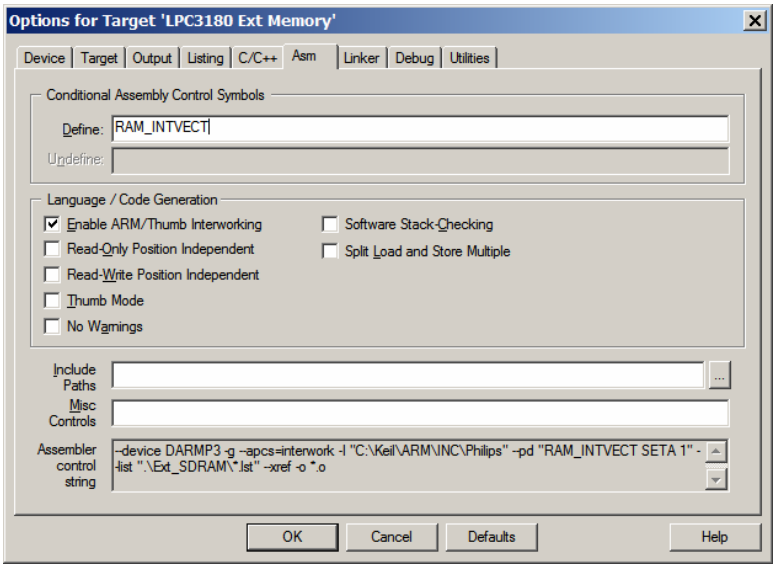
- In the **C/C++** tab, leave the default settings. The **C/C++** tab should appear as follows:



6.3.5 Configure the Asm Options

- Change to the **Asm** tab. In the *Conditional assembly control Symbols, Define* field, type: **RAM_INTVECT**. When set, the startup code copies exception vectors from on-chip ROM to on-chip RAM.

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



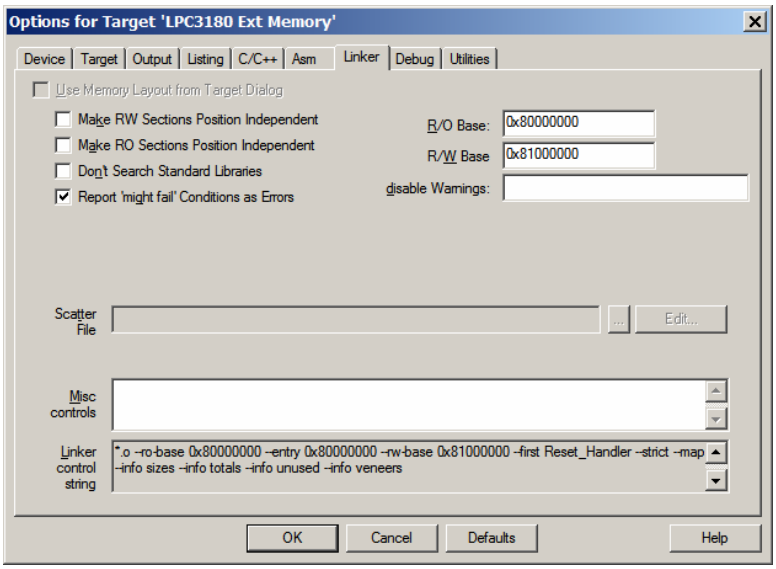
6.3.6 Configure the Linker Options

- In the **Linker** tab, set

  R/O Base: 0x80000000 and

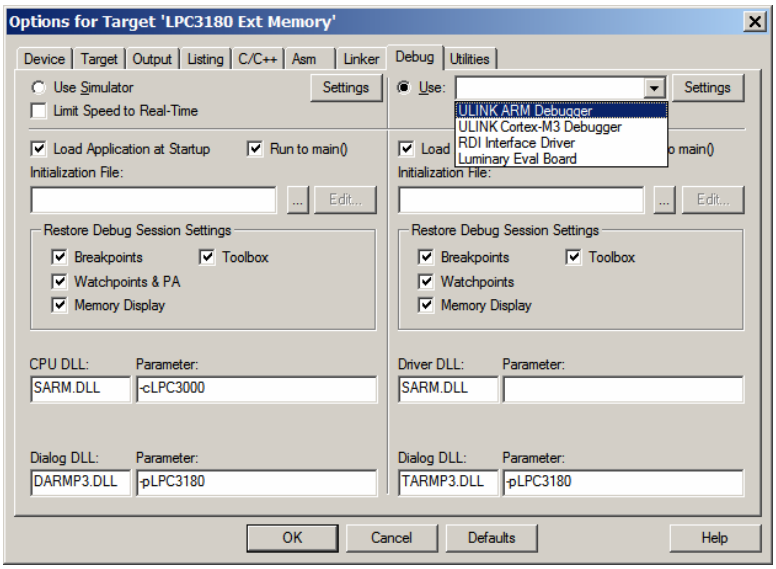
  R/W Base: 0x81000000

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

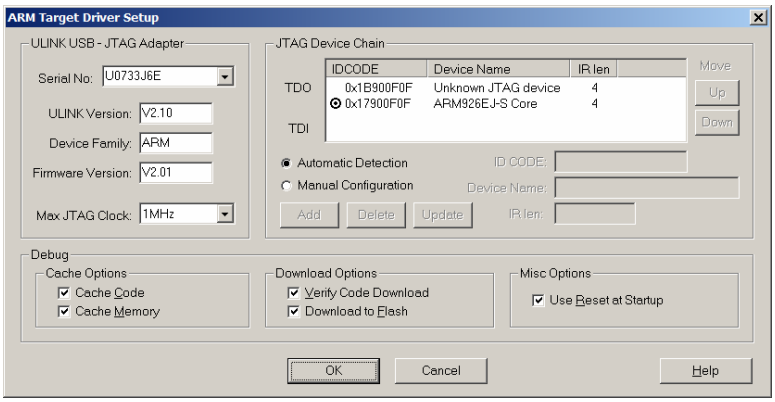


6.3.7 Configure the Debug Options

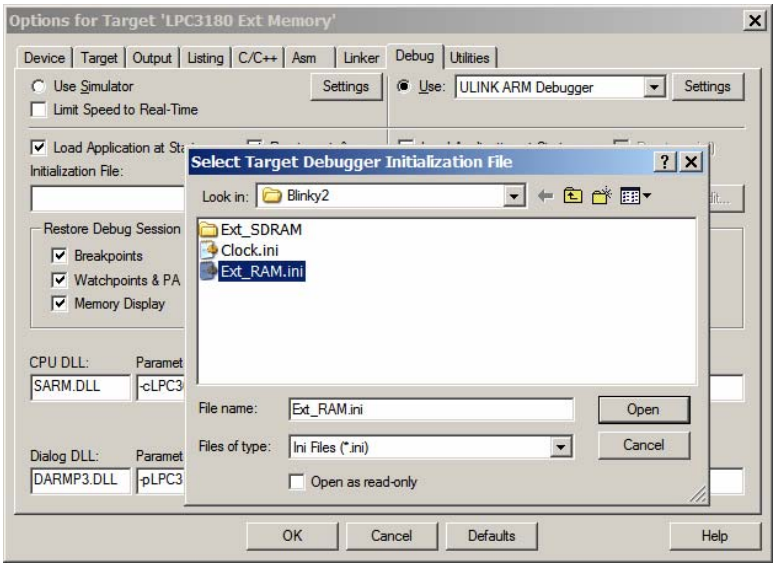
- In the **Debug** tab, check the *Use: ULINK ARM Debugger* option.



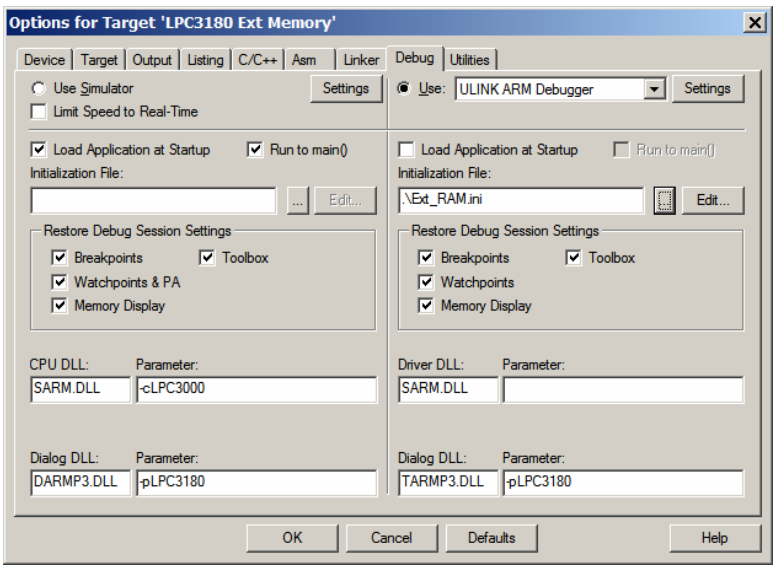
- Configure the *ARM Target Driver Setup* by clicking the *Settings* button. The *ARM Target Driver Setup* settings should be as follows with all *Debug* options checked (Cache Options, Download Options and Misc Option):



- 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 ***Ext_RAM.ini*** file *(see below)*.
- Add the correct *Initialization File* by clicking the browse button  and select ***Ext_RAM.ini*** from the **Blinky2** project folder ***C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky2***.

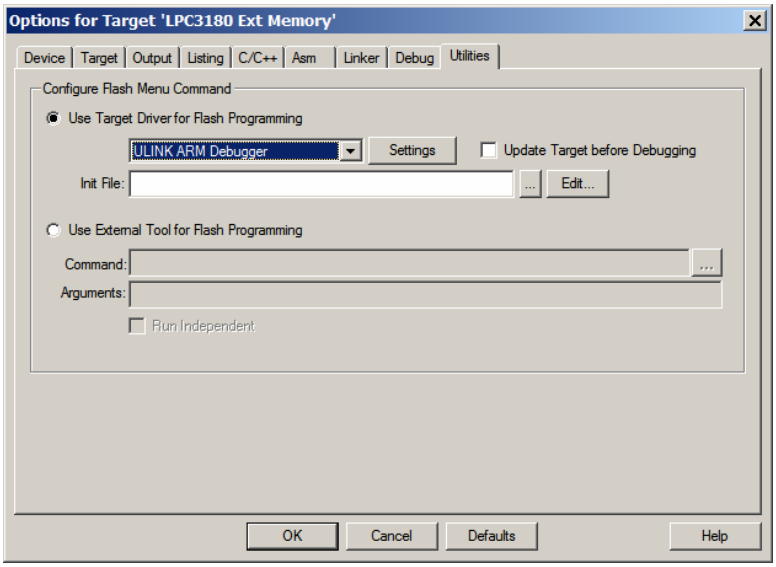


- The ***Debug*** tab should now appear as follows:

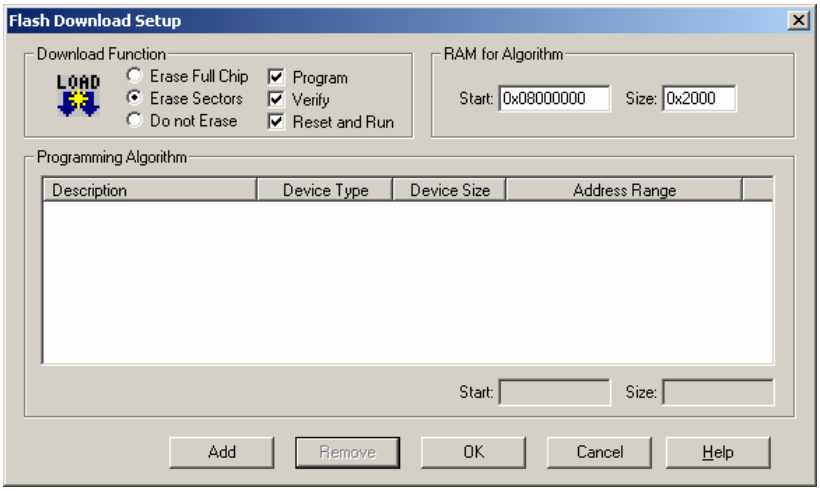


6.3.8 Configure the Utilities Options

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

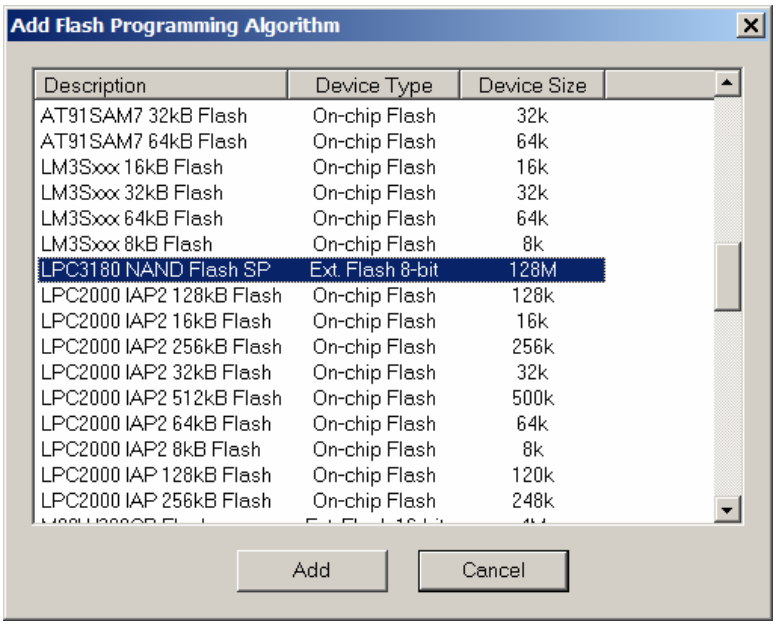


- Click the *Settings* button.
- In the **Flash Download Setup** window select*: Erase Sectors, Program Verify* as well as *Reset and Run* in the *Download Function* section.

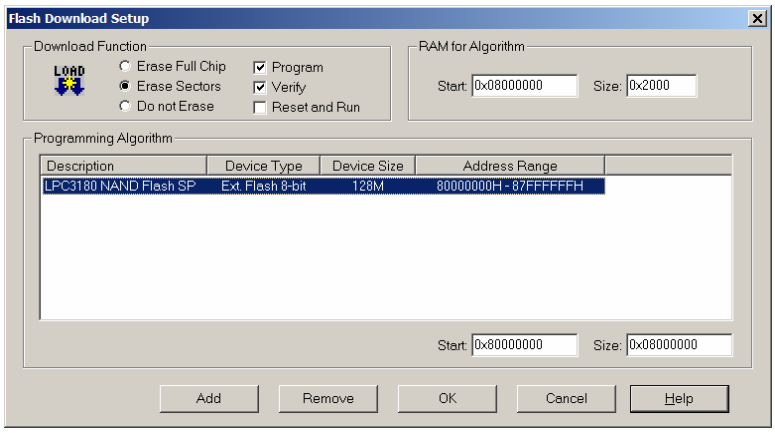


- Click on *Add* to add a *Programming Algorithm*.

- In the **Add Programming Algorithm** window, select *LPC3180 NAND Flash SP* and click *Add.* The NAND Flash device is the external Flash memory populating the phyCORE-LPC3180 module.



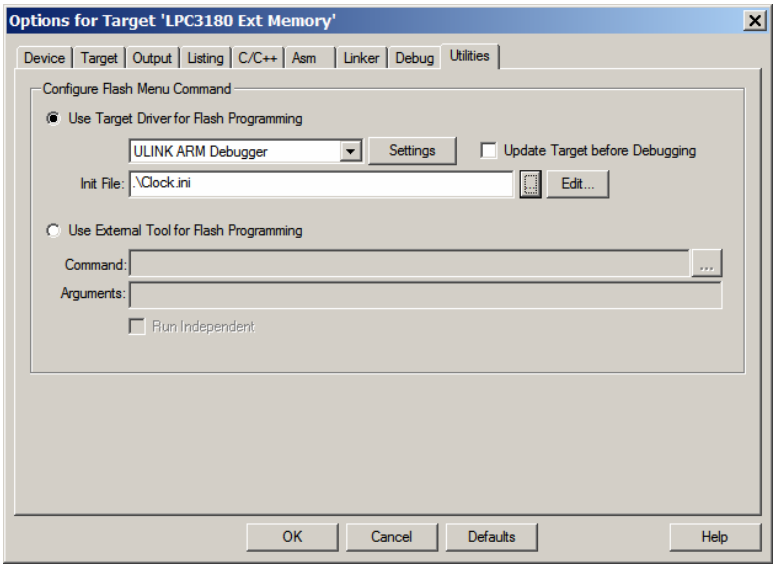
- Click on *Add***.**
- In the *RAM for Algorithm* fields, set the *Start* address to **0x08000000** and address *Size* to **0x2000** as shown below.
- In the *Programming Algorithm* window click on the *LPC3180 NAND Flash SP* device and set the *Start* address to **0x80000000** and *Size* to **0x08000000** as shown below and click *OK*.



- Back in the **Utilities** tab, select the browse button  in the *Init File* line.
- In the **Select Flash Initialization File** pop-up window browse to

  C:\PHYBasic\pC-LPC3180\Demos\Keil\Blinky2 and select the ***Clock.ini*** file.
- Click the *Open* button.

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



- Click *OK*.
- In the main µVision3 menu select *File* / *Save All.*

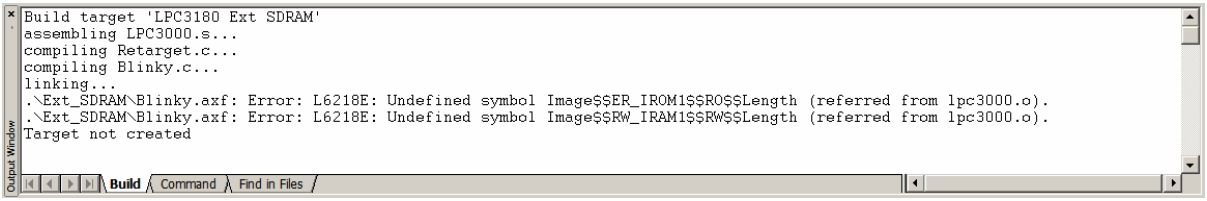
6.4 Building the Project

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, save the file and repeat the build.

If there are linking errors as shown below, check that the ***PHYTEC_LPC3000.s*** startup code was included in the project as described in *section 6.2*.



If there are no errors, the code is ready to be downloaded into the phyCORE-LPC3180 external SDRAM for debugging and to the phyCORE-LPC3180 external NAND Flash memory. Refer to *section 5, Flash Programming* for Flash download details.

Note: Always use the ***PHYTEC_LPC3000.s*** file provided by PHYTEC in your application project when working with the Keil evaluation version. Keep in mind that this version is limited in code size. The default Keil ***LPC3000.s*** startup file uses scatter loading functionality, which is disabled in the evaluation version of the Keil tool chain. Using the default Keil startup code, which is offered when creating a new project, will lead to linker errors.

The scatter loading functions in the Keil tools provide the size of the compiled application binary. The secondary boot loader uses this size information to determine how much code to load into SDRAM for execution. The ***PHYTEC_LPC3000.s*** startup file has the binary code size hard coded to 16KB, which is the maximum size for the Keil evaluation tools, rather than using a dynamic size determination via the scatter loading function.

For projects built with the full version of the Keil ARM Development Tools, please use the default Keil startup code for the LPC3180. This startup code will determine the binary file size based on actual values rather than using the fixed 16KB size.

Document:	**phyCORE-LPC3180 QuickStart Instructions - KEIL**
Document number:	**L-690e_1, November 2006**

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

Document Number: L-690e_1
Printed in Germany