AN10498

LPC952 debugging/programming using the debug interface

Rev. 01 — 20 July 2006   **Application note**

PHILIPS

Revision history

Rev	Date	Description
01	20060720	Initial revision

Contact information

For additional information, please visit: http://www.semiconductors.philips.com

For sales office addresses, please send an email to: sales.addresses@www.semiconductors.philips.com

AN10498_1

Application note **Rev. 01 — 20 July 2006** **2 of 16**

1.   Introduction

The P89LPC952 is a microcontroller with on-board flash and a 2-wire debugging interface. The 2-wire debug interface can both be used to program and debug the application that the microcontroller is used in. This application note will describe in detail how to use the development tools to program the internal program Flash memory, and how to debug the application using the same development tools.

2.   Development tools for the LPC952

This section describes different tools that are available for the development for the LPC952.

These tools include:

1. MCB950 Development board; Hardware

2. FS2 System Navigator 2-wire debugger; Hardware

3. Keil µVision 3 IDE (Integrated Development Environment); Software

2.1   MCB950 development board

The MCB950 is a low cost development board for the LPC952 microcontroller. The LPC952 can enter debug mode on this board using the FS2 System Navigator for emulation.

Either the FS2 System Navigator or the FlashMagic ISP software can program the LPC952 microcontroller on the MCB950 development board.

Fig 1 shows the MCB950 development board.

Fig 1.   MCB950 development board

2.2 FS2 system navigator

The FS2 System Navigator is the 2 wire debugger that can be used for emulation on the LPC952. The FS2 System Navigator is also capable of programming the internal Flash memory of the LPC952.

The latest firmware for the FS2 System Navigator can be downloaded from:

http://www.semiconductors.philips.com/fs2_sysnav_firmware

Follow the software installation procedure provided by FS2.

Fig 2 shows the FS2 System Navigator.

Fig 2. FS2 System Navigator

2.3 Keil's µVision 3 IDE

The MCB950 comes with a CD that contains the LPC Development Studio version of the µVision 3 IDE. The latest version of the LPC Development Studio can be downloaded from: http://www.semiconductors.philips.com/keil_LPCdevstudio

µVision 3 will be the IDE used for both the programming and the debugging of the LPC952.

# 3. Programming the LPC952 using the FS2 system navigator

In section 2.2 the FS2 System Navigator and the software installation for this tool was described.

In this section we will look at how to use the System Navigator and the µVision 3 IDE to program the Flash memory of the LPC952 on the MCB950.

3.1 Hooking up the system navigator to the MCB950

The power for the MCB950 is supplied through the USB cable. The System Navigator can be hooked up to the 10-pin connector on the MCB950. Make sure the Jumper on the MCB is in the RUN position.

AN10498_1

Application note **Rev. 01 — 20 July 2006** **4 of 16**

Fig 3 shows the MCB950 development board.

Fig 3.   MCB950 development board
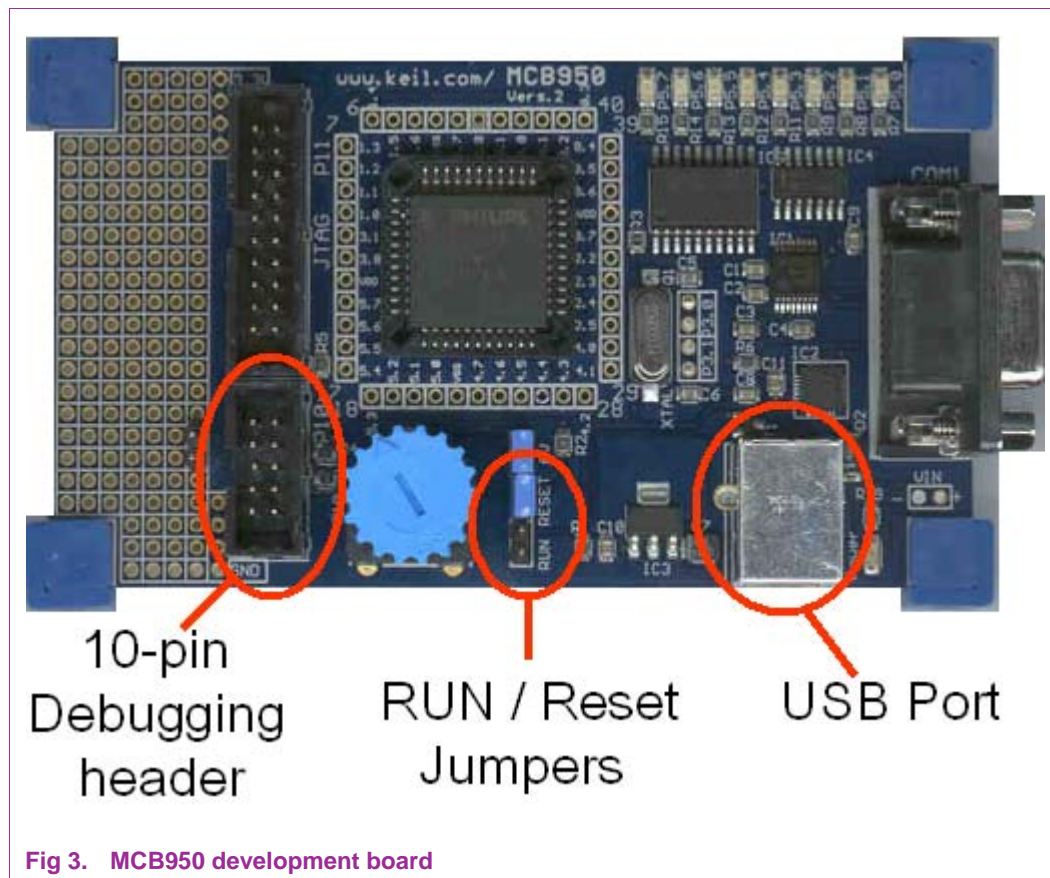
3.1.1 System navigator set-up

The System Navigator can be set-up with the Console program that is included with the System Navigator software.

Start up the System Navigator Console (with Connect Dialog). You should get a window as in Fig 4.

Click the following radio buttons:

    SysNav

    USB

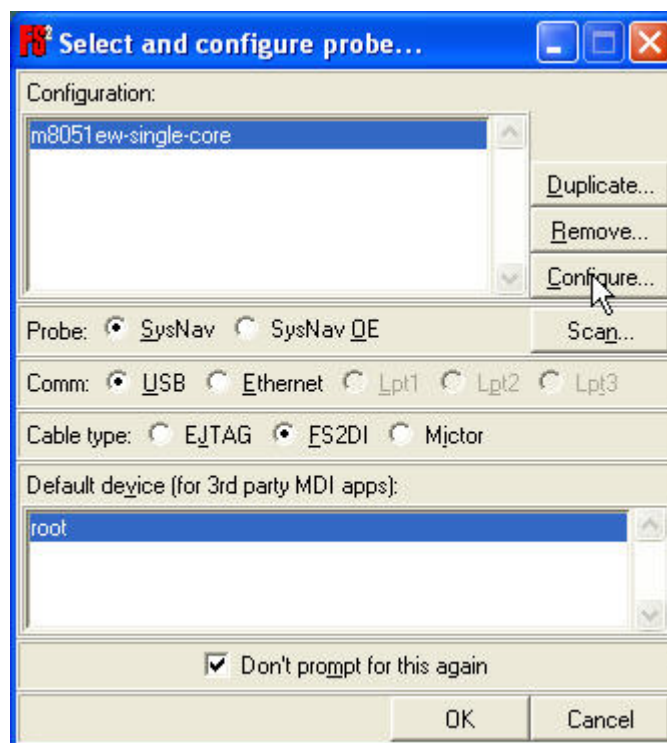   FS2DI

AN10498_1

**Application note** **Rev. 01 — 20 July 2006** **5 of 16**

Fig 4.   System Navigator Console

Next go to the Configure window in the System Navigator Console. In the Configure window make sure the following settings are made in the window:

Tck Rate is set to 1000000

ResetDuration is set to 100

ResetAsserted is set to low

ResetNegated is set to off

The settings in the Configure window are shown in Fig 5.

   a. ResetDuration   b. ResetAsserted   c. ResetNegated

Fig 5. System Navigator Console (Cofigure window)
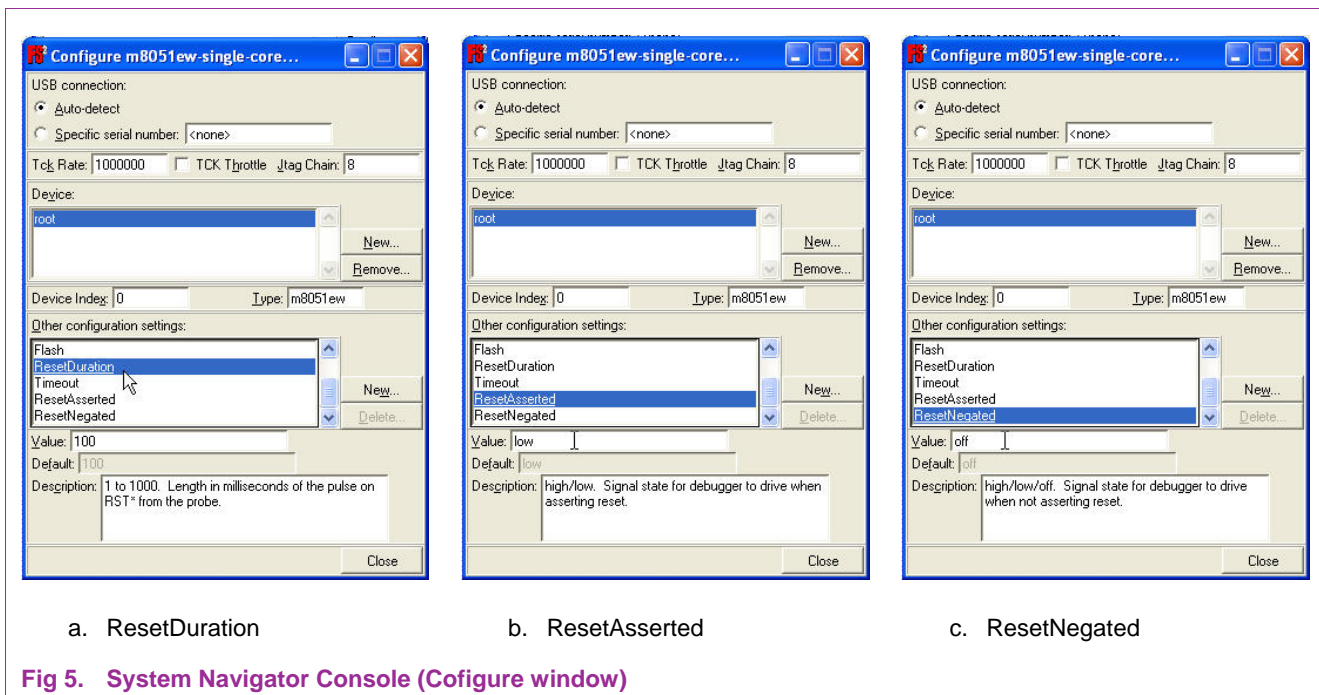
3.1.2 Testing the set-up

After the System Navigator is set-up the console window will open. From this console a variety of commands can be used. All the different commands for the System Navigator are described in the FS2 Getting Started manual. For the purposes of this appnote we will just use 1 command to test the communication between the LPC952 microcontroller and the System Navigator.

In the System Navigator Console type the "Status" command. The Console will display the OCI version of the LPC952 device. The correct OCI version for the LPC952 microcontroller is 0xF2.

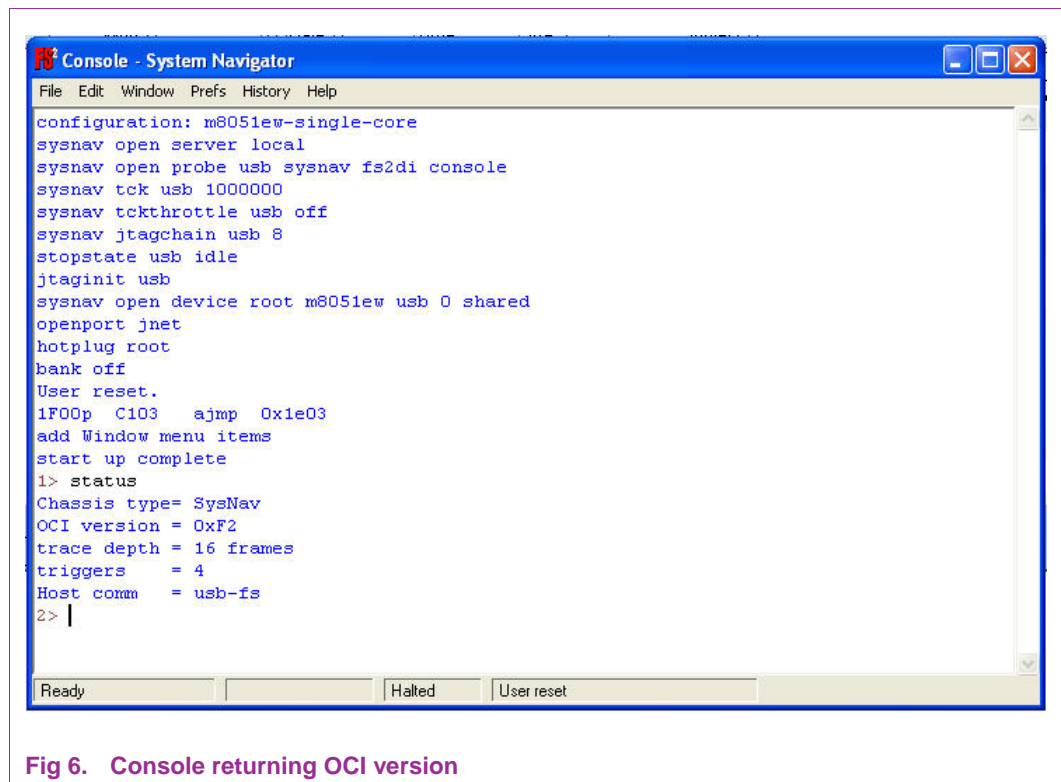Fig 6 shows the System Navigator Console return the correct OCI version.

Fig 6.   Console returning OCI version

If an error occurs check that the USB is powering the MCB950 board, the Power light should be on.

Check if the RUN / Reset jumper is set on the RUN position.

3.2   Using µVision3 to program the LPC952

The µVision3 IDE can interface to the FS2 System Navigator Console software. This way the LPC952 can be programmed through the same IDE as the code / firmware for the LPC952 is developed in.

3.2.1   Setting up µVision3 to use the FS2 System Navigator as Programmer

Open µVision 3, open the project that you would like to program / debug using the System Navigator. If you don't have a project the P5_Blinky example, that is include in the ZIP file with this appnote, can be used.

Go to options for Target select the Utilities tab. From the list of "Target Driver for Flash Programming" select the FS2/Keil M8051E Driver. The selection of the driver is shown in Fig 7.
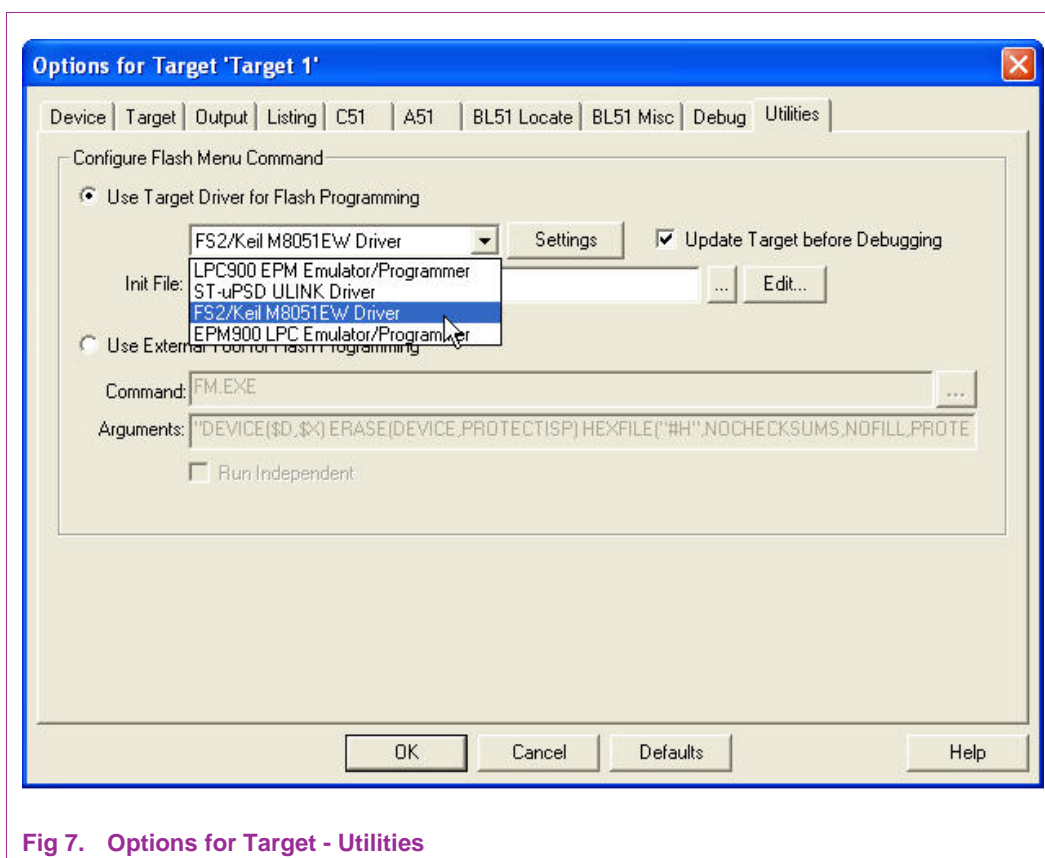
Fig 7. Options for Target - Utilities

Go to the Settings of the target driver and make sure the following are set-up

    Erase is selected

    Program is selected

    Verify is selected

    Protect ISP is selected

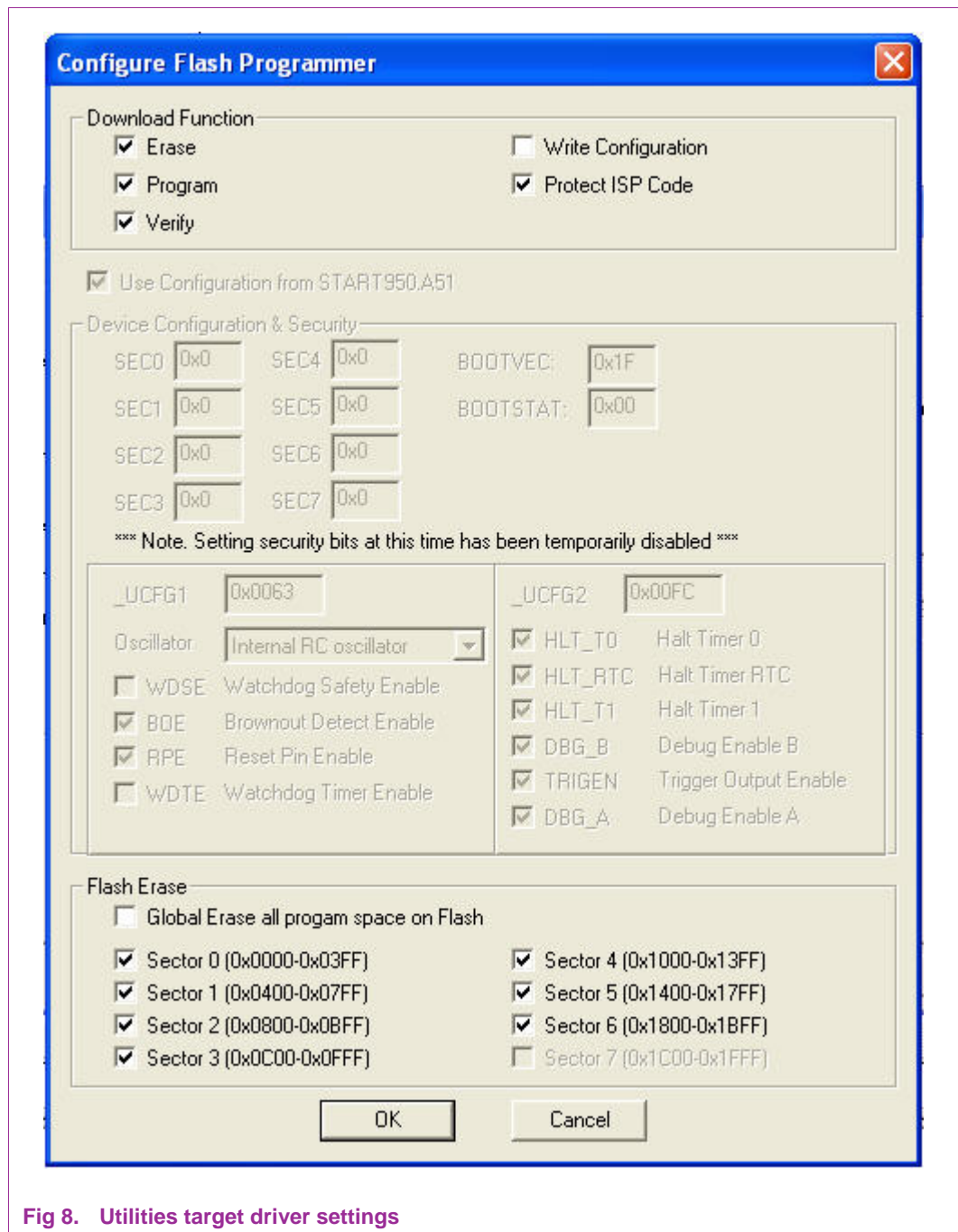   Write configuration

Fig 8.   Utilities target driver settings

3.2.2   Programming the LPC952

Once the µVision IDE is set up to use the System Navigator it is quite easy to program an LPC952 device.

The µVision IDE has a LOAD shortcut button that will program the selected target with the selected target driver settings. Once the LOAD button is pressed µVision 3 will open a System Navigator Console and send the correct commands through the console to the LPC952 microcontroller for erasing, programming and verifying.

Fig 9 shows the LOAD button, and the IDE status
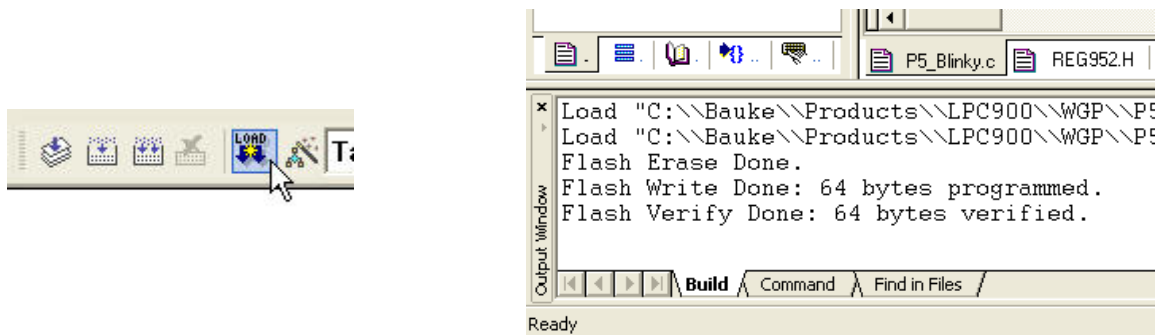
a. LOAD button b. Programming status

Fig 9. IDE programming interface

4. Debugging the LPC952 using the FS2 System Navigator

This section describes how the System Navigator can be used to debug the LPC952 in an application target. How to interface the debugger to the µVision IDE will also be shown.

4.1 Setting up µVision3 to use the FS2 System Navigator as Programmer

Setting up the System Navigator as a debugger should be quite easy if you have followed the previous steps with programming the LPC952 with the System Navigator.

Open µVision 3, open the project that you would like to program / debug using the System Navigator. If you don't have a project the P5_Blinky example, that is include in the ZIP file with this appnote, can be used.

Go to options for Target; select the Debug tab. Select to use the FS2/Keil M8051E Driver. The selection of the driver is shown in Fig 10.
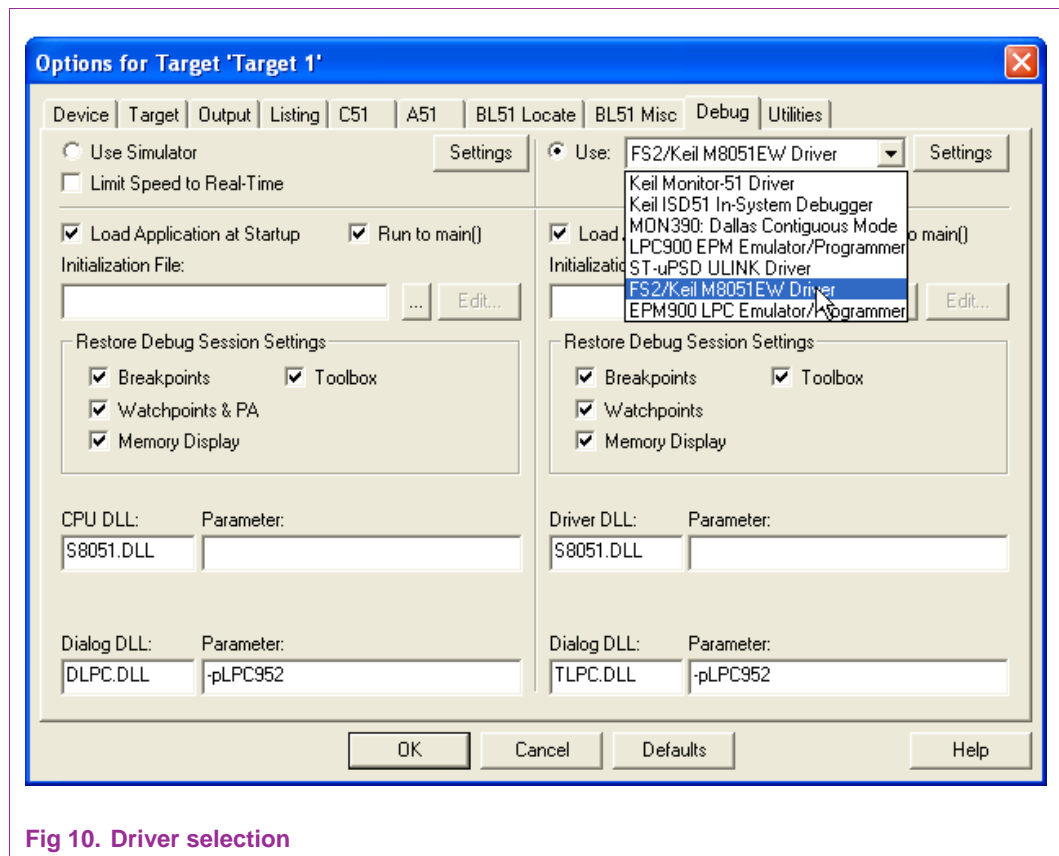
AN10498_1

Application note **Rev. 01 — 20 July 2006** **11 of 16**

Fig 10. Driver selection

In the settings window the FS2 System Navigator Console will be opened and the same settings as described in section 3.1 can be used.

4.2 Entering debug mode

Once the debugger is set-up correctly the device has to be programmed with the code from the project you would like to debug. The steps that are shown in section 3 cover programming. Once the programming is set-up the LOAD button from the µVision 3 IDE can be used to program the code from the latest build of the open project.

Now clicking on the debug icon can enter debug-mode. Fig 11 shows some debug icons.
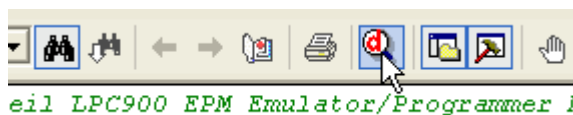
a. Debug icon   b. RST, RUN, single step, step over icons

Fig 11. Debug interface

Once in debug-mode, double clicking on a code line can set breakpoints. Fig 12 shows a breakpoint in the C code. The yellow cursor indicates where the program counter is currently pointing in the code.

Fig 12. Breakpoint

The µVision IDE can also be used to display a memory window. Fig 13 shows a memory window of the on-chip data RAM.

Fig 13. Memory window in uVision 3

5. Conclusion

In this appnote the detailed steps have been described how to program and debug the LPC952 microcontroller with the FS2 System Navigator and the µVison 3 IDE. This knowledge can be used to program and debug LPC952 applications in the end target instead of on the MCB950 development board.

If the provisions for debugging have been made on the target board, this set-up can even be used out in the field if a problem occurs. The same debugger can be hooked up to a target board to debug the actual system and conditions that cause a failure.

If the firmware of the LPC952 needs to be updated in the field the LPC952 can be used to reprogram the LPC952 if needed.

AN10498_1

Application note **Rev. 01 — 20 July 2006** **14 of 16**

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. Philips Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, Philips Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — Philips Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — Philips Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a Philips Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Philips Semiconductors accepts no liability for inclusion and/or use of Philips Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

6.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

AN10498_1

Application note

Rev. 01 — 20 July 2006

15 of 16

7. Contents