Integrated development environment for
32bit Power Architecture® derivatives

Introduction

Figure 1. SPC5-UDESTK starter kit version

This document, constituted by two main sections, Getting started and User guide, is related to an ST dedicate version of PLS commercial debugger product.

The ST version is a starter kit constituted by a software and an hardware component; the software is a starter version of PLS UDE (Universal Debug Engine) and is defined as SPC5-UDE/STK (starter kit version); its corresponding hardware, the PLS USB/JTAG adapter, is named SPC5-UDESTK as per the label applied on it.

Contents

List of figures

1 Overview and naming convention

This document will help you to configure and install the hardware and software tools necessary to operate the SPC5-UDE/STK starter kit version.

At the end of the instructions described in this document, you will have a running environment that could be used as a starting point for further development or evaluation work.

This tutorial goes step-by-step through the necessary procedures in order to:

- Install the SPC5-UDE/STK starter kit version
- Configure the SPC5 board and connect it to the host PC
- Set up a project
- Debug a sample application.

Note: *If you need more information about the SPC5-UDE/STK, please contact your nearest STMicroelectronics sale's office. Contact information is available on STMicroelectronics website: http://www.st.com/web/en/support/online_tech_support.html.*

1.1 Feedback

The SPC5-UDE/STK starter kit version is part of the SPC5 MCU' s family tool chain based on the ST SPC5 Studio Integrated Development Environment.

Regarding any comments about SPC5-UDE/STK use the website *http://www.st.com/web/en/support/online_tech_support.html*

You are invited to browse to the PLS Development Tools' website at *http://www.pls-mc.com* to get latest information about the professional and full version of Universal Debug Version (UDE).

The PLS Development Tools company welcomes feedback on products and documentations. If you have any comments, suggestions or improvements about the products you are using, please use the Feedback Form from website *http://www.pls-mc.com, send an email to sales @pls-mc.com*.

2   Getting started

2.1   Before you start

To execute this getting started, it is necessary to have a:

- PC equipped with USB port
- Operating system: Windows 8, Windows 7, Windows Vista, Windows XP SP3
- A connection to the Internet during installation for downloading and license activation
- Administrator or power user's rights to install the required programs
- One of the ST SPC56 boards equipped with a SPC56xx microcontroller:
    - Discovery board
    - Discovery+ board
    - Premium Evaluation board

The full list of supported boards is available on ST WEB in the Automotive MCUs page in the "Resources Hardware" area which direct URL is:

http://www.st.com/stonline/stappl/productcatalog/app?page=partNumberSearchPage&levelid=FM2098&parentid=1675&resourcetype=HW

- Power supply for the starter kit board
- A mini-USB cable
- A SPC5-UDESTK: PLS USB/JTAG adapter for SPC5 with a 14-pin JTAG connector

2.2   System requirements

To run SPC5-UDE/STK starter kit version at least the following minimum system configuration is required:

- Microsoft Windows® compatible PC
- 1 GHz or faster 32-bit (x86) or 64-bit (x64) processor
- 1 GByte RAM (32-bit) or 2 GByte RAM (64-bit), 1 GByte available hard disk space
- Operating system Windows® XP, Windows® Vista, Windows® 7, Windows® 8
- Microsoft.NET™ Framework 3.5 SP1
- Microsoft Windows® Scripting Host V5.6, Microsoft Internet Explorer® 6.0 or higher
- Java Runtime Environment 6, Adobe® Acrobat Reader 4.0 or higher

2.3   Downloading the latest SPC5-UDE/STK version

The SPC5-UDE/STK starter kit software is available for downloading from the following website: *http://www.pls-mc.com/spc5-udestk*

2.4   Installing SPC5-UDE/STK

Run setup.exe and follow in the installing instructions. The SPC5-UDE/STK starter kit version will be installed into the directory C:\SPC5Studio\ude

2.5 License registering and activation of SPC5-UDE/STK

The starter kit version of SPC5-UDE/STK can be used for an evaluation purposes without registration. In this case, the SPC5-UDE/STK is restricted to a limited code size for downloading of 128 kBytes.

The registered version of SPC5-UDE/STK unlocks the limitation of code size downloading. STMicroelectronics offers licenses with a validation period of 1 year with option to extend to further 2 years. Send the following information via the PLS website form at:

http://www.pls-mc.com/spc5-udestk

1. Company data
2. Customer data
3. Host PC's MAC (Media Access Control) address
4. Serial number of PLS USB/JTAG adapter for SPC5 (see the sticker backside of the adapter)
5. Activation code, delivered by STMicroelectronics

After that, the license file will be delivered by PLS.

Note: *SPC5-UDESTK functionality is guaranteed only for ST evaluation boards. Visualization functions at runtime and script support are not supported features.*

2.6 Installing hardware

2.6.1 Static electricity precautions

Electrostatic Discharge (ESD) can damage a sensitive electronic component. Under several conditions static electricity and ground potential differences between the host PC, the SPC5-UDESTK PLS USB/JTAG adapter for SPC5 and the user's target hardware can build up high voltages - over 10000 V (10 kV) in some cases. The electrostatic discharge of this build-up voltage results in fast high current waveforms and fast magnetic (H-field) or electrostatic (E-field) disturbances. The discharge into the electronic components and circuitry can damage or destroy hardware components, resulting in failures and reduced reliability.

To protect your hardware against damage from static electricity and ground potential discharge, you should follow some basic precautions:

1. Please ensure that the static electricity and ground potentials between the SPC5-UDESTK PLS USB/JTAG Adapter for SPC5, the host PC and the starter kit board are balanced
2. Establish the target connection and power-on the systems.

2.6.2 Driver installation of PLS USB/JTAG adapter for SPC5

If the previous steps are done successfully, the SPC5-UDE/STK starter kit version installed the hardware driver components automatically, when the SPC5-UDESTK PLS USB/JTAG adapter for SPC5 is connecting to the host PC via the mini-USB cable the first time.

Connect the SPC5-UDESTK PLS USB/JTAG adapter for SPC5 to the PC host system using the mini-USB cable. The Windows system will find a new hardware device on your system

called "SPC5-UDESTK PLS USB/JTAG adapter for SPC5" in the "Universal serial bus controllers" group.

Figure 2. Universal serial bus controller

GAPGCFT0608131153CFT
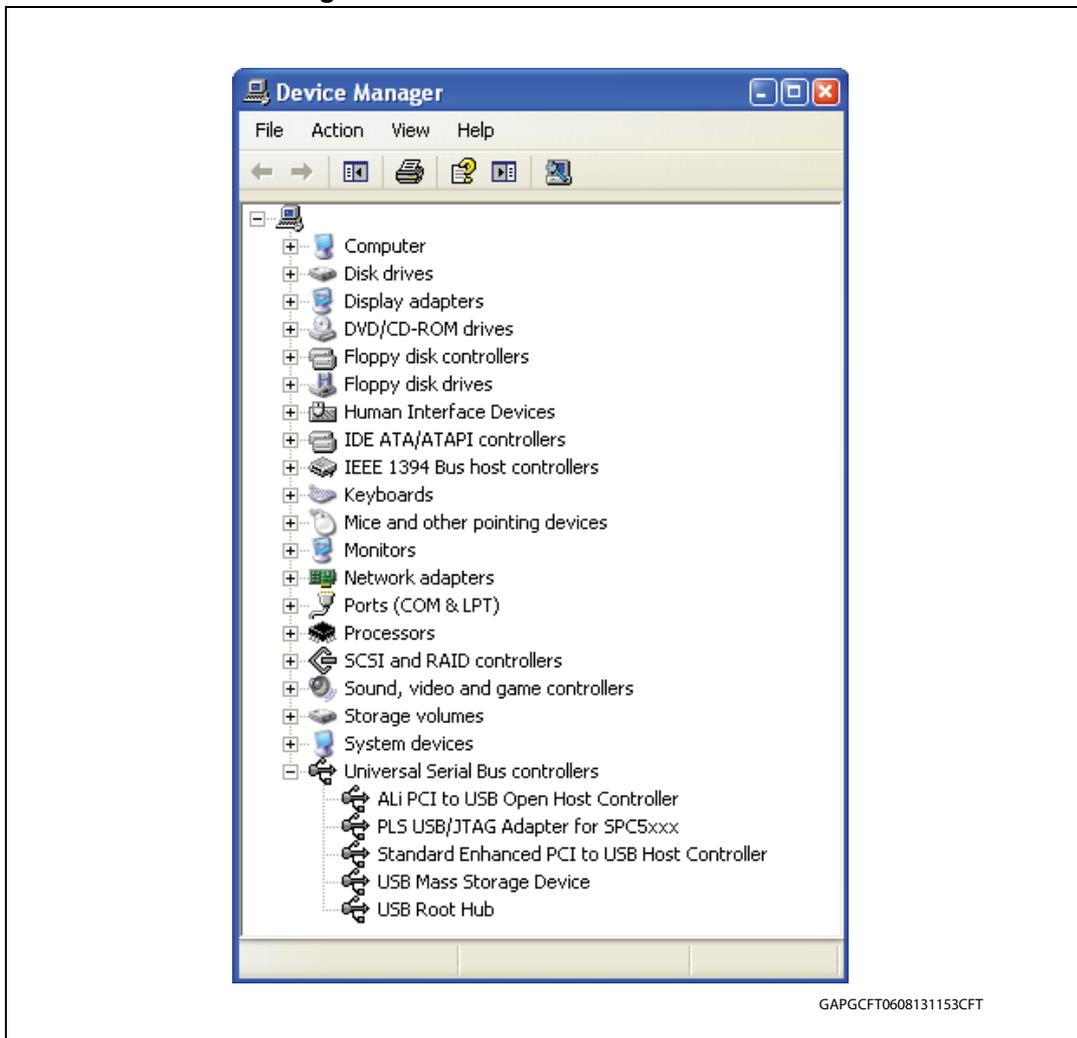
2.6.3 Trouble shooting

If the previous step fails, you have to install the driver software manually.

1. Try to run the driver installing setup first at
    C:\SPC5Studio\ude\driver\JtagUsbDriver\InstallUsbJtagDriver.bat

Alternatively, you can browse the driver directly, when Windows is asking for that.

1. Connect the SPC5-UDESTK PLS USB/JTAG Adapter for SPC5 with your PC using the mini-USB cable. The Windows system will find a new hardware device on your system called "PLS USB/JTAG adapter for SPC5" in the "universal serial bus controllers" group and you will be prompted to install a new device driver.
2. Click Next to continue
3. Click Search for a suitable driver and click Next

4. Click Specify a location, click Next and browse for the driver file
    C:\SPC5Studio\ude\driver\JtagUsbDriver\Driver\plsusbjtag.inf driver

5. Click Next and Finish

2.6.4 Installing SPC5-UDESTK PLS USB/JTAG adapter for SPC5

The hardware installation of Universal Access Device is done within the following steps:

1. Connect the **SPC5-UDESTK PLS USB/JTAG adapter for SPC5** with an USB port of the host PC by the mini-USB cable ①

2. Plug the **SPC5-UDESTK PLS USB/JTAG adapter for SPC5** into the 14-pin JTAG debug connector of the SPC5 starter kit board ②

3. Plug in the power supply of the starter kit board ③

4. Start your computer system and login with administrator rights

Figure 3. SPC56-L-discovery kit with installed SPC5-UDESTK PLS USB/JTAG adapter


GAPG0608131411CFT
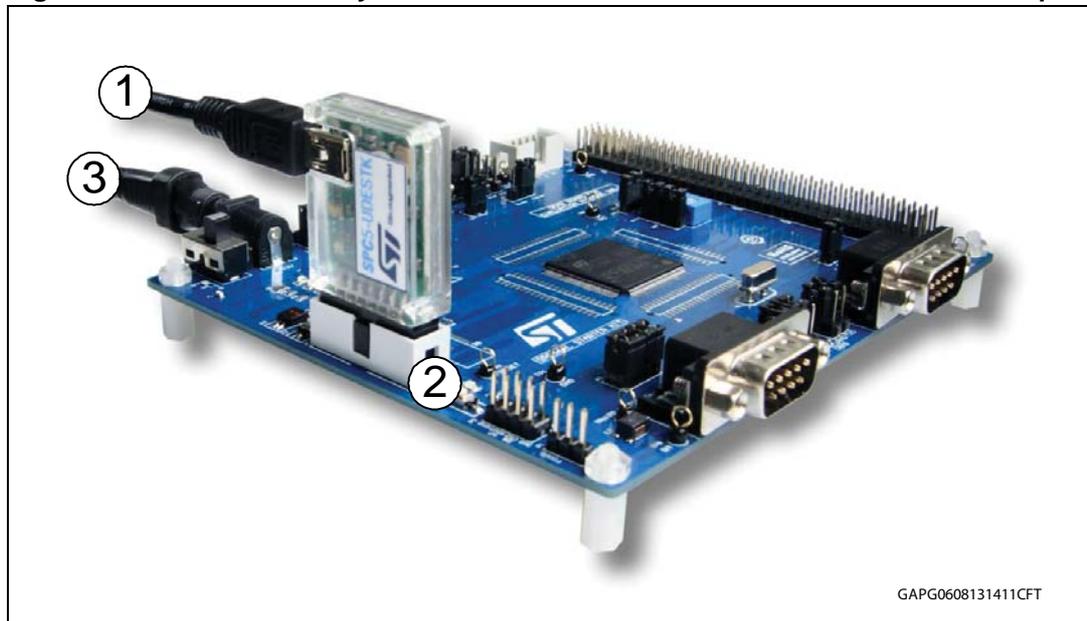
2.6.5 Technical details of SPC5-UDESTK PLS USB/JTAG adapter

The SPC5-UDESTK PLS USB/JTAG adapter for SPC5 is the adapter between the USB interface on the Host PC and the JTAG interface on the starter kit board.

Plug in the adapter into convenient mini-USB connectors and target connectors only.

Figure 4. SPC5-UDESTK PLS USB/JTAG adapter drawing
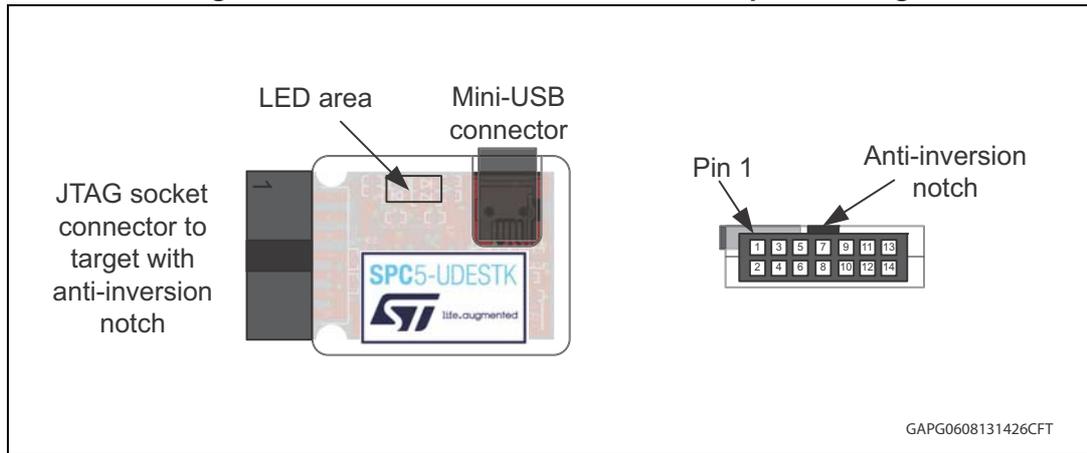
2.6.6 LED status indication

The green LED indicates the target's IO voltage on the target connector, the yellow LED indicates the target connect state, the red LED indicates the target running state to the user.

3 User guide

3.1 SPC5 Studio and SPC5-UDE/STK (UDE) integration

SPC5 Studio is a development environment containing compiler tools, editor and debug visualization and based on the eclipse platform. The SPC5-UDE/STK starter kit version can be integrated into the SPC5 Studio as debug plug-in.

Beside of this, the SPC-UDE/STK (UDE) starter kit version can be used as stand-alone debug environment. In this context, it is called UDE (Universal Debug Engine). The following chapter shows features of the UDE (Universal Debug Engine) in a stand-alone example.

3.2 A first example with SPC56L-Discovery +

We assume that you now have successfully installed the SPC5 Studio, the SPC5-UDE/STK (UDE) starter kit version and the PLS USB/JTAG adapter for SPC5.

In this section you will learn about:

- How to start SPC5-UDE/STK (UDE) starter kit version
- How to use the windows in SPC5-UDE/STK (UDE) starter kit version
- How to load code into the starter kit board and start it as well as furthermore principles of how to debug an existing application.

We recommend you to go through this tutorial step-by-step. This example is shown under Windows XP and using the SPC56L-Discovery+ starter kit board offered by STMicroelectronics.

3.2.1 Starting with SPC5-UDE/STK (UDE)

The SPC5-UDE/STK (UDE) comes with a set of example programs demonstrating the features of SPC5-UDE/STK (UDE). The example program, prepared for GNU Compiler and SPC56 starter kit boards is located in

C:\SPC5Studio\ude\samples\PowerPC\SPC56-L\timedemo

Once the operating system is up and running, double-click on the icon SPC5-UDE/STK on the desktop. Alternatively, SPC5-UDE/STK (UDE) may be launched also via Start – Programs – SPC5-UDE/STK – UDE Visual Platform. This will start the desktop of SPC5-UDE/STK (UDE) development system.

The next step is creating a new workspace. An SPC5-UDE/STK (UDE) workspace saves all configurations and settings of SPC5-UDE/STK (UDE) desktop, windows and their content, path and name of loaded files. The file extension is *.wsx.

Click New Workspace from the File menu and choose a new file name from the file selection box, e.g. SPC56-L.wsp.

Figure 5. New workspace

GAPGCFT0608131431CFT
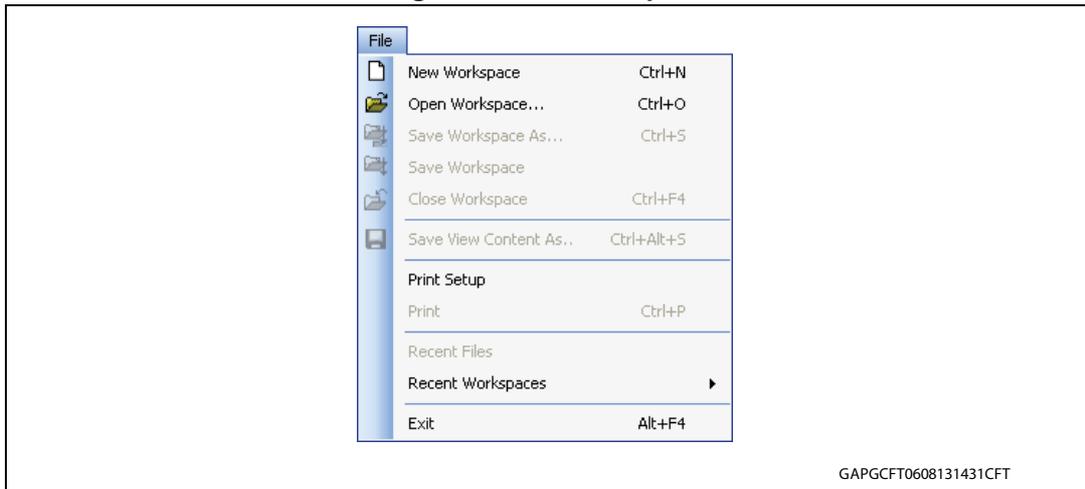
After creating the new workspace, you will be asked to select a target hardware configuration. SPC5-UDE/STK (UDE) comes with some default target configurations of starter-kits. Click "Default" button and enable the "Use a default target configuration" to select a predefined configuration.

Create a default target configuration for Power Architecture SPC56-L. Push the button Default in the "Select Target Configuration" dialog. Enable the Use default target configuration and select the corresponding configuration to your target

PowerPC – STM – XPC56XL Mini Module – STM XPC56XL Mini Module with SPC56EL70, Lockstep mode (Jtag)

Figure 6. Default configuration

GAPGCFT0608131434CFT
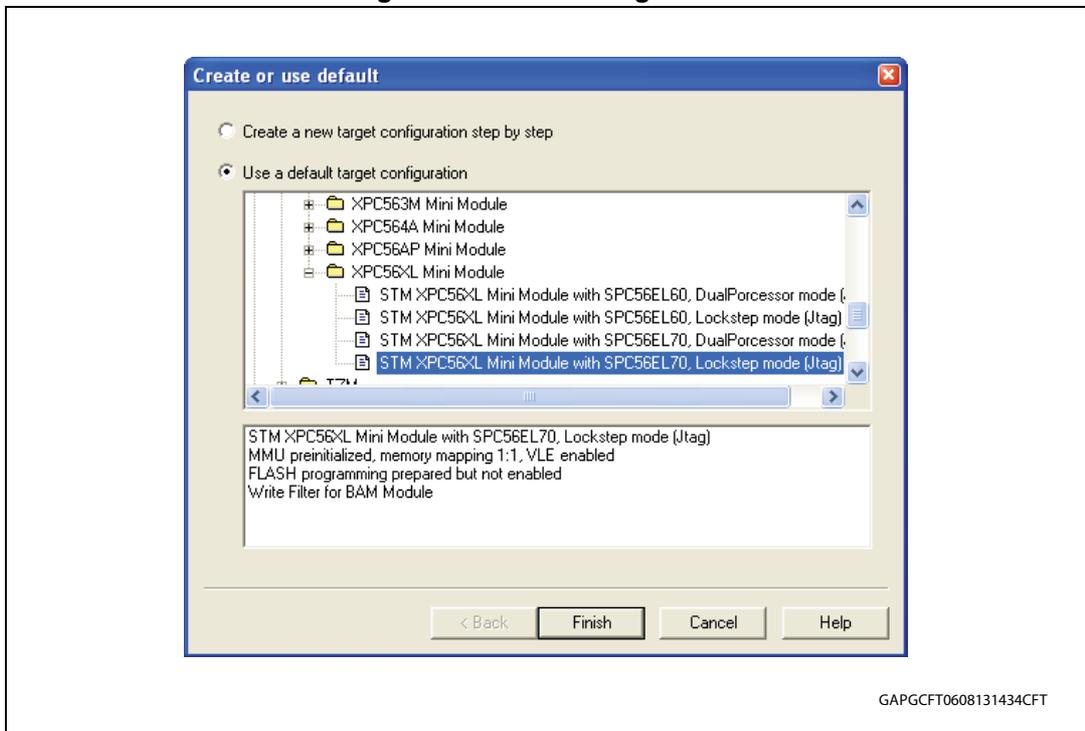
Push OK.

The SPC5-UDE/STK (UDE) will now try to connect to the target system.

When the connection is successfully established, the following message will appear in the command window: *Core::UDEDebugServer: Connection to SPC56EL70 target monitor established: PowerPC Target, JTAG-ID: 0x0AEA9041*

When launching SPC5-UDE/STK (UDE) for the second time, you may use either File - Open Workspace or File - Recent Workspaces to select the workspace and start a new session with settings from the saved workspace.

If you get an error message, return to the section Software Installation and make sure that all settings are correct. Refer also to the Troubleshooting section to get further information. If the problem persists ask the STMicroelectronics support team at: *http://www.st.com/web/en/support/online_tech_support.html for qualified help*.

3.2.2 Loading an executable

After having SPC5-UDE/STK (UDE) for Power Architecture SPC56L started, we want to load a program that can be executed on the SPC56L discovery board.

Figure 7. Load a program

GAPG0608131444CFT
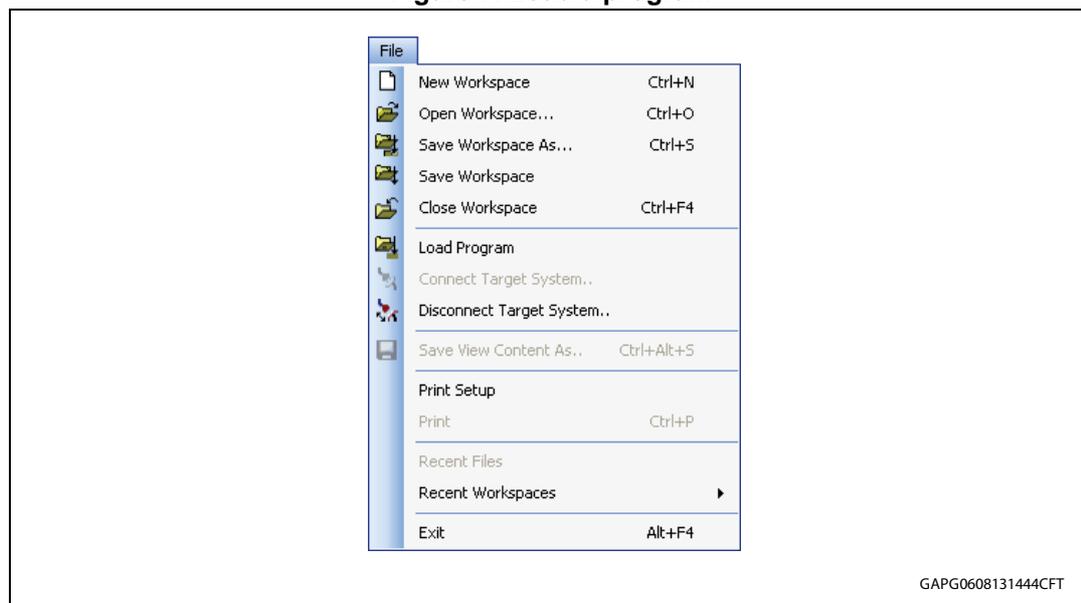
First, the GNU compiler variant of the *TimeDemo* example located for using in the internal RAM should be used. This example is a basic demonstration and toggles the LED on the SPC5L-Discovery only.

The example is located to the internal sRAM of the SPC56-L at memory address 0x4000'0000.

Select from the SPC5-UDE/STK menu Load Program in the File menu bar, browse to

C:\SPC5Studio\ude\samples\PowerPC\SPC56-L\timedemo\GnuPpc\iRam\timedemo.elf

In the workbench, the Program window appears showing the source code of the main function of the ample application. When clicking with the right-hand mouse button into the

program window, a context button appears to switch between source code only and source/assembly code display via the Mixed Mode entry.

3.2.3 Project management

A docked window at the left-hand side of the SPC5-UDE/STK (UDE) desktop houses the Explore Symbols tab where the application's source files and their inside procedures are shown after unfolding the markers.

If no workspace window is shown, you can make it visible via the menu Window - Project Workspace.

The SPC5-UDE/STK (UDE) project contains source files, C/C++ functions, address sections and user-defined breakpoints.

Figure 8. Project management
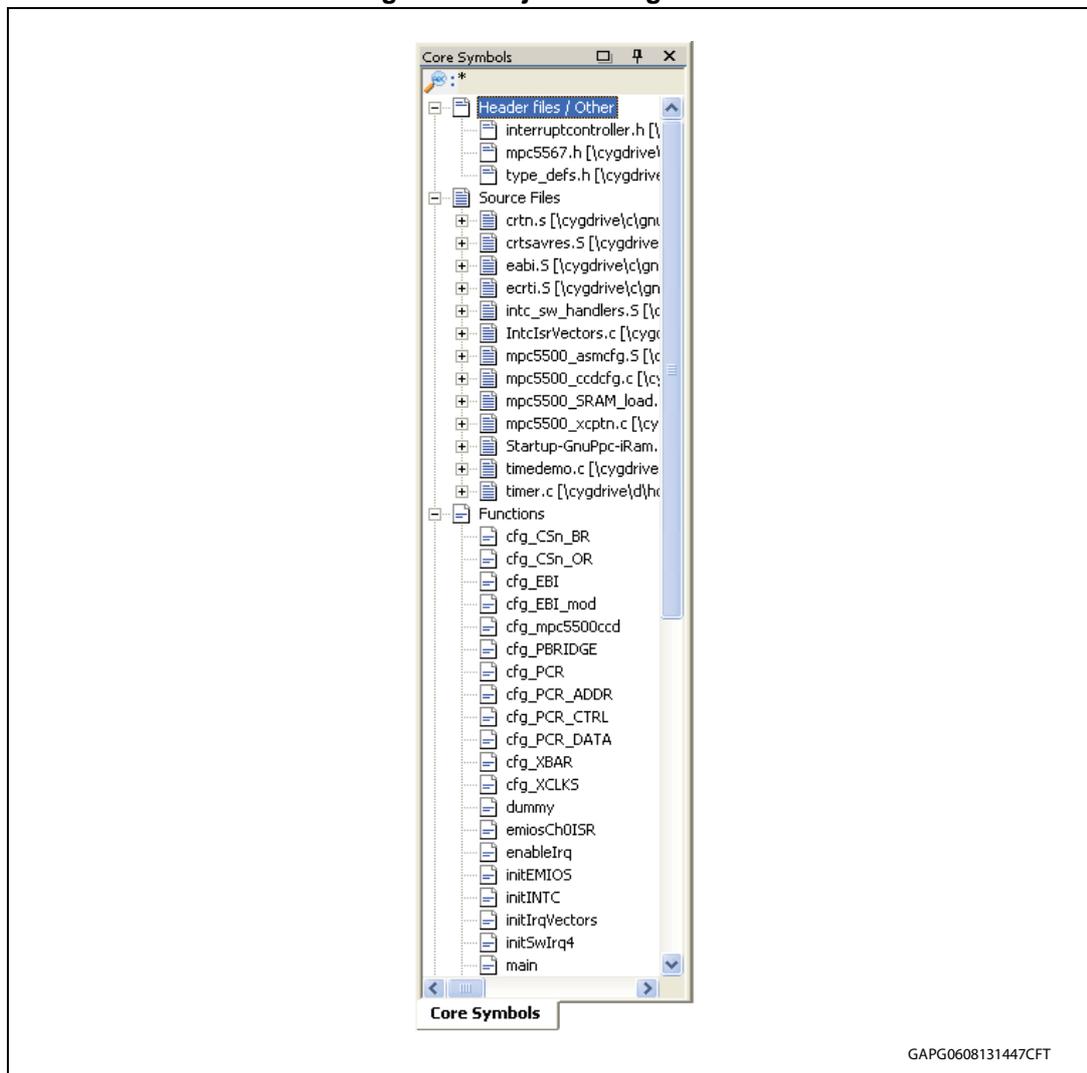
By double-clicking on one of the source files, the selected file will be brought into the Program window; by double-clicking on one of the procedures, the selected procedure is displayed.

After loading the program code, the Program window shows following content. In the Program window, a yellow arrow indicates the current instruction pointer position.

Figure 9. Example of program code
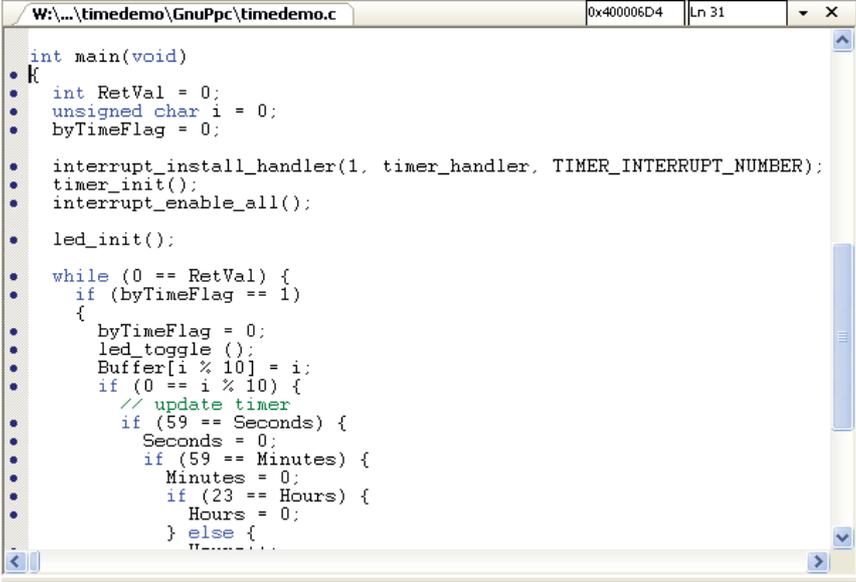

```
W:\...\timedemo\GnuPpc\timedemo.c          0x400006D4  Ln 31    ▼ ✕

int main(void)
{
  int RetVal = 0;
  unsigned char i = 0;
  byTimeFlag = 0;

  interrupt_install_handler(1, timer_handler, TIMER_INTERRUPT_NUMBER);
  timer_init();
  interrupt_enable_all();

  led_init();

  while (0 == RetVal) {
    if (byTimeFlag == 1)
    {
      byTimeFlag = 0;
      led_toggle ();
      Buffer[i % 10] = i;
      if (0 == i % 10) {
        // update timer
        if (59 == Seconds) {
          Seconds = 0;
          if (59 == Minutes) {
            Minutes = 0;
            if (23 == Hours) {
              Hours = 0;
            } else {
```

GAPG0608131450CFT

Warning: **After downloading a program executable, the instruction pointer is set to the entry point of the program. Usually the entry point is located at the start-up code. That is why the default C/C++ Program window above does not show an instruction pointer. To force the view of the current instruction pointer use the context menu Show Next Statement or the main menu Show - Show IP.**

3.2.4 Running and stepping through the application

After the application has been loaded successfully you may open now the menu Debug to run or step through the example procedures.

Figure 10. Debug menu
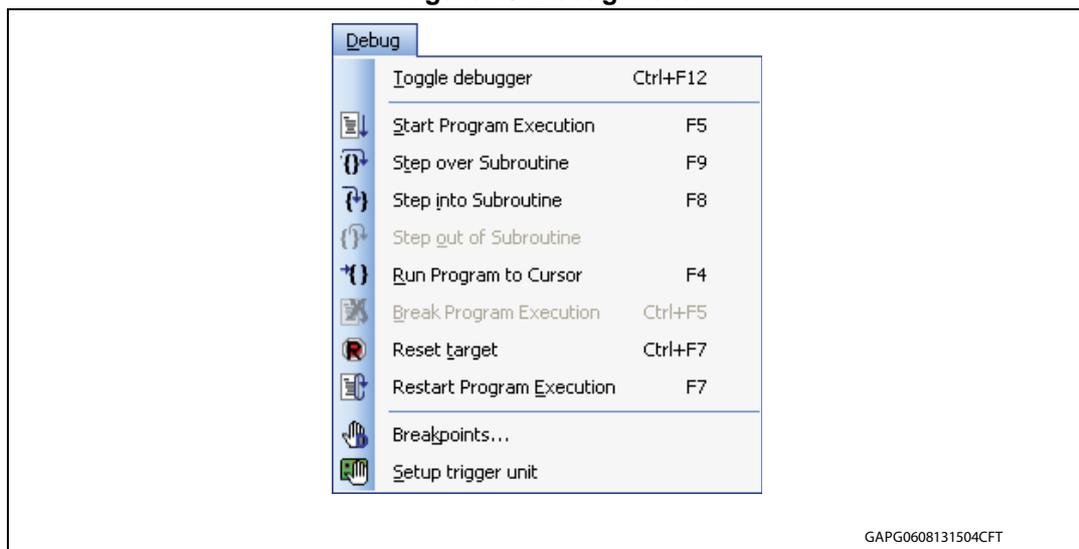
The icons from the Debug menu are also located in the tool bar covering the same functionality. Short-cuts are available, too.

Click now onto the Start Program Execution entry or button and watch the LED on the SPC56-L Discovery kit board flashing the TimeDemo code for UDE. When clicking onto Break Program Execution, the application is halted and the current instruction pointer position (code line) is displayed.

You may also step through the application by using Step over Subroutine (steps with freely executing subroutines in one-step) or Step into Subroutine with following function calls and executing subroutines instruction by instruction.

Warning: **For debugging the C/C++ parts of the example program only, the start-up code must be executed first. For this, make a Step over Subroutine from the Debug menu of UDE. After that, the IP will be shown at the main function; the start-up code has been executed.**

The application can be reloaded with Restart Program Execution. If the program is running already, it will be started immediately after the reload terminates.

3.2.5 Setting breakpoints

Now, we assumed that a loaded application is error-free and ready for running. However, for

debugging purposes single step executions and breakpoints have to be performed to watch program behavior and processor status.

Now we want to set a breakpoint in the *timedemo* program. To do this, click with the mouse on a line in a procedure and then click on the simple Hand symbol in the tool bar. A red-filled dot appears in the line indicating that the breakpoint has been successfully set. Alternatively, you may also select the menu Debug - Breakpoints or the Hand symbol in the tool bar marked with 'D' to open the Breakpoints dialog.

Figure 11. Setting breakpoints

GAPG0608131506CFT
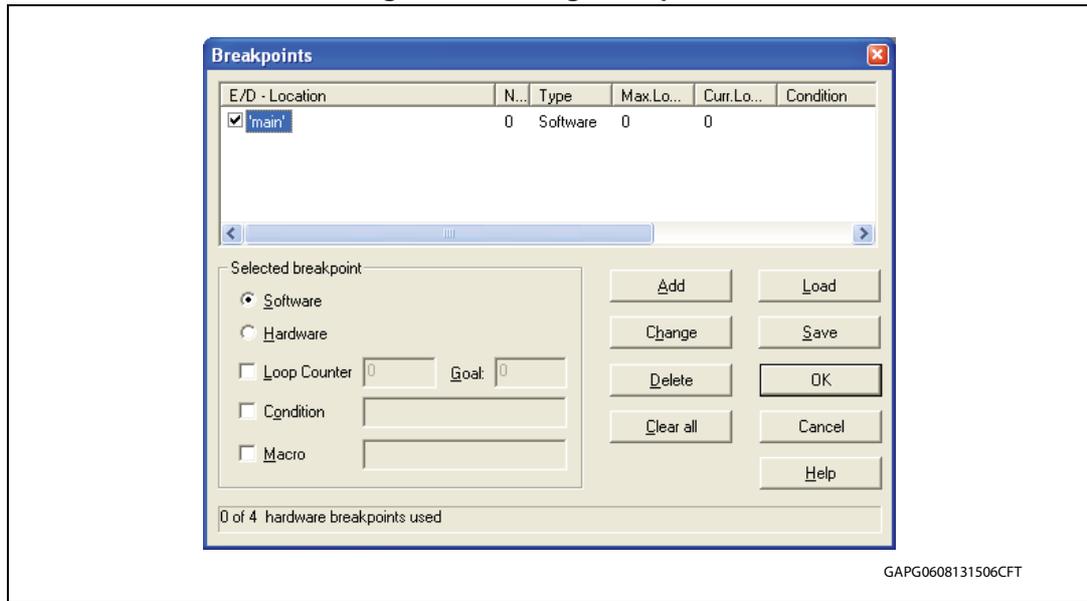
Within the Breakpoint dialog, breakpoints may be added, changed of type and deleted using the corresponding buttons. By clicking on the E/D (Enable/Disable) checkbox, you toggle the breakpoint between active and suspended. Disabled breakpoints are indicated by a red-shaped circle.

Now start the application again. The application will be executed until the first breakpoint in the execution path is reached. The application will be stopped then immediately.

Another possibility to execute certain portions of code without setting a breakpoint explicitly is placing the cursor into the line where the application is required to halt and then select Debug - Run Program to Cursor from the menu or Run to Cursor from the context menu.

3.2.6 CPU registers

The CPU register window is opened by the menu View - CPU Window or the corresponding tool bar button.

Figure 12. CPU register window

GAPG0608131515CFT
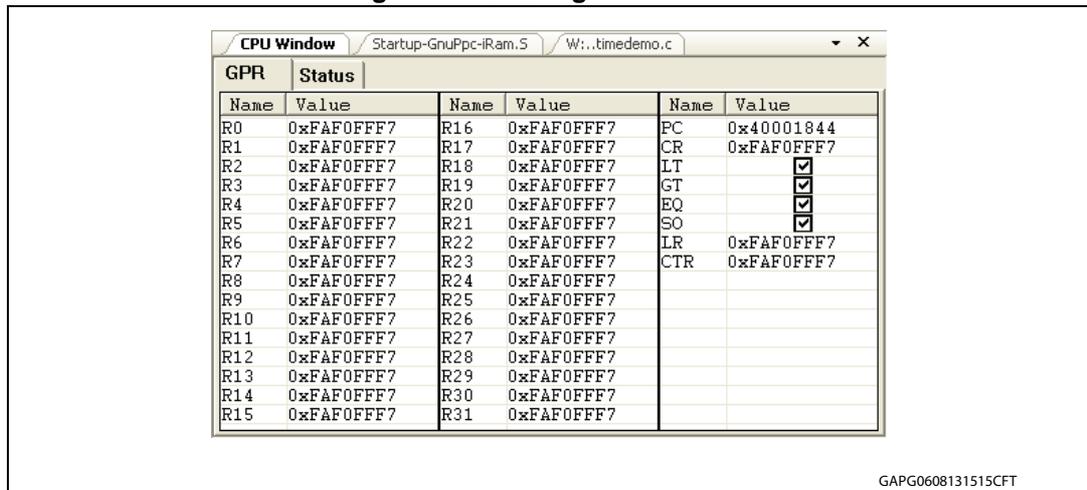
Perform a few single steps to see the CPU register values changing according to the executed instructions. Registers which values have been changed compared with the previous state are red highlighted to provide quick overview.

While the program is stopped (e.g. between single steps) you may alter the content of registers. Simply click on the register's value in the CPU registers window and type in the new value of the register.

3.2.7 SFR registers

Special function register values are changed in the same way in the SFR register window (open this window by selecting the menu View - SFR Window). To add a new register entry, select Browse Ins from the context menu of the SFR Register window and take the SFR that you want from the list.

Figure 13. SFR registers

GAPG0608131522CFT
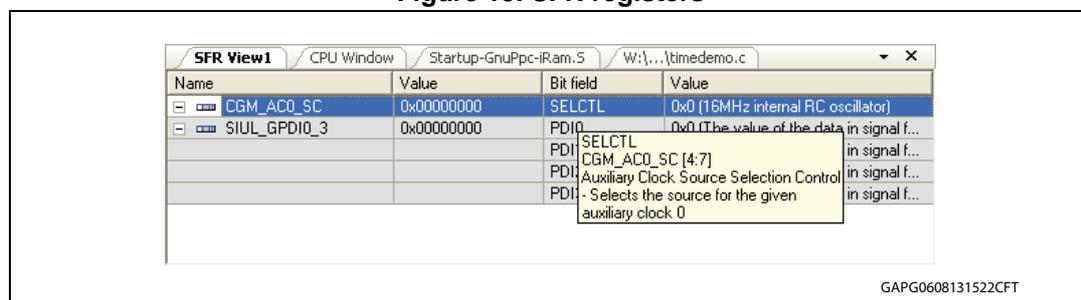
Tooltips show the address, the reset value of the current SFR and further information about the focused register. Expand a SFR for viewing all available fields composing this register. With a right click on the values you can change them.

Warning: **Various registers are protected which means that a special unlock sequence is required to change the register value. SPC5-UDE/STK (UDE) can unlock these registers. Use the context menu of the register name and disable the entry Write protect.**

3.2.8 Viewing variables

Viewing and changing global/static variables

All global and static variables from the C/C++ source code may be observed directly using the Watch window. Open the Watch window by selecting the menu View - Watch Window or the corresponding tool bar button.

Figure 14. Watch window

GAPG0608131528CFT
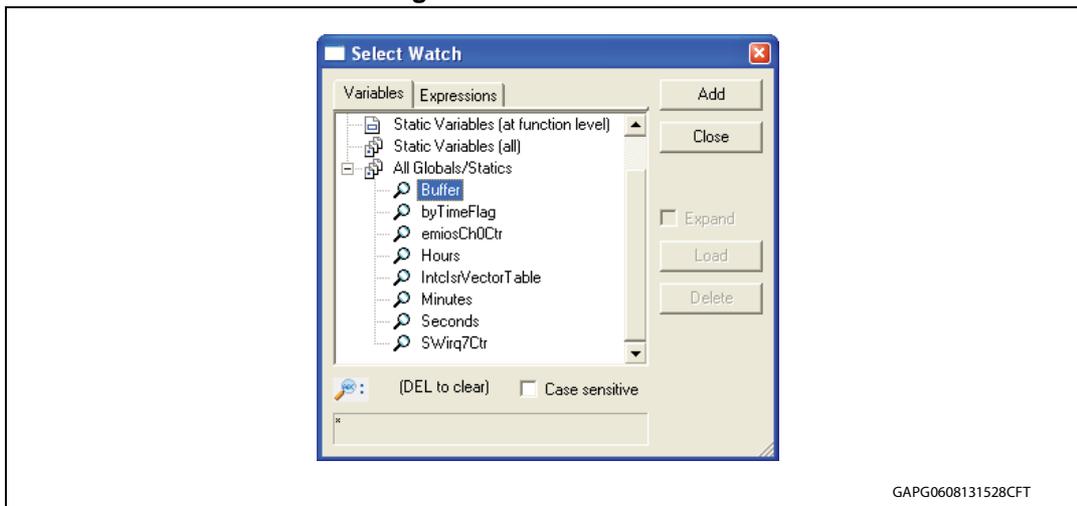
The variables can be added by double-clicking to <new variable> or using the context menu of the <new variable> entry via Browse Ins. The browser dialog shows you all available global and static variables. Click Add for adding a new variable to the watch window.

The variables are sorted in following groups:

- Global variables: shows all global variables with a global scope
- Static variables (at file level): shows only variables visible at a specific file
- Static variables (at function level): shows only static variables visible at a specific function
- Static variables (all): shows all static variables, which are not global variables
- All global/static variables: shows all global and static variables

If the variable is expandable, i.e. it is a pointer or an array, clicking on the '+' sign in front of the variable's name will expand it. This means, that the location where the pointer points to or the content of the elements of the array will be displayed. Expanding is possible for more than one level.

Variable values can be changed easily by double-clicking in the value area or pressing <F2> and typing in the new value.

Watch tips

Furthermore, SPC5-UDE/STK (UDE) offers so-called Watch tips, which show you the content of simple variables in the Program window. Highlight i.e. the *Buffer[]* variable from the *main* function by a double-click, move the mouse pointer over and the content will be displayed in a watch tip after a short waiting time.

Figure 15. Watch tips

GAPG0608131532CFT
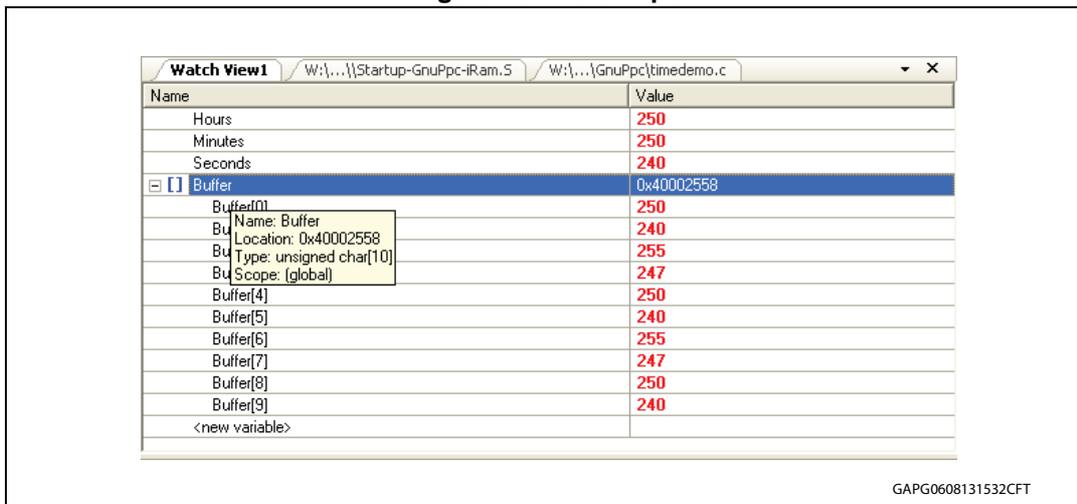
Viewing and changing local variables

Viewing local variables is provided by the Locals window that can be reached via the menu View - Locals Window. In this window, all currently valid local variables are displayed. The value of the local variable can be changed in the same way as in the Watch window.

3.2.9 Trigger functions

This chapter demonstrates how the Power Architecture Book E defined triggers of the SP56x derivatives can be used for debugging purposes. Again, the starting point is SPC5-UDE/STK (UDE) with the application timedemo.elf loaded. We want to create a trigger configuration that stops program execution when a write access to the variable Buffer[0] occurs.

Open the Hardware Debug Resources dialog by menu Debug – Setup Trigger unit. Select the rider Data Address. We want use DAC1 as trigger comparator. Enter the address of Buffer[0] into the address box. For simplification C-style expressions could be used, so simple enter &Buffer[0].

Figure 16. Hardware debug resources

GAPG0608131537CFT
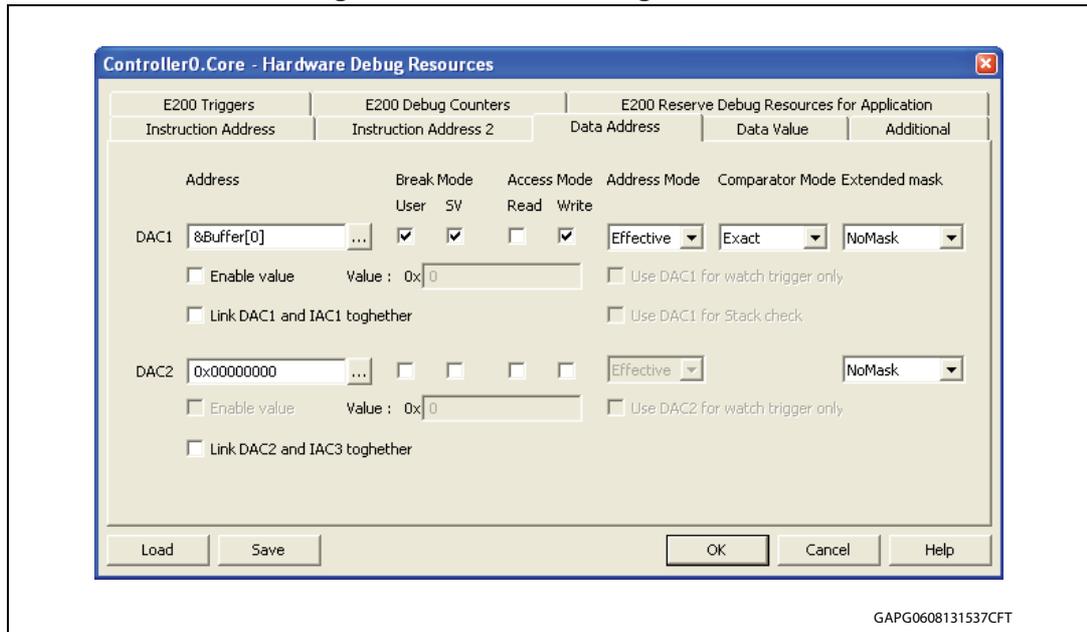
Enable User and SV (Supervisor) as Break Mode, select Write Access mode, Effective address and Exact comparator.

Start the application. The application stops when the fist write to *Buffer[0]* occurs.

You will find the message

```
Core::UDEDebugServer: halted by trigger event
```

inside of the command view.

3.2.10 Leaving the project

To leave the current Project select File - Close Workspace from the SPC5-UDE/STK (UDE) Desktop menu. The workspace with all settings will be saved automatically. If you want to save the current project under a different project name, select Save Workspace As from File menu. In the file selection box the new workspace name must be selected and confirmed.

3.3 Programming FLASH memories

3.3.1 Basic concept

UDE MemTool, a part of SPC5-UDE/STK (UDE), is intended to handle on-chip and external memory devices that do not permit direct and random write accesses unlike a RAM device permits. Typically, on-chip FLASH/OTP memory devices and external FLASH are of this type.

A target may contain several on-chip and external memory devices that can all be handled by MemTool. At a given time, only one device is activated. For each memory device, a Memory Device Handler inside UDE MemTool handles all accesses to the corresponding device. These Memory device handlers may be activated and deactivated individually.

Programming of the memory device is done by the Memory Device Driver which is a small application executed by the target MCU. MemTool uses functions provided by the SPC5-UDE/STK (UDE) target interface to load and run this driver application.

SPC5-UDE/STK (UDE) observes the download channel and activates the FLASH programming handling when it detects write accesses to the FLASH memory range. The FLASH memory settings are done already, when you use the default target configuration.

3.3.2 Enabling the FLASH programming

The UDE MemTool is an Add-In of SPC5-UDE/STK (UDE) starter kit version and must be activated. This is done via the Add-In Manager, menu Config - Add-in Components. Enable the entry UDE FLASH/OTP Memory Programming Tool.

When the UDE FLASH/OTP Memory Programming Tool is enable, a new menu entry is created in UDE MemTool menu Tools - FLASH programming. Open this dialog and the main front-end of MemTool will be opened. Choose the FLASH device and try to enable it. If all settings were correct, a list of FLASH sectors will displayed as shown below.

Figure 17. Enabling the FLASH programming
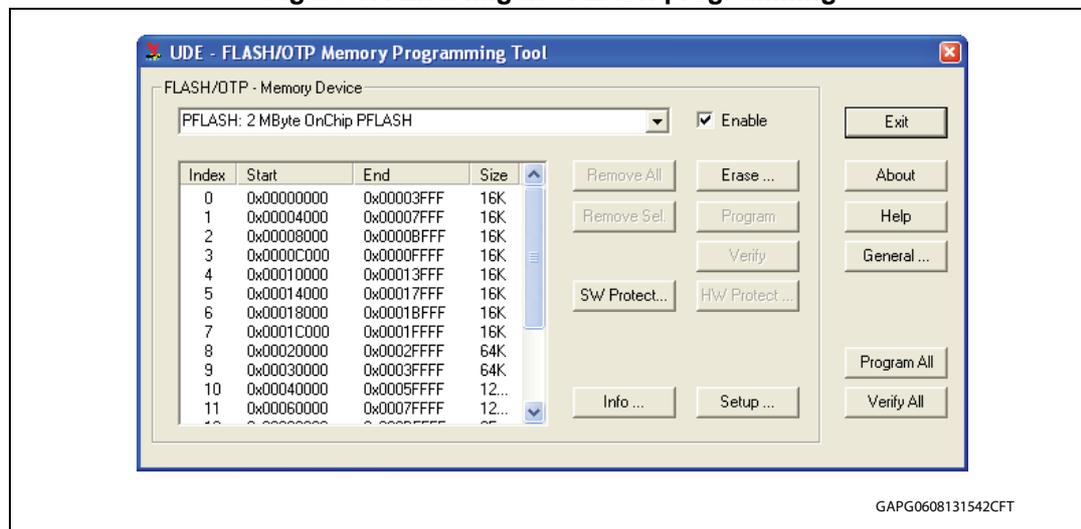
3.3.3 FLASH programming

If the UDE FLASH/OTP Memory Programming Tool is enabled, all registered FLASH devices are installed with special filters. These filters watch the download stream for address ranges met with a registered FLASH device. If the filters detect, that a code section is loaded, which is destined for the FLASH device, the code section will be marked for FLASH programming.

For the practical work it means, that you have to load the FLASH/ROM version of your program code File - Load Program only.

After loading the program code sections, the FLASH Programming Tool will open the main dialog and will offer the erasing, programming and verifying of code sections.

Figure 18. FLASH programming
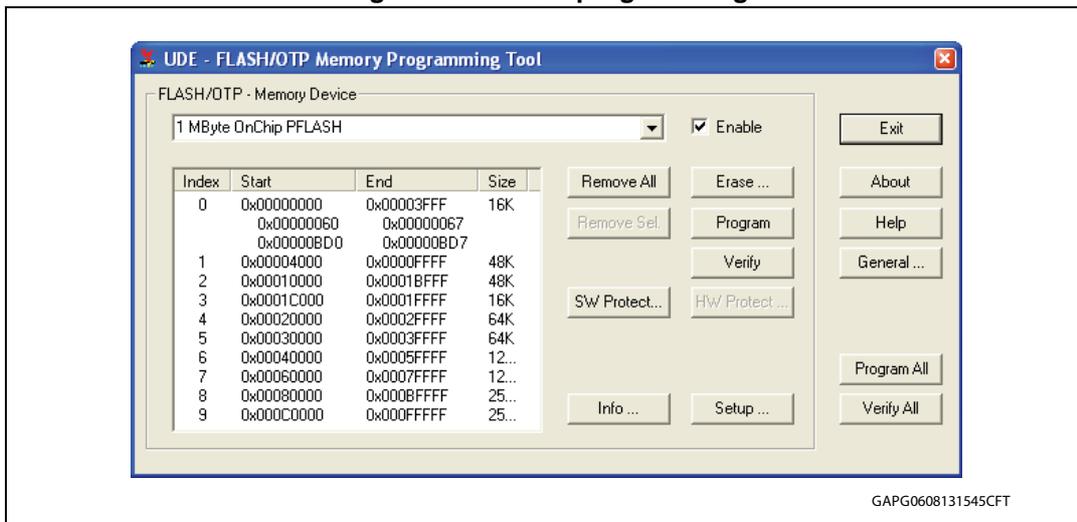
GAPG0608131545CFT

3.3.4 FLASH driver selection

The Power Architecture SPC56 microcontroller can run in VLE mode and standard Power Architecture mode. This results in different FLASH drivers. You can select the suitable driver to your requirements, please push the button Setup and select the driver from the list.

Figure 19. FLASH driver selection
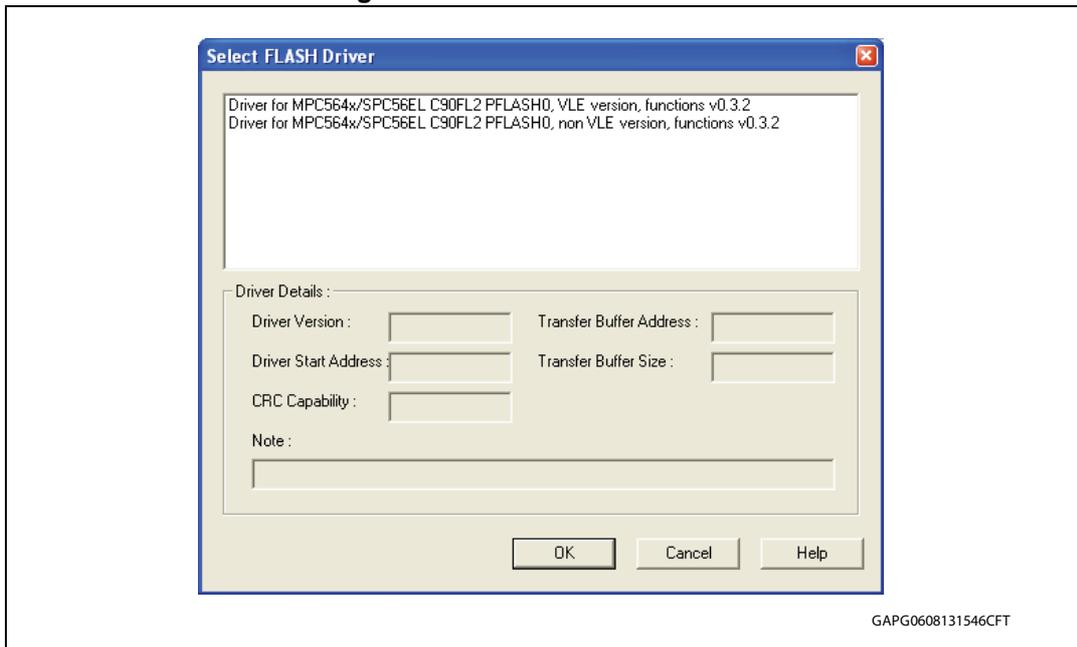
GAPG0608131546CFT

3.4 Help and support

If this getting started does not help to solve problems in detail, please contact
http://www.st.com/web/en/support/online_tech_support.html

4 Revision history

Table 1. Document revision history

Date	Revision	Changes
08-Aug-2013	1	Initial release.
17-Sep-2013	2	Updated Disclaimer.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER'S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR "AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL" INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.
Information in this document supersedes and replaces all information previously supplied.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com