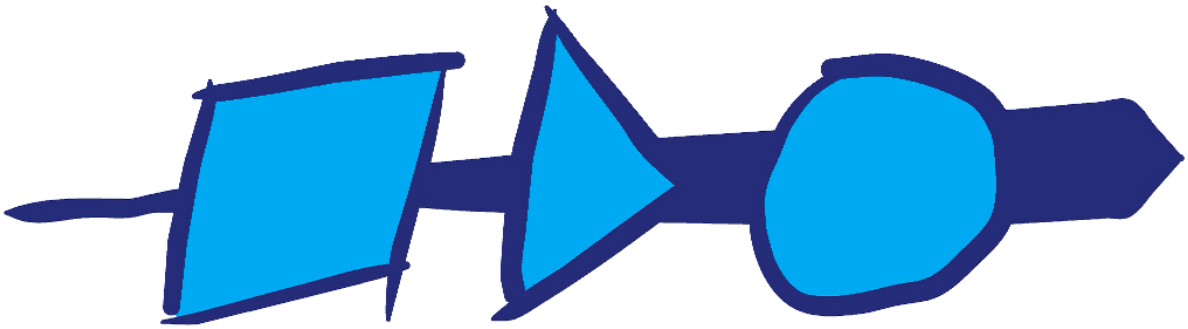
A Getting Started to

SPC5-UDESTK
Debugging with UDE

Integrated Development Environment for 32bit PowerArchitecture derivatives



universal debug engine®

pls Development Tools

This manual contains 23 pages.

Contact us at:

EMail: support@pls-mc.com
info@pls-mc.com
WWW: https://www.pls-mc.com

PLS Programmierbare Logik & Systeme GmbH
Technologiepark Lauta
D - 02991 Lauta
Germany
Phone: + 49 35722 384 - 0
Fax: + 49 35722 384 - 69

PLS Development Tools
19925 Stevens Creek Blvd
Cupertino, CA 95014
USA
+ 1 408 451 - 8408
+ 1 408 501 - 8808
Toll Free: + 1 877 - 77DEBUG

Contents

Introduction

Overview

This **Getting Started** document will help you to configure and install the Hardware and Software tools necessary to operate the SPC5-UDESTK starterkit version for SPC5 starterkit boards. 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-UDESTK starterkit version.
- Configure the starterkit board and connect it to the host PC.
- Set up a project.
- Debug a sample application.

If you need more information about the SPC5-UDESTK, 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**.

We wish you a lot of success with the SPC5-UDESTK starterkit version !

Feedback

The SPC5-UDESTK is a product of STM which is based on PLS' technology. The SPC5-UDESTK starterkit version is part of the SPC5-STUDIO. Regarding any comments about SPC5-UDESTK please use the website

http://www.st.com/web/en/support/online_tech_support.html

You are invited to browse to our Web site at **https://www.pls-mc.com/** to get the newest information about the professional and full version of **Universal Debug Engine® (UDE)**.

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

Getting Started

Before you start

To execute this **Getting Started**, it is necessary to have a

- Operating System Windows® 10, Windows® 11 64-bit
- PC equipped with USB port
- **A connection to the internet during installation for downloading and license activation**
- **Administrator** or **power user's rights** to install the required programs.

All hardware items below are included in the Starter Kit from STMicroelectronics:

- One of the following SPC5 starterkit boards equipped with a SPC5xx microcontroller:
  - Discovery kits SPC56, SPC57, SPC58
    - SPC560B_DIS, SPC560D-DIS, SPC560P-DISP
    - SPC560EL70L5DISP
    - SPC563M-DISP
    - SPC564A-DISP
    - SPC570S-DISP
    - SPC574K-DISP
    - SPC582xL1-DIS
  - SPC58EC-DISPPremium Evaluation board SPC56
    - SPC560B64A100S
    - SPC560BADPT144S, SPC560BADPT176S, SPC560BADPT64S, SPC560BADPT64S
    - SPC560PADPT100S, SPC560PADPT144S, SPC560PADPT176S, SPC560PADPT64S
    - SPC563M64A100S, SPC563M64A176S, SPC563MADPT100S, SPC563MADPT144S, SPC563MADPT176S
    - SPC564AADPT324S
    - SPC56EC74A176S, SPC56EC74A256S
    - SPC56ELADPT100S (Sphaero Cut2), SPC56ELADPT144S, SPC56ELADPT257S
  - Chinese boards
    - Bolero Board 2.0 with SPC560B40, SPC560B60 MCU
    - Monaco Board 1.4 with SPC563M64 MCU
    - Pictus Board 1.0 with SPC560P50 SPC56AP60 MCU
  - Premium Evaluation board SPC57
    - SPC570SADPT64S
    - SPC572LADPT80S, SPC572LADPT100S

        - SPC574KADPT144S, SPC574KADPT172F
        - SPC574SADPT100S, SPC574SADPT144S, SPC574SADPT244S
    - Premium Evaluation board SPC58
        - SPC58XXADPT64S (Chorus 1M Cut1, Cut2)
        - SPC58XXADPT64S (Chorus 4M Cut1)
        - SPC58XXADPT100S (Chorus 4M Cut1)
        - SPC58XXADPT144S (Chorus 1M Cut1, Cut2)
        - SPC58XCADPT292S (Chorus 4M Cut1)
        - SPC58XXADPT100S (Eiger Cut1/2, only Core2)
        - SPC5XXADPT292S (Eiger Cut1/2, only Core2)
        - SPC5XXADPT292S (Bernina Cut1, only Core2)
        - SPC584BADPT176S (Chorus 2M Cut1)
        - SPC58NHADPT386S (Chorus 10M Cut1)
- Power supply for the starterkit board.
- A mini-USB cable.
- **PLS USB/JTAG Adapter for SPC5xxx** with a 14-pin JTAG connector.

System Requirements

To run SPC5-UDESTK starterkit version at least the following minimum system configuration is required:

	Minimum	**Recommended**
CPU	Intel or AMD x86_64 (64-bit) processor	Intel Core i7™ or AMD R7 processor
RAM	4 GByte	8 GByte
Free disk space	3 GByte HDD	8 GByte SSD
Display	SXGA	WUXGA
Operating System	Windows®10 or Windows®11 64-bit	Windows® 10 or Windows®11 64-bit

Additional requirements:

- Optional: CD-, DVD- or BD-drive for installation from CD-ROM
- Microsoft .NET™ Framework 4
- Microsoft Internet Explorer® 10 or higher
- Adobe® Acrobat Reader 10 or higher
- Administrator permissions for the current login during installation.

The UDE Eclipse Integration Package **Eclipse 4.x UDEEclipse4Integration** requires following further environment:

- 64-bit version of Java JRE 8, JRE 11 or higher
- 64-bit version of Eclipse 4.8 - 4.23 or newer and the appropriate CDT package.

Downloading the latest SPC5-UDESTK version

The SPC5-UDESTK starterkit software is available for downloading from the following website

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

Installing SPC5-UDESTK

Please note that you must have administrator rights for successful execution of the installation process.

Run setup.exe and follow in the installing instructions. The SPC5-UDESTK starterkit version will be installed into the directory

```
C:\Program Files\pls\UDE Starterkit 2021\
```

Workspaces, target configurations and samples are saved in

```
C:\Users\<USER>\Documents\pls\UDESTK 2021\
```

License registering and activation of SPC5-UDESTK

The **starterkit version of SPC5-UDESTK** can be used for an evaluation purposes without registration. In this case, the SPC5-UDESTK is restricted to a limited code size for downloading of 128 kBytes.

The **registered version of SPC5-UDESTK** 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:

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

1. Company data.
2. Customer data.
3. Host PC's MAC address (MAC means the Media Access Control).
4. Serial number of PLS USB/JTAG Adapter for SPC5xxx (See the sticker backside of the adapter).
5. Activation code, delivered by STMicroelectronics.

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

The following further restrictions of the SPC5-UDESTK starterkit version exist in comparison to the PLS' **professional product Universal Debug Engine (UDE):** Useable for SPC5-UDESTK starterkit boards only, no visualization functions at runtime, no Script support, no separate flash programing tool.

Installing Hardware

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 **PLS USB/JTAG Adapter for SPC5xxx** and the user's target hardware can build up high voltages - over 10000 Volts (10 kVolts) 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:

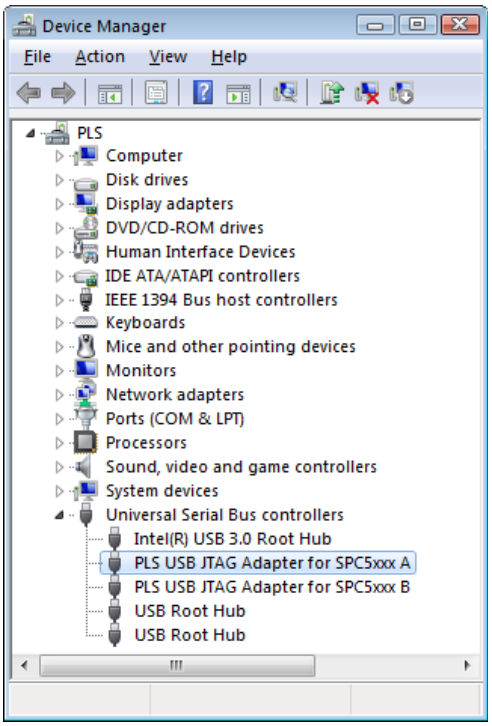
1. Please ensure that the **static electricity** and **ground potentials** between the **PLS USB/JTAG Adapter for SPC5xxx**, the host PC and the starterkit board are **balanced**.
2. Establish the target connection and **power on** the systems.

Driver Installation of PLS USB/JTAG Adapter for SPC5xxx

If the previous steps are done successfully, the SPC5-UDESTK starterkit version installed the hardware driver components automatically, when the **PLS USB/JTAG Adapter for SPC5xxx** is connecting to the host PC via the mini-USB cable the first time.

Connect the **PLS USB/JTAG Adapter for SPC5xxx** to the PC host system using the mini-USB cable.

The Windows device manager will find a new hardware device on your system called "**PLS USB/JTAG Adapter for SPC5xxx A**" in the "**Universal Serial Bus controllers**" group.



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:\Program Files\pls\UDE Starterkit 2021\Driver\JtagUsbDriver\InstallPlsUsbJtagDriver.bat`

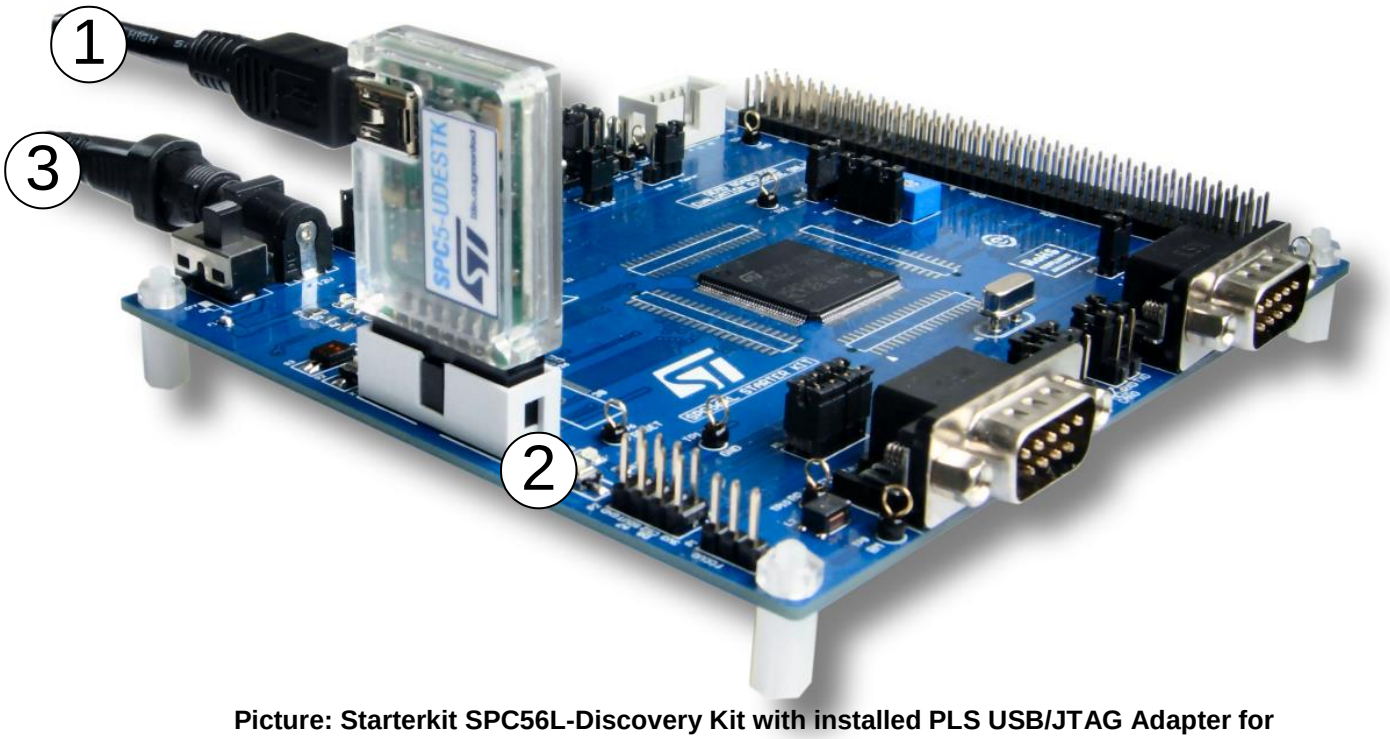
Alternatively, you can browse the driver directly, when Windows 7 is asking for that.

1. Connect the **PLS USB/JTAG Adapter for SPC5xxx** with your PC using the mini-USB cable. The Windows 10 system will find a new hardware device on your system called "PLS USB/JTAG Adapter for SPC5xxx" 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:\Program Files\pls\UDE Starterkit 2021\Driver\JtagUsbDriver\Driver\plsusbjtag.inf`
   driver.
5. Click **Next** and **Finish**.

Installing PLS USB/JTAG Adapter for SPC5xxx

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

1. Connect the **PLS USB/JTAG Adapter for SPC5xxx** with an USB port of the host PC by the mini-USB cable ①.
2. Plug the **PLS USB/JTAG Adapter for SPC5xxx** into the 14-pin JTAG debug connector of the SPC5 starterkit board ②.
3. Plug in the power supply of the starterkit board ③.
4. Start your computer system and login with administrator rights.

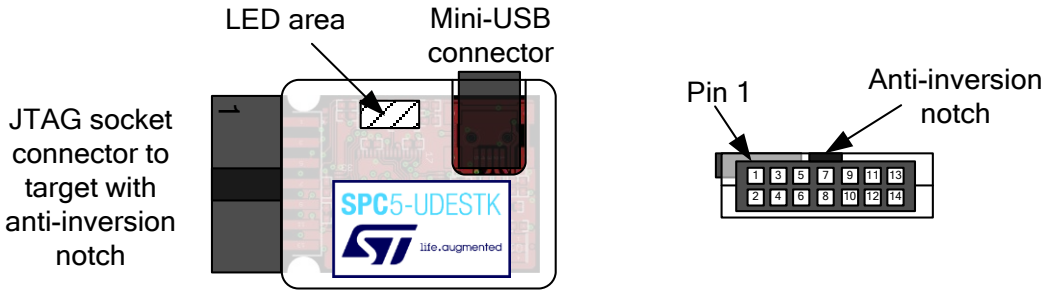


Picture: Starterkit SPC56L-Discovery Kit with installed PLS USB/JTAG Adapter for SPC5xxx

Technical details of PLS USB/JTAG Adapter for SPC5xxx

The **PLS USB/JTAG Adapter for SPC5xxx** is the Debug Adapter between the USB interface on the Host PC and the JTAG interface on the starterkit board.

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

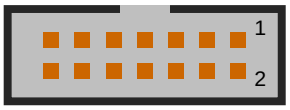


Picture: PLS USB/JTAG Adapter for SPC5xxx drawing

JTAG interface connector pinout description

JTAG/OnCE	Debugging Channel for the IEEE1149.1	up to 5 MHz

JTAG Debug Adapter for 100mil standard OnCE connector:



Pin 1	**TDI**	Pin 2	GND
Pin 3	**TDO**	Pin 4	GND
Pin 5	**TCK**	Pin 6	GND
Pin 7	n.c.	Pin 8	n.c.
Pin 9	**RESET#**	Pin 10	**TMS**
Pin 11	**V_{REF}**	Pin 12	n.c.
Pin 13	n.c.	Pin 14	**TRST#**

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.

User Manual

SPC5-STUDIO and SPC5-UDESTK (UDE) Integration

SPC5-STUDIO is a development environment containing compiler tools, editor and debug visualization and is based on the Eclipse platform. The SPC5-UDESTK startkerkit version can be integrated into the SPC5-STUDIO as debug plug-in.

Beside of this, the SPC-UDESTK (UDE) starterkit 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 in a stand-alone example.

A First Example with SPC56L-Discovery

We assume that you now have successfully installed the **SPC5-STUDIO**, the **SPC5-UDESTK** (UDE) starterkit version and the **PLS USB/JTAG Adapter for SPC5xxx**.

In this section you will learn about...

- how to start SPC5-UDESTK (UDE) starterkit version
- how to use the windows in SPC5-UDESTK (UDE) starterkit version
- how to load code into the starterkit board and start it as well as further principles how to debug an existing application.

We recommend you to go through this tutorial step-by-step. This example is shown under Windows 7 and using the **SPC56L-Discovery Kit** board offered by STMicroelectronics.

Starting with SPC5-UDESTK (UDE)

This Getting Started uses the example program Timedemo, prepared for GNU Compiler and SPC56L discovery kit, for demonstration the features of SPC5-UDESTK (UDE). Please download

`SPC5-UDESTK Manual and GettingStarted Timedemo_SPC56EL_VLE`

from **https://www.pls-mc.com/spc5-udestk** and unzip the package to

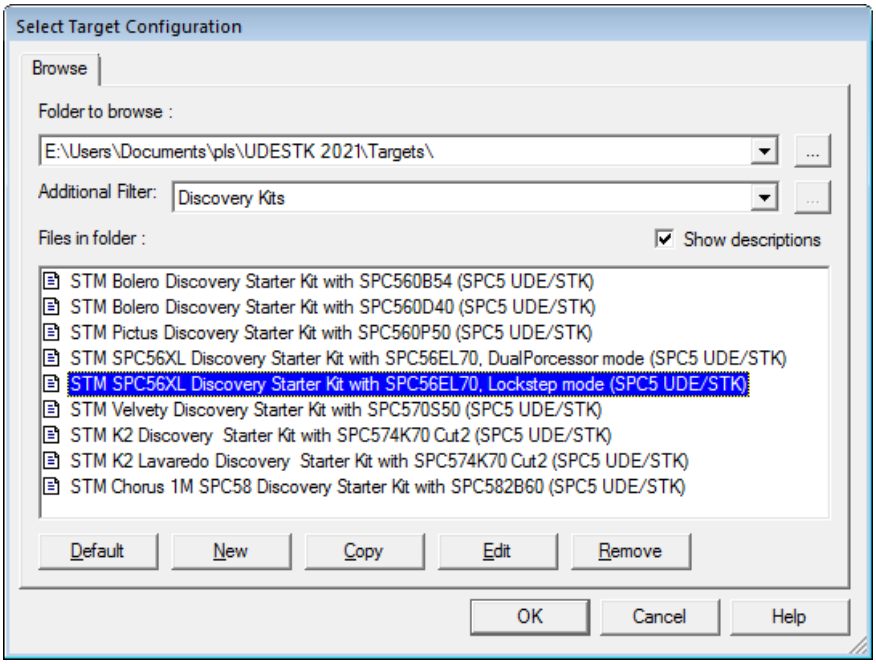
`C:\Users\<USER>\Documents\pls\UDESTK 2021\Samples\`

Double-click on the icon SPC5-UDESTK on the desktop. Alternatively, SPC5-UDESTK (UDE) may be launched also via **Start – Programs – UDE Starterkit Version 2021 – UDE Starterkit 2021**. This will start the desktop of the SPC5-UDESTK (UDE) development system.

The next step is creating a new workspace. An SPC5-UDESTK (UDE) workspace saves all configurations and settings of SPC5-UDESTK (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. `C:\Users\<USER>\Documents\pls\UDESTK 2021\Workspaces\SPC56L.wsx`.

After creating the new workspace, you will be asked to select a target hardware configuration. SPC5-UDESTK (UDE) comes with some default target configurations for several SPC5 starterkits. Select first from **Additional Filter** a suitable preselection, in this case Discovery Kits, and afterwards select the used board, in this case the Discovery Kit **STM SPC56XL Mini Module with SPC56EL70, Lockstep mode (SPC5 UDESTK)**.



Push **OK**.

The SPC5-UDESTK (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
```

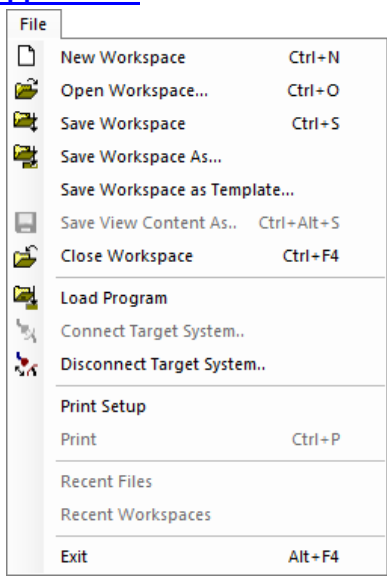
When launching SPC5-UDESTK (UDE) for the second time, you may use either **File - Open Workspace** or **File - Recent Workspaces** to select the workspace and start a new session with settings from the saved workspace.

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.

Loading an executable



After having SPC5-UDESTK (UDE) for PowerArchitecture SPC56L started, we want to load a program that can be executed on the SPC56L starterkit board.

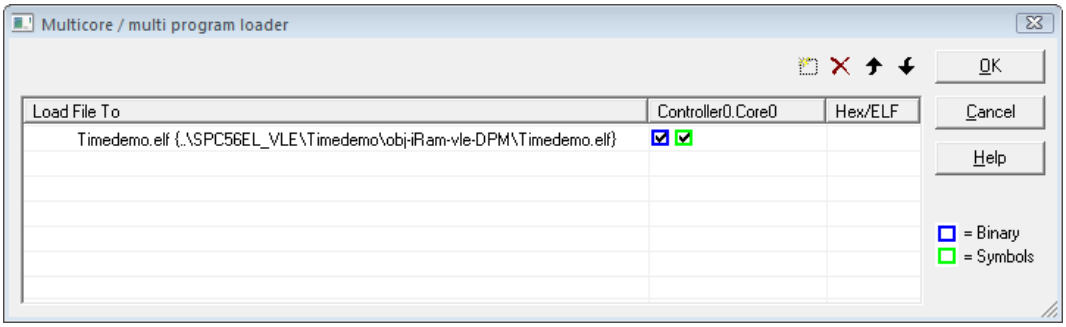
First, the GNU compiler variant of the `TimeDemo` example located for using in the internal sRAM should be used. This example is a basic demonstration and toggles only some LED of the SPC5 starterkit board.

The example is located to the internal sRAM of the SPC56L at memory address 0x4000'0000.

Select from the SPC5-UDESTK menu **Load Program** in the **File** menu bar, browse to

```
C:\Users\<USER>\Documents\pls\UDESTK
2021\Samples
\SPC56EL_VLE\Timedemo\obj-iRam-vle-LSM\Timedemo.elf
```

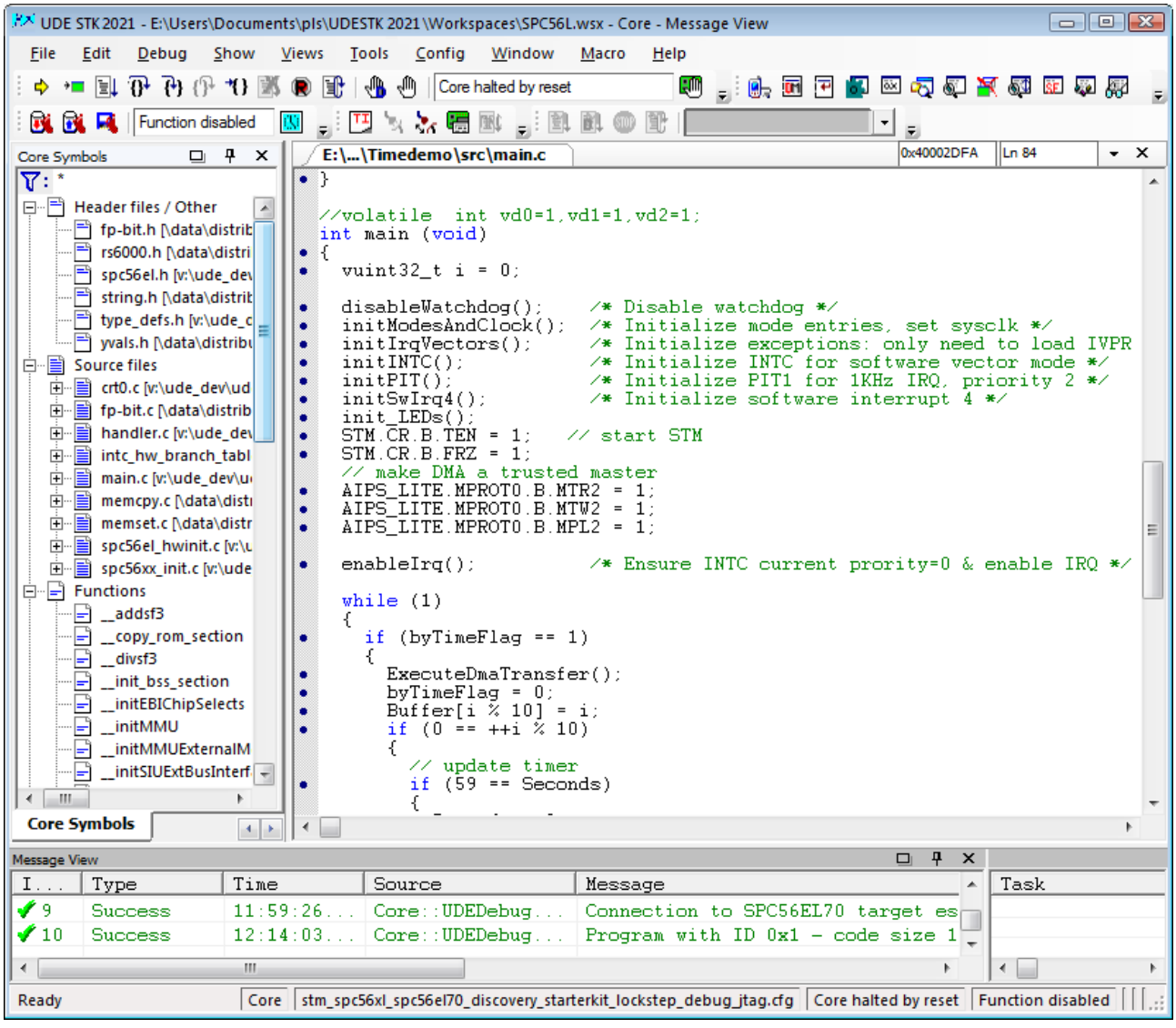
In the **Multicore / multi program loader** dialog select the settings **Binary** (blue) and **Symbol** (green) box. Both, binaries and symbols of the Controller.Core0 should be loaded.



After that, SPC5-UDESTK will ask for the location of the source files of the executable. Browse to

`C:\Users\<USER>\Documents\pls\UDESTK 2021\Samples\SPC56EL_VLE \Timedemo\src\main.c`

for the first question, click **Open** and repeat this if necessary. In the workbench, the Program window appears now showing the source code of the main function of the sample application.



Project management

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

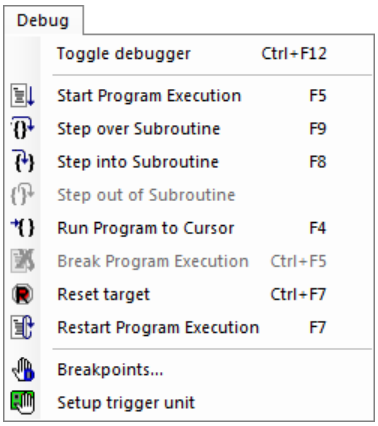
The SPC5-UDESTK (UDE) project contains source files, C/C++ functions, address sections and user-defined breakpoints. By double-clicking on one of the source files, the selected file will be brought into the Program window; by double-clicking on one of the procedures, the selected procedure is displayed.

When clicking with the right-hand mouse button into the Program window, a context menu appears to switch between Source code only and Source/Assembly code display via the **Mixed Mode** entry.

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

Please note: 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**.

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

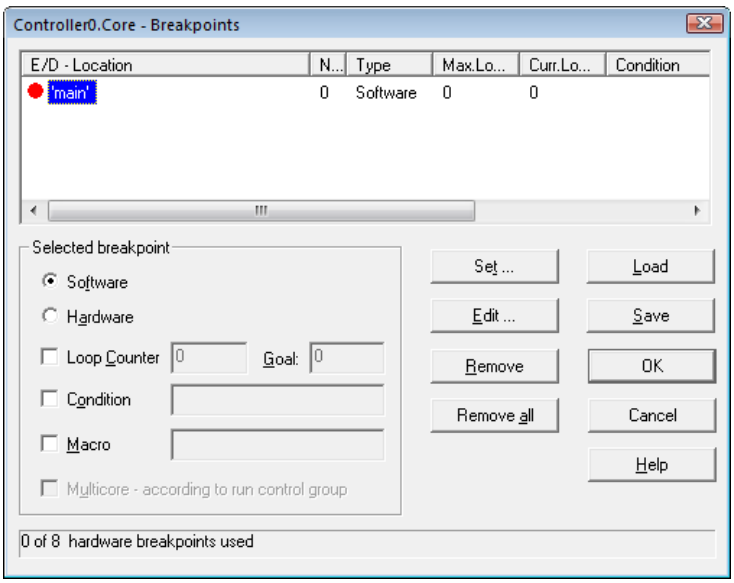
Click now onto the **Start Program Execution** entry or button and watch the LED on the SPC56L Discovery kit board flashing the TimeDemo code for UDE. When clicking onto **Break Program Execution**, the application is halted and the current instruction pointer position (code line) is displayed.

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.

Note: 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 finishes.

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.

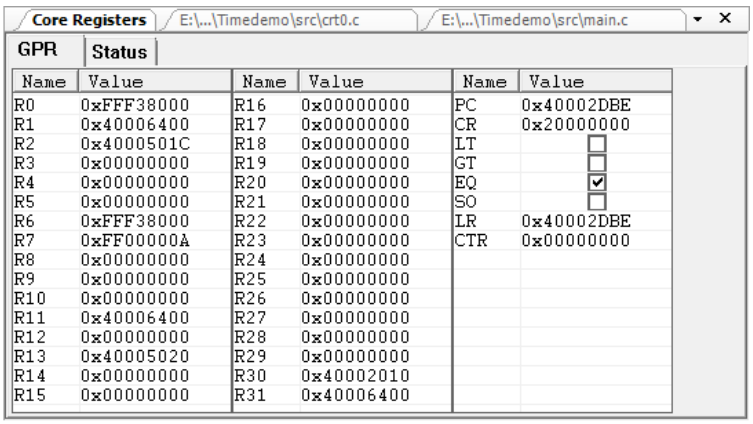
We want to set a breakpoint in the `Timedemo` program. To do this, click with the mouse on a line in a procedure and then click on the simple **Hand** symbol in the tool bar. A red-filled dot appears in the line indicating that the breakpoint has been successfully set. Alternatively, you may also select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

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.

Core Registers



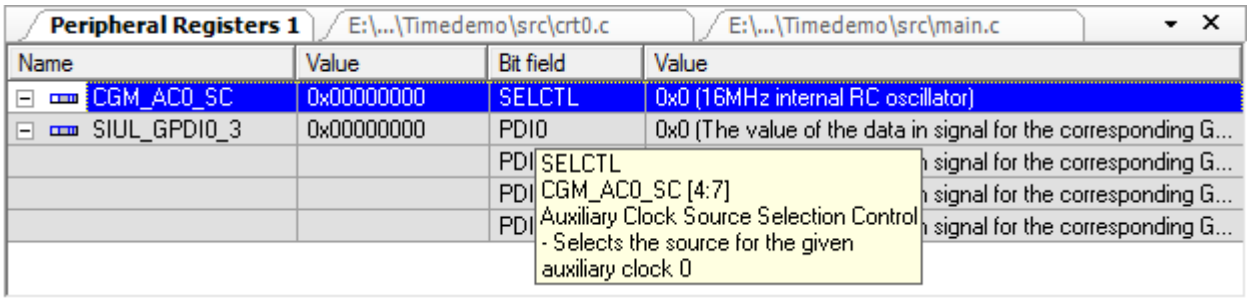
The CPU Register window is opened by the menu **Views - Core Registers** or the corresponding tool bar button.

Perform a few single steps to see the Core 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.

Peripheral Registers

Peripheral register values are changed in the same way as in the Core Register window (open this window by selecting the menu **Views – Peripheral Registers**). To add a new register entry, select **Browse Ins** from the context menu of the Peripheral Registers window and double-click on the Peripheral register from the list to insert it into the Peripheral Registers window.

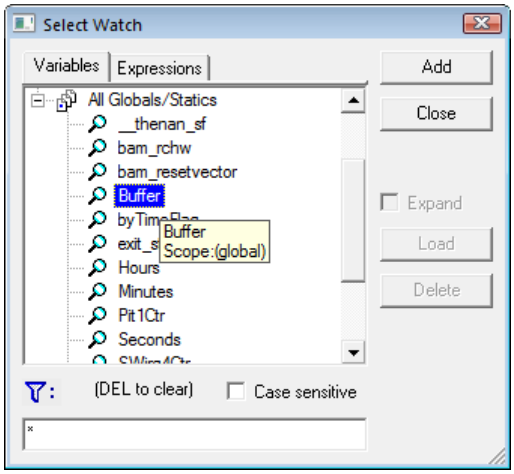


Tooltips show the address, the reset value of the current Peripheral Registers and further information about the focused register. The expanded Peripheral Registers view contains all available fields this register is composed off. The value can be changed by right-click on the values and entering the new value.

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

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 **Views - Watches** or the corresponding tool bar button.

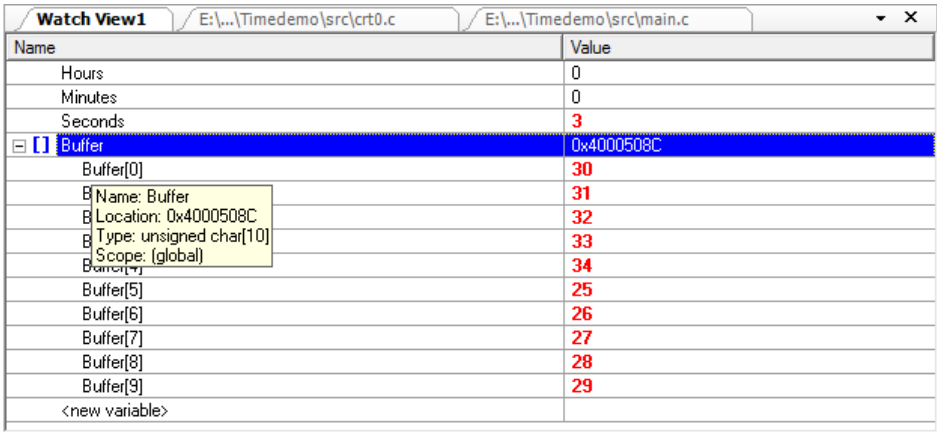
The variables can be added by double-clicking to <new variable> or using the context menu of the <new variable> entry via **Browse Ins**. The browser dialog shows you all available global and static variables. Click **Add** for adding a new variable to the watch window.

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-UDESTK (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.

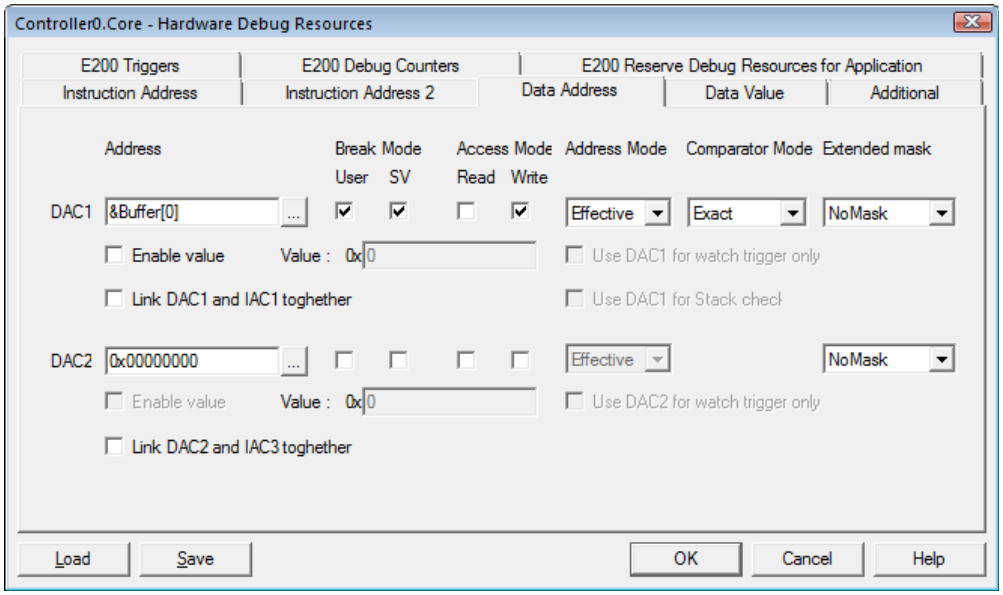
Viewing and changing local variables

Viewing local variables is provided by the Locals Window that can be reached via the menu **Views - Locals**. 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.

Trigger Functions

This chapter demonstrates how the PowerArchitecture Book E defined triggers of the SP56x derivatives can be used for debugging purposes. Again, the starting point is SPC5-UDESTK (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]`.



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: Core halted by internal breakpoint
```

inside of the message view.

Leaving the project

To leave the current Project select **File - Close Workspace** from the SPC5-UDESTK (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.

Programming FLASH memories

Basic concept

UDE MemTool, a part of SPC5-UDESTK (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 on the target MCU. MemTool uses functions provided by the SPC5-UDESTK (UDE) Target Interface to load and run this driver application.

SPC5-UDESTK (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.

Enabling the FLASH programming

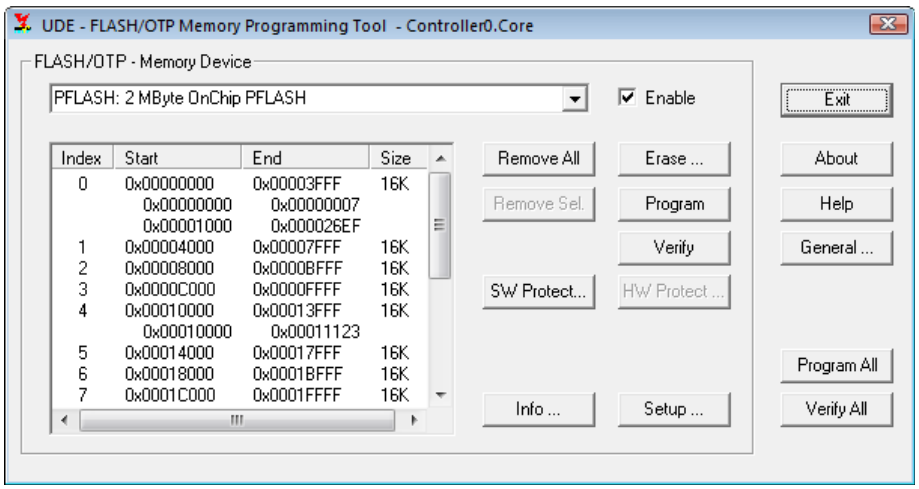
The UDE MemTool is an Add-In of SPC5-UDESTK (UDE) starterkit version and is be activated by default. This can be tested via menu **Config** - **Add-in Components**.

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.

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.

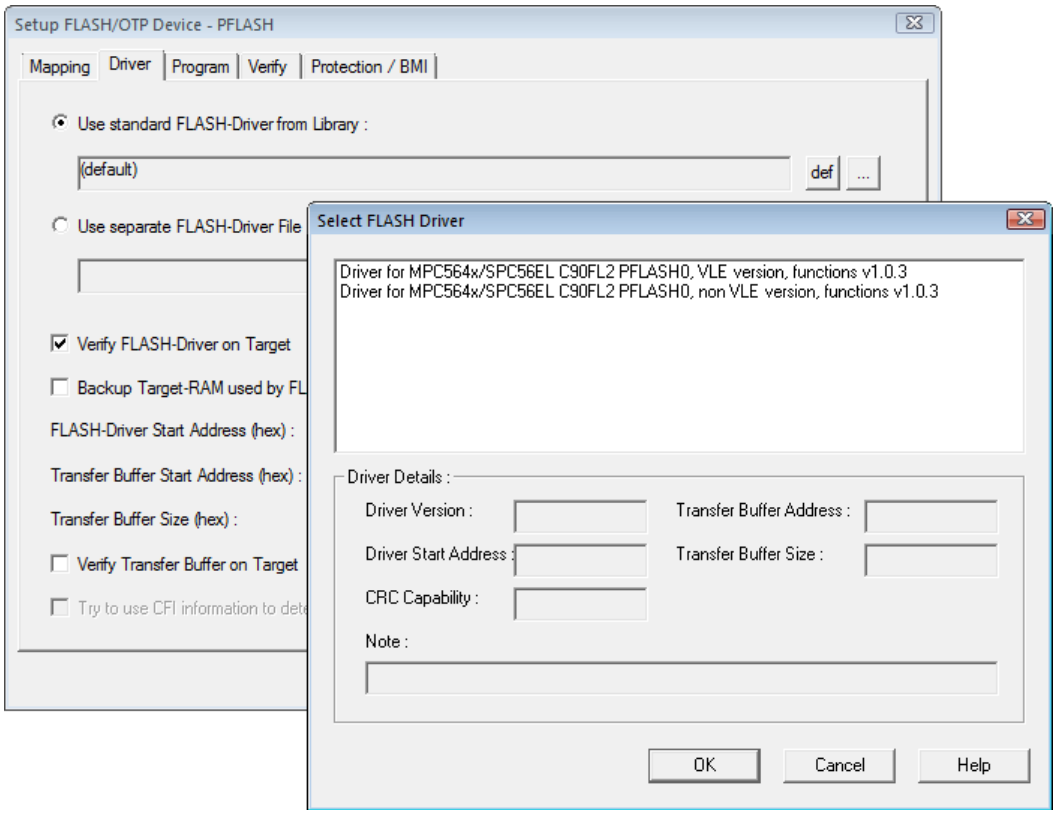
For the practical work it means, that you have to load the FLASH/ROM version of your program code via **File** - **Load Program** only.



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.

FLASH Driver Selection

The PowerArchitecture SPC56 microcontroller can run in VLE mode and standard PowerArchitecture mode. This results in different FLASH drivers. You can select the suitable driver to your requirements, please push the button **Setup ...**, select page **Driver**, use the **...** button from the selected driver and select the driver from the list.



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

Index