

phyCORE[®]-LPC3250

QuickStart Instructions

**Using Keil's ULINK and the Keil Microcontroller
Development Tool Chain**

Note: The PHYTEC Kit CD includes the electronic version of
the English phyCORE[®]-LPC3250 Hardware Manual

Edition: September 2009

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.

© Copyright 2009 PHYTEC America LLC, Bainbridge Island, WA.

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 America LLC.

	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

3rd Edition: September 2009

1	Introduction to the Rapid Development Kit.....	4
1.1	Rapid Development Kit Documentation	4
1.2	Overview of this QuickStart Instruction	4
2	System Description	5
2.1	Hardware Description	5
2.2	Host System Requirements.....	5
2.3	The PHYTEC phyCORE®-LPC3250	6
2.4	Keil RealView Microcontroller Development Kit	6
3	Getting Started	7
3.1	Installing Rapid Development Kit Software	7
3.1.1	Installing Documentation from PHYTEC Kit CD	7
3.1.2	Installing Keil Microcontroller Development Tools.....	8
3.2	Interfacing the phyCORE®-LPC3250 to a Host-PC.....	8
3.3	Downloading Blinky Example Code with μ Vision3	9
3.3.1	Build the Project.....	11
3.3.2	Start the Debugger and Download to Internal RAM.....	11
4	Debugging	13
4.1	Keil μ Vision3 Debug Features.....	13
4.2	Debugging Modes	15
4.3	Starting the Debugger	16
5	NAND Flash Programming.....	17
5.1	Loading the Secondary Bootloader for NAND Flash.....	17
5.2	Loading the Blinky Application to NAND Flash	19
6	NOR Flash Programming	20
6.1	Loading the Secondary Bootloader for NOR Flash.....	20
6.2	Loading the Blinky Application to NOR Flash.....	22
7	Getting More Involved	23
7.1	Creating a New Project.....	23

1 Introduction to the Rapid Development Kit

The phyCORE® System on Module (SOM) is designed to be plugged into a PHYTEC Carrier Board. The PHYTEC Carrier Board contains the I/O connectors as well as any other interface circuitry not provided on the phyCORE module itself. The phyCORE SOM, 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 a system on module mounted on an applicable Carrier Board, cables, power supply, printed schematics, applicable evaluation software development tool CDs, and the PHYTEC Kit CD. The PHYTEC Kit 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 Kit CD:

- the [phyCORE®-LPC3250 Hardware Manual](#)
- controller User's Manuals and Data Sheets
- this QuickStart Instruction

1.2 Overview of this QuickStart Instruction

This QuickStart Instruction provides a general kit description, as well as software installation hints and example programs. It is structured as follows:

1. The "*Getting Started*" section describes how to interface the phyCORE-LPC3250 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 Keil RealView Microcontroller Development Tools.
2. The "*Debugging*" section introduces the main debugging features using a new target within the **Blinky** example project and uses the **Blinky** example project to demonstrate the download of user code to the external SDRAM using the Keil ULINK and Keil Microcontroller Development Tools.
3. The "*NAND Flash Programming*" section shows how to download a secondary boot loader as well as the **Blinky** executable into the external NAND Flash.
4. The "*NOR Flash Programming*" section shows how to download a secondary boot loader as well as the **Blinky** executable into the external NOR Flash.
5. The "*Getting More Involved*" section provides instructions for creating a new project with the **Blinky** project as a template.

2 System Description

2.1 Hardware Description

The following PHYTEC hardware components are included in the phyCORE-LPC3250 Keil Rapid Development Kit (part number KPCM-040-Keil) and are necessary for completing the instructions in this QuickStart instruction:

- the PHYTEC phyCORE®-LPC3250 System on Module (PCM-040)
- the phyCORE®-LPC3250 Carrier Board (GPCM-967)
- Bare PCB Expansion Board (PCM-988)
- AC adapter supplying 5 VDC /3.2A adapter, center positive
- Straight Ethernet cable
- Serial cable
- USB Standard A to mini-B cable
- System on Module extraction tool
- Hard copy schematics
- Keil ULINK JTAG-USB adapter¹
- PHYTEC Kit CD
- Keil Microcontroller Development Tools Evaluation CD

2.2 Host System Requirements

- This QuickStart guide will require the installation of Keil ARM/μVision3 software development tool chain
- Operating System: Windows 2000, Windows XP or Windows Vista
- CPU architecture: Any x86 32-bit or 64-bit (x64: AMD64 or Intel EM64T) processor
- 60 MB Free Hard Disk Space
- 128 MB of RAM

¹ : The Keil ULINK is included in the Rapid Development Kit version with the part number KPCM-040-KEIL.

2.3 The PHYTEC phyCORE®-LPC3250

The phyCORE®-LPC3250 SOM is an ARM-9 based, small form factor, OEMable module populated with the NXP LPC3250. State-of-the-art power management, Vector Floating Point Unit (VFP), and rich peripherals such as USB OTG, Ethernet, and integrated LCD controller make this module the ideal candidate for embedded applications requiring high performance and low power consumption. The on-board MMU supports major operating systems, including Linux and Windows Embedded CE. Other chip-level features include 7 UARTs, SPI, I2C, a real-time clock with a separate power domain, and NAND Flash and DDR memory controllers. These features make the devices particularly suitable for automotive and industrial control applications as well as medical systems.

Please refer to the [phyCORE®-LPC3250 Hardware Manual](#) for specific information on board-level features, jumper configuration, memory mapping, pin layout, and Carrier Board features.

2.4 Keil RealView Microcontroller Development Kit

The Keil RealView Microcontroller Development Tools for ARM7, ARM9, and Cortex-M3 microcontrollers are easy to learn and use, yet powerful enough for the most demanding embedded applications. The RealView C/C++ compiler supports all ARM-compatible devices including the NXP LPC3250 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. For more information on Keil ARM/µVision3 tools visit their website at:

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

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

PHYTEC

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

**KEIL
SOFTWARE**

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

3 Getting Started

What you will learn with this Getting Started example:

- installing Rapid Development Kit software
- interfacing the phyCORE®-LPC3250, mounted on the Carrier Board, to a host-PC using the Keil ULINK
- downloading example user code to the LPC3250 internal SRAM.

3.1 Installing Rapid Development Kit Software

3.1.1 Installing Documentation from PHYTEC Kit CD

The PHYTEC Kit CD installs documentation, datasheets, and demo files for the phyCORE-LPC3250. When you insert the PHYTEC Kit CD into the CD-ROM drive of your host-PC, the CD should automatically launch a setup program. Otherwise the setup program (setup.exe) can be manually executed from the root of the CD.

1. Insert the PHYTEC Kit CD.
2. On the *Welcome to the phyCORE-LPC3250 Rapid Development Kit Setup Wizard* window, select **Next** to continue.
3. Keep **C:\PHYTEC** as the installation path and select **Next** to continue.
4. On the *Ready to Install* window, select **Install**.
5. After the installation process is complete, select **Finish**.

The PHYTEC Kit CD installation path is **C:\PHYTEC\phyCORE-LPC3250** and the following five subfolders are located within the **phyCORE-LPC3250** directory:

Demos – This folder contains third party demos.

Documentation – This folder contains datasheets, hardware manual, QuickStarts, and other product related documentation.

Linux – This folder contains the Linux BSP and tools.

Stage1 Loader – This folder contains the Stage1 Bootloader for the LPC3250.

WinCE – This folder contains a binary Eboot Bootloader, Windows Embedded CE images, demo application for Visual Studio, and SDK.

3.1.2 Installing Keil Microcontroller Development Tools

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

- Install the Keil ARM Evaluation tools, from the enclosed Keil Microcontroller Development Tools CD, following the steps indicated in the install procedure.
- Alternately, the Keil ARM Evaluation tools can be installed from:

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

3.2 Interfacing the phyCORE®-LPC3250 to a Host-PC

Connecting the phyCORE®-LPC3250, mounted on the phyCORE® Carrier Board, to your computer is simple.

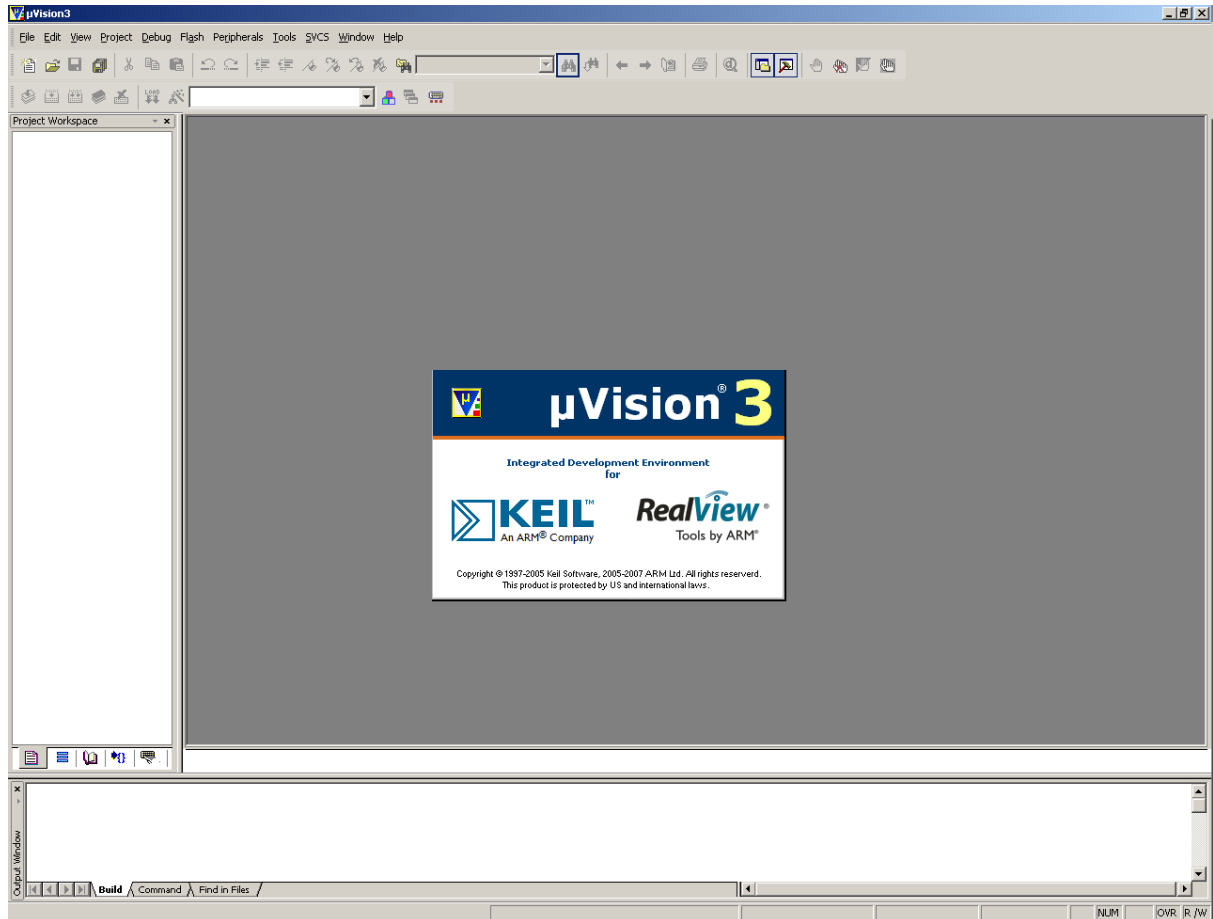
- Ensure proper jumper settings on the phyCORE® Carrier Board as indicated in the phyCORE®-LPC3250 Hardware Manual.
- Connect the 2.54 mm pitch ribbon cable from the Keil ULINK JTAG adapter to X12 on the phyCORE® Carrier Board.
- Connect the USB end of the ULINK JTAG adapter to the USB port of your host-PC using the included USB cable.
- Connect the included 5V power adapter to the power socket X10 on the phyCORE® Carrier Board.
- The red power LED D20, located next to the power socket at X10, should light. This indicates that proper voltage is supplied to the phyCORE® module and Carrier Board.
- The phyCORE® System on Module and Carrier Board should now be properly connected to a host PC via the Keil ULINK. You are now ready to use the Keil Development Tools to establish communication between the host-PC and target hardware.

3.3 Downloading Blinky Example Code with μ Vision3

The μ Vision3 evaluation software development tool chain should have been installed as described in section 3.1.2

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

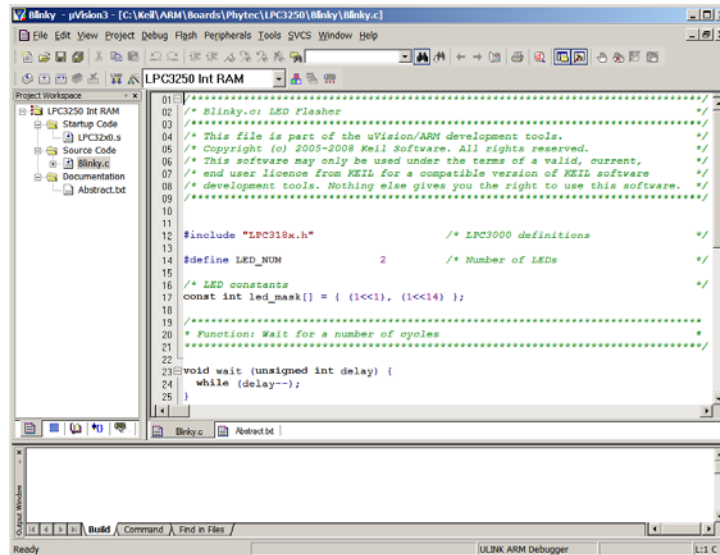
After you start μ Vision3, the window shown below appears.




From this window you can create projects, edit files, configure tools, assemble, link and start the debugger.

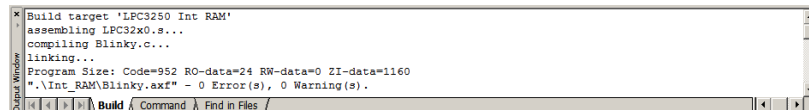
Close all projects that might be open by selecting *Project \ Close Project*.

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.




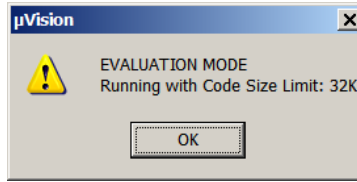
3.3.1 Build the Project

- 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

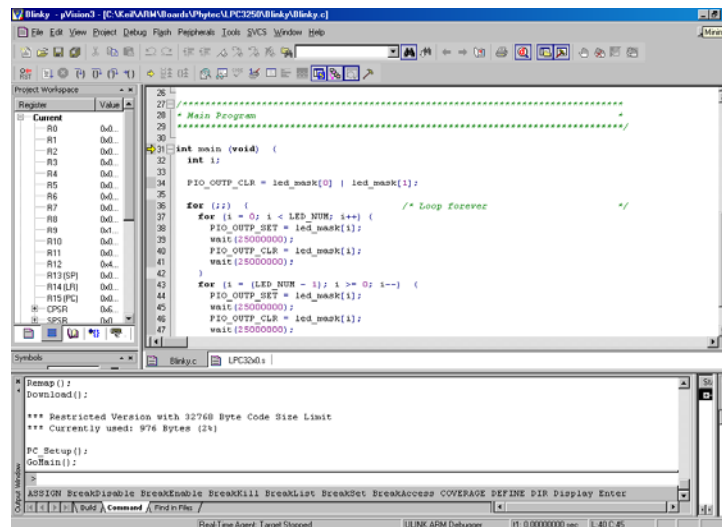
- To start the μ Vision3 debug environment, click on the debugger icon  on the μ Vision3 toolbar or select from the tool menu under Debug.
- While using the μ Vision3 Evaluation Tools, the following warning will appear. Please select **OK** and continue.



- 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 run.

Successful execution of the program will alternately flash LED1 and LED2 (D15 and D18) on the Carrier Board.

4 Debugging

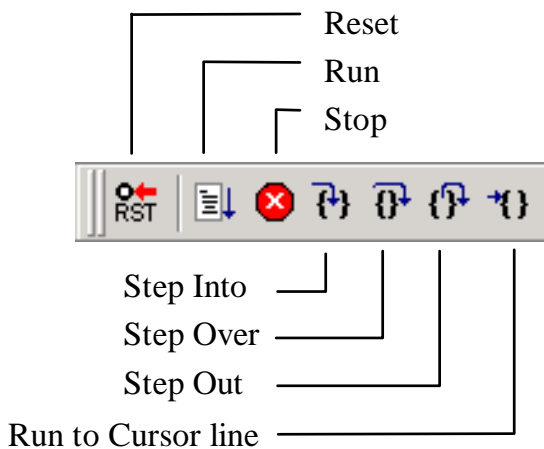
What you will learn with this example:




- familiarizing yourself with simple debug functions provided by the Keil μ Vision3 debug environment
- downloading example user code in the phyCORE-LPC3250 external SDRAM memory





This Debugging section provides a basic introduction to the debug functions included in the Keil ARM/ μ Vision3 evaluation tool chain. For a more detailed description of the debugging features, please refer to the appropriate manuals provided by Keil.

4.1 Keil μ 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*.



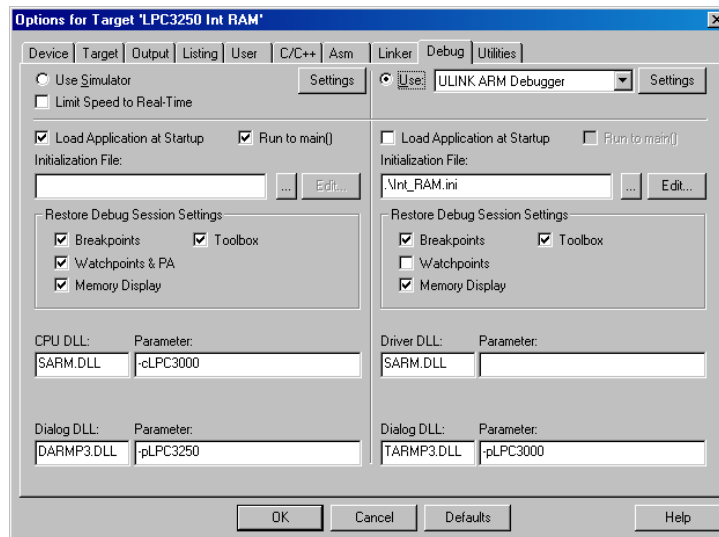
- The first button on the debugger toolbar is the *Reset*  button. The *Reset* command sets the program counter to 0.
- The button to the right of the Reset button starts the *Run*  command. Clicking this button runs the program without active debug functions. To stop program execution at a desired point, 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.

4.2 Debugging Modes

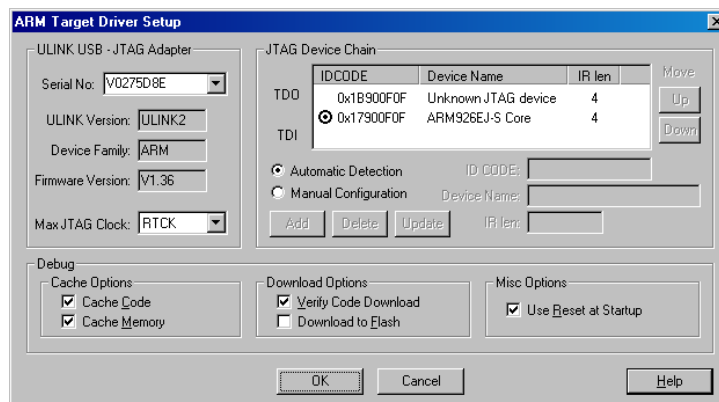
The μ Vision3 Debugger offers two operating modes that can be selected in the **Project / Options for Target phyCORE-LPC3250 / Debug** dialog:

- The **Simulator** allows PC-based simulation of most features of the LPC3250 microcontroller without actually having target hardware. You can test and debug your embedded application before the hardware is ready. μ 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.



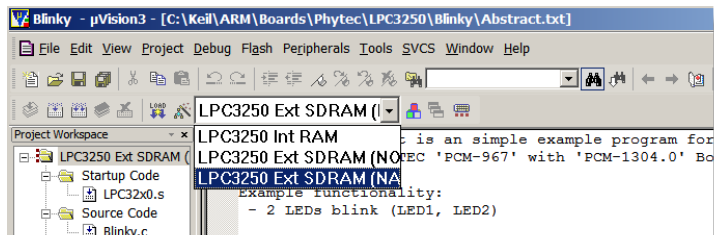
The **Blinky** demo in this section utilizes the **ULINK ARM Debugger** environment.



- To view the **ULINK ARM Debugger** settings, select the *Settings* button in the Debug tab




4.3 Starting the Debugger

- Open the **Blinky** project as described in section 3.3
- In the **Select Target** pull-down menu select the *LPC3250 Ext SDRAM (NAND)* target.



- To start the ARM/μVision3 debug environment, click on the debugger icon  on the μVision3 toolbar.
- Click on the *Run*  icon and the program will run.

Successful execution of the program will alternately flash LED1 and LED2 (D15 and D18) on the Carrier Board.

Click on the debugger icon  on the μVision3 toolbar or select from the tool menu under Debug lto return to the editor window displaying the **Blinky.c** source file.

- Select *Close Project* from the *Project* pull-down window

5 NAND Flash Programming

What you will learn with this example:

- downloading the Keil secondary bootloader for the LPC3250 to on board NAND Flash
- downloading example user code to the NAND Flash memory using ARM/μVision3 tools

5.1 Loading the Secondary Bootloader for NAND Flash

To better understand the reasons for the need of a secondary bootloader on the LPC3250 a concise explanation of the boot process is presented below.

After a reset the LPC3250 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 LPC3250, it is not possible to execute code out of internal RAM which exceeds 64kB in size. Further, despite the presence of 64kB iRAM in the LPC3250 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 greater in size than internal SRAM. To execute code out of SDRAM three basic steps must be followed:

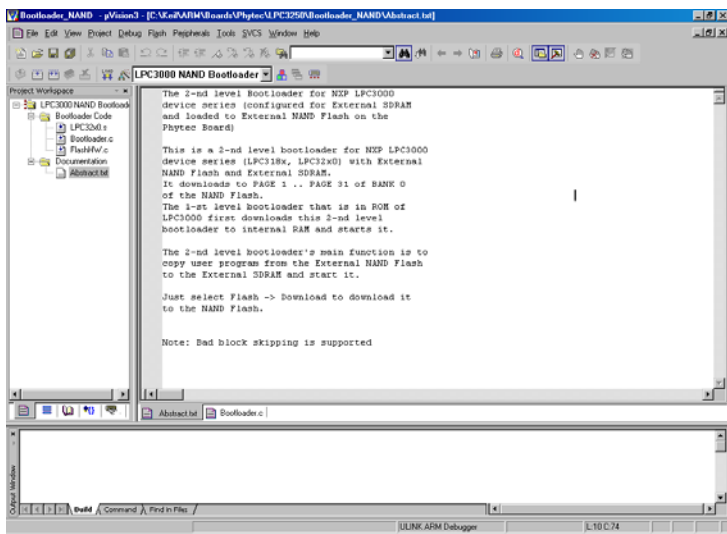
- (1) the SDRAM interface must be initialized, and
- (2) the code must be copied from NAND Flash into SDRAM, and
- (3) execution must be transferred to SDRAM.


The LPC3250 bootstrap software is not capable of initializing SDRAM, copying code into it from NAND Flash, and executing it. For this reason a secondary bootloader must be implemented on the LPC3250. The secondary bootloader is the first thing placed in NAND Flash which the LPC3250 bootstrap software will load into SRAM and execute. This secondary bootloader will then initialize the SDRAM, copy application code from NAND Flash into SDRAM, and then transfer execution to SDRAM.

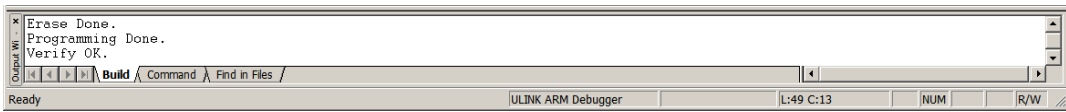
For a more detailed description of the boot process, *please refer to the [phyCORE-LPC3250 Hardware Manual](#)*

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

- Open the **Bootloader_NAND.Uv2** project from the μVision3 menu *Project / Open Project*.
- Browse to **C:\Keil\ARM\Boards\PHYTEC\LPC3250\Bootloader_NAND**



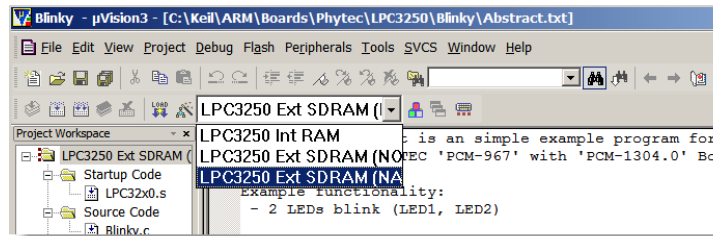
- Download the code into Flash memory by either selecting the *Download to Flash Memory* icon  on the build toolbar or in the main menu bar select *Flash / Download*.
- The individual steps of the Flash download procedure can be viewed at the bottom of the μVision3 **Output Window - Build** tab.
- Wait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




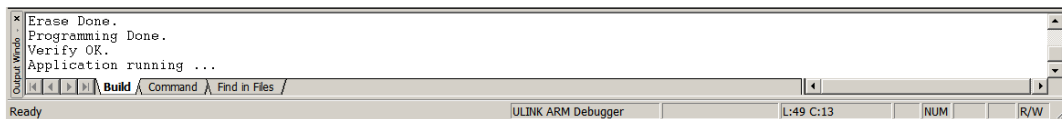
5.2 Loading the Blinky Application to NAND Flash

This section shows how to download the **Blinky** executable into the external NAND Flash

- Open the **Blinky** project as described in section 3.3
- In the **Select Target** pull-down menu be sure that the *LPC3250 Ext SDRAM (NAND)* 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 μ 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 code will execute without further user interaction.



- Push the **reset button** located under CB13 and CB46 on the Carrier Board

Successful execution of the program will alternately flash LED1 and LED2 (D15 and D18) on the Carrier Board.

- Select *Close Project* from the *Project* pull-down window

6 NOR Flash Programming

What you will learn with this example:

- downloading the Keil secondary Bootloader for the LPC3250 to on Board NOR Flash
- downloading example user code to the NOR Flash memory using ARM/μVision3 tools

6.1 Loading the Secondary Bootloader for NOR Flash

To better understand the reasons for the need of a secondary bootloader on the LPC3250 a concise explanation of the boot process is presented below.

After a reset the LPC3250 executes the on-chip bootstrap software located in the on-chip boot ROM. This software is responsible for reading code out of NOR Flash and loading it in internal SRAM and executing it. Because the SRAM is limited to 64kB on the LPC3250, it is not possible to execute code out of internal RAM which exceeds 64kB in size. Further, despite the presence of 64kB iRAM in the LPC3250 the bootstrap software will only copy 15.5kB from the NOR Flash into iRAM for execution.

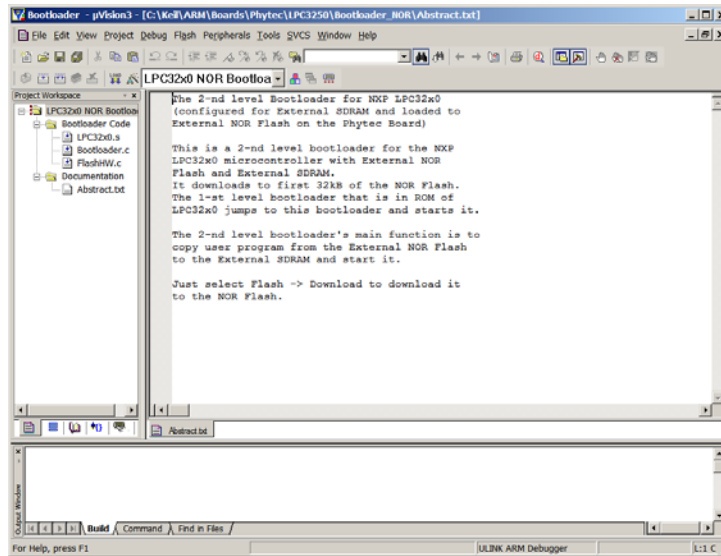
To get around this limitation code must be executed from external SDRAM which is much greater in size than internal SRAM. To execute code out of SDRAM three basic steps must be followed:


- (1) the SDRAM interface must be initialized, and
- (2) the code must be copied from NOR Flash into SDRAM, and
- (3) execution must be transferred to SDRAM.

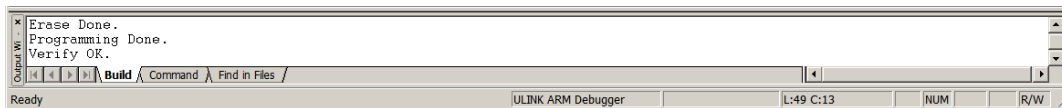
The LPC3250 bootstrap software is not capable of initializing SDRAM, copying code into it from NOR Flash, and executing it. For this reason a secondary bootloader must be implemented on the LPC3250. The secondary bootloader is the first thing placed in NOR Flash which the LPC3250 bootstrap software will load into SRAM and execute. This secondary bootloader will then initialize the SDRAM, copy application code from NOR Flash into SDRAM, and then transfer execution to SDRAM.

For a more detailed description of the boot process, [please refer to the phyCORE-LPC3250 Hardware Manual](#)

- Open the **Bootloader.uv2** project from the μVision3 menu *Project / Open Project*.
- Browse to **C:\Keil\ARM\Boards\PHYTEC\LPC3250\Bootloader_NOR**



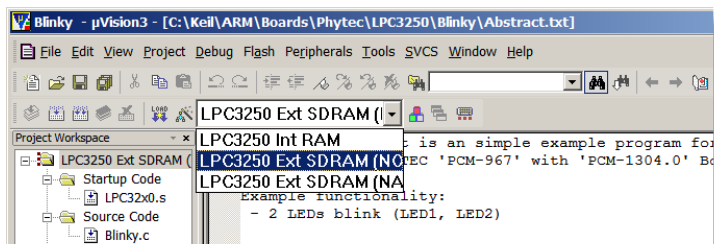
- Download the code into Flash memory by either selecting the *Download to Flash Memory* icon  on the build toolbar or in the main menu bar select *Flash / Download*.
- The individual steps of the Flash download procedure can be viewed at the bottom of the μVision3 **Output Window - Build** tab.
- Wait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




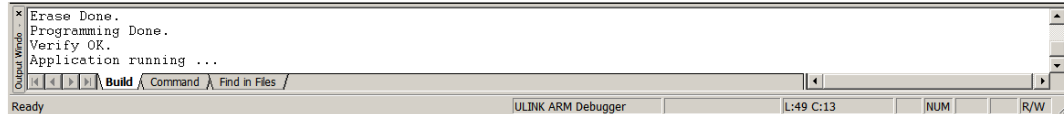
6.2 Loading the Blinky Application to NOR Flash

This section shows how to download the **Blinky** executable into the external NOR Flash

- Open the **Blinky** project as described in section 3.3
- In the **Select Target** pull-down menu be sure that the *LPC3250 Ext SDRAM (NOR)* 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 μ 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 code will execute without further user interaction.



- Push the **reset button** located under CB13 and CB46 on the Carrier Board

Successful execution of the program will alternately flash LED1 and LED2 (D15 and D18) on the Carrier Board.

Select *Close Project* from the *Project* pull-down window.

7 Getting More Involved

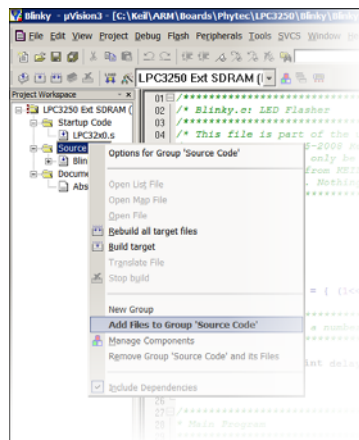
What you will learn with this example:


- how to create a new project with the **Blinky** project as template

7.1 Creating a New Project

For ease of creating a new project for the phyCORE-LPC3250 it is recommended that the given **Blinky** project is used as a template. The **Blinky** project already has the target, linker, compiler, debugger, and startup code correctly configured for the phyCORE-LPC3250 SDRAM, NAND, NOR target. Here are some suggestions for getting you started:

- Create a new directory for your project and copy the contents of **Blinky** into the new directory.
- From the **Project Workspace** you can manage components of your project and add or remove source files, add code libraries, etc... For example, add source code from the **Project Workspace** window by right-clicking on the *Source Code* folder and selecting *Add Files to Group 'Source Code'*.



- Advanced users can modify the target, compiler and linker settings from the Options for Target area by selecting the Options for Target icon  on the build toolbar or from the main toolbar menu select Project / Options for the Target'