nohau
CORPORATION

Seehau™

Seehau Launcher (SLaunch)

March 11, 2003

Contents

About This Guide

Seehau is the debugger GUI (graphical user interface) from Nohau. It is used for all our current emulator/debugger families. This User Guide describes the Seehau Launcher (SLaunch) – which is intended to make it easier to integrate Seehau into third party development tools – such as code editors and IDEs (Integrated Development Environments).

1 USING THE SEEHAU LAUNCHER (SLaunch)

Introduction

Seehau is the debugger GUI (graphical user interface) from Nohau. It is used for all our current emulator/debugger families. The user normally starts Seehau like most Windows programs – by (double) clicking an icon. Then the user will load the program to debug by finding it on the hard drive, and the debug session can start.

The Seehau Launcher (SLaunch) intends to add a new method for starting Seehau – from within a code editor or an IDE (Integrated Development Environment). These are often customizable, and can be configured to start external applications when the user requests to debug. Examples include; ARM ADS, CodeWright, VIDE – (MicroCross GNU), Visual X-Tools, IAR Embedded Workbench.

SLaunch is necessary for this purpose. Without it, a new instance of Seehau would be started for each debug session (unless the user manually closes down the existing Seehau before starting the next session). Multiple instances of Seehau competing for the one hardware connection will not work correctly.

The operation of SLaunch is very simple, for this reason it does not have a user interface. There is no interaction with the user, except for error messages.

Operation

When started, SLaunch will test if Seehau is already active.

- If active, SLaunch will stop execution in the target.
- If not active, SLaunch will start Seehau.

Then SLaunch will cause Seehau to load code from the file(s) supplied on the command line.

SLaunch will terminate when load is completed.

SLaunch will start always the “Seehau.exe” that resides in the same directory as “SLaunch.exe” itself. This allows the versions of SLaunch and Seehau to be synchronized.

Command Line Options

All command line options that are given SLaunch are passed onto Seehau when Seehau is started, except for the following option that is used by SLaunch itself (note that the **two** forward slashes specifys that this option is used by SLaunch):

- **//fxFileName.** – File to load, where x specifies core number 0..9 for multi core systems. X can be left out for single core systems – i.e. //fFileName is accepted. This means that a filename cannot start with a digit. SLaunch translates all '/' to '\' in file names to deal with GNU tools better.

- **//go** – Start target execution after load.

- **//noreset** – Do not do target reset after load. For small changes, this actually allows continuing the debug session from where it was left of since data and registers are unchanged.

SLaunch ignores all entries on the command line that do not start with an '/' character. The reason being some development environments has the file name as first parameter, with no option to change it, but then allows customizing the rest of the command line.

Some users may want to specify the startup macro(s) file used by Seehau. This can be done with following command line option (see the help system for additional command line options):

- **/bStartup_Launch.bas** – Launch Seehau with a startup macro other than Startup.bas. The purpose of the startup macro is to configure Seehau. In the example here "Startup_Launch.bas" can be substituted for any file name. Use full path, unless the macro resides in the "Macro" directory.

Seehau Configuration

SLaunch does not require any special configuration in Seehau. However, the operation will be smoother if the automatic load of last file on startup is disabled. To disable automatic load, select menu "Config | Environment", uncheck "Load Code at Startup".

Error Handling

- When SLaunch first is started, make sure that one or more files were given on the command line, and that they exist. If an error occurs, SLaunch should be terminated with no further processing.

- Two instances of SLaunch cannot be run at the same time.

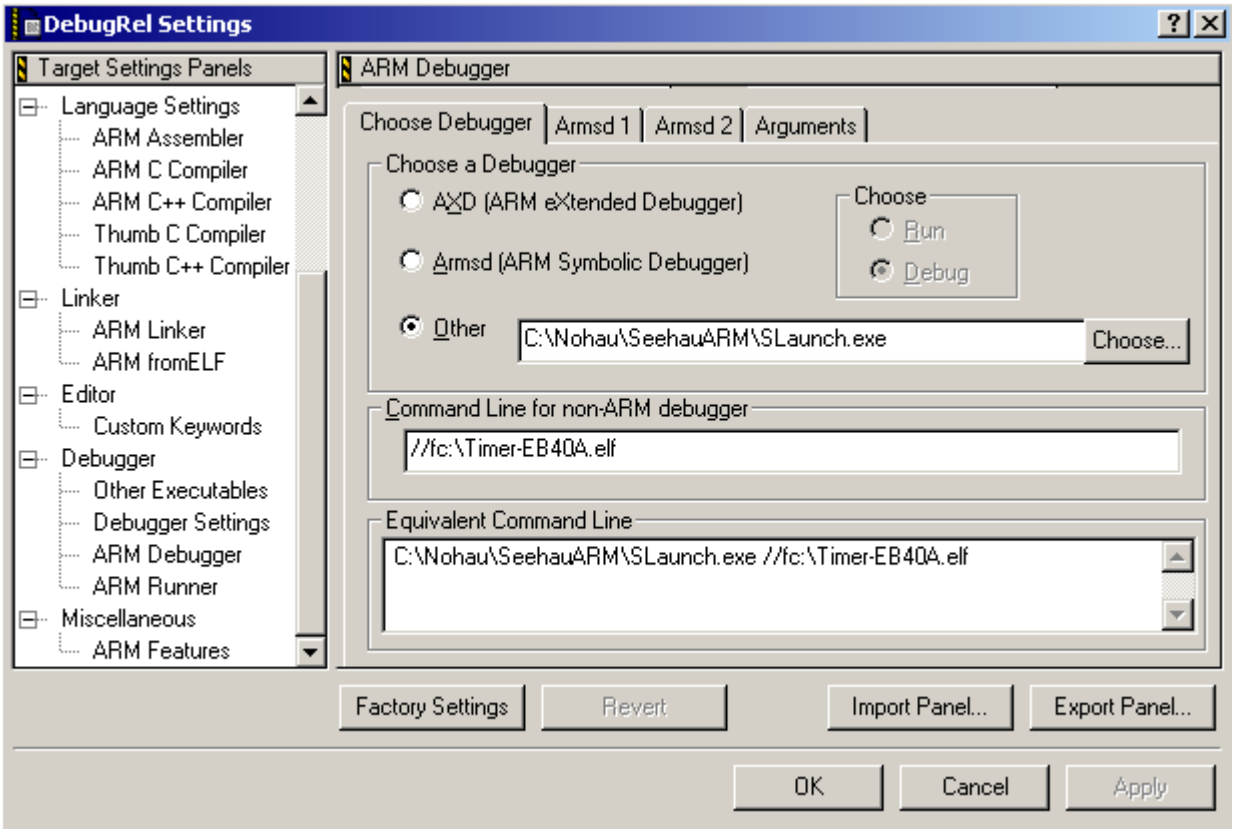
- SLaunch cannot distinguish between different Seehau families and versions. If there is already an instance of Seehau running from another family (i.e. HC12 instead of ARM etc), errors will occur. Similarly, problems are likely to occur if another version of Seehau is running (i.e. Seehau for same family, but with different build date).

2 EXAMPLES

This section shows how to use SLaunch to integrate Seehau into some popular IDEs (Integrated Development Environments). Note that SLaunch can be used with most IDEs – also those not shown here.

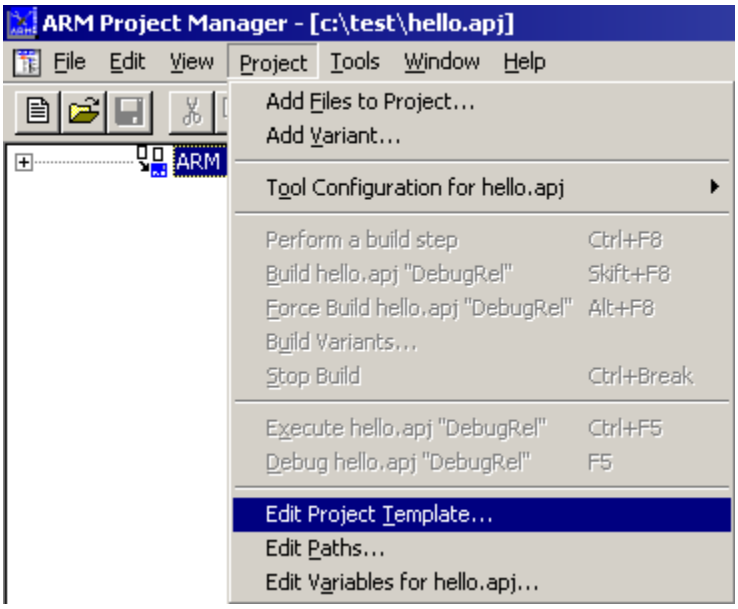
ARM ADS 1.2

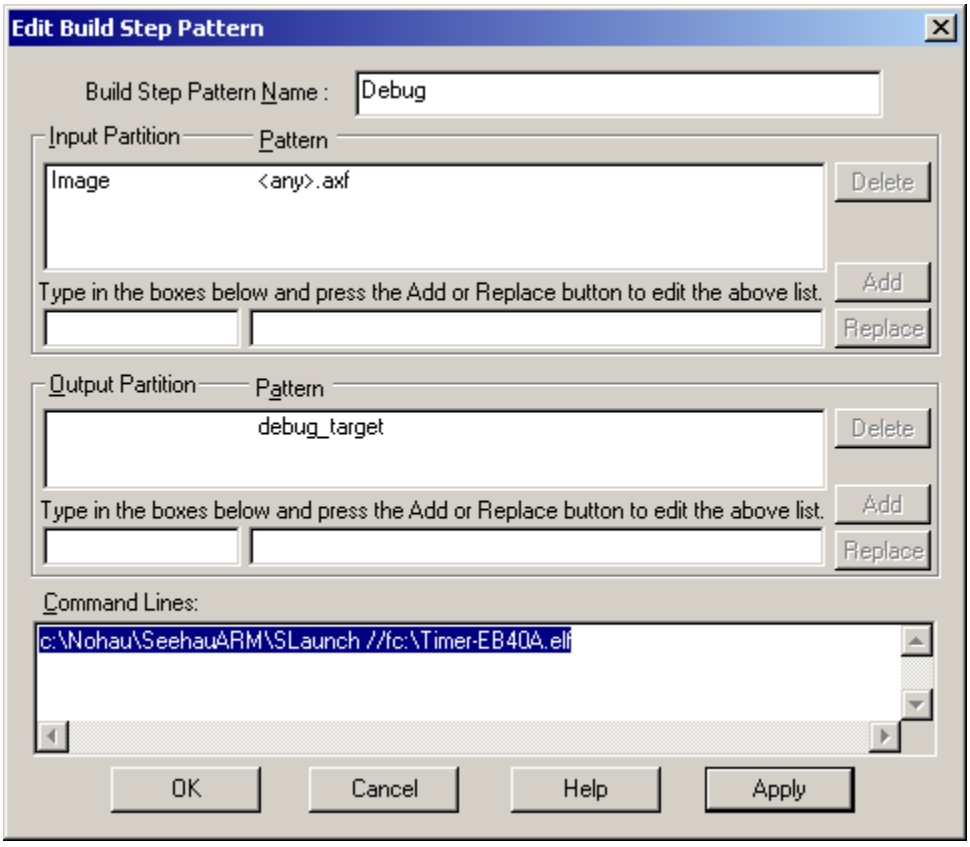
ARM ADS (ARM Developer Suite) has two commands to start the debugger – menu option "Menu | Debug" and "Menu | Run". Both of these can be customized to use a third party debugger. To do this, open the project settings. On the popup, select "Debugger | ARM Debugger" on the left pane, and enter data as shown below. For run select "Debugger | ARM Runner" and enter same information, and add "//go" on the command line options.



ARM SDT 2.5x

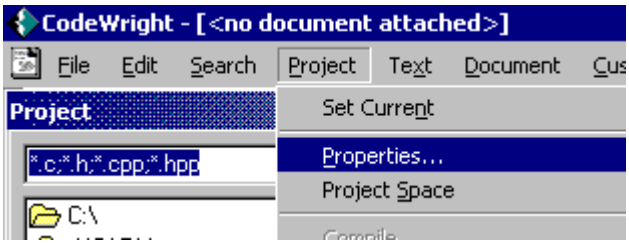
To configure the SDT ARM Project Manager (APM to launch an external debugger rather than ADW, change the launch command from APM by, selecting menu: "**Project | Edit Project Template**", then select "Debug" and press the "'Edit" button. Finally on the popup change the command line to "c:\Nohau\SeehauARM\SLaunch //fc:\Timer-EB40A.elf" or similar.



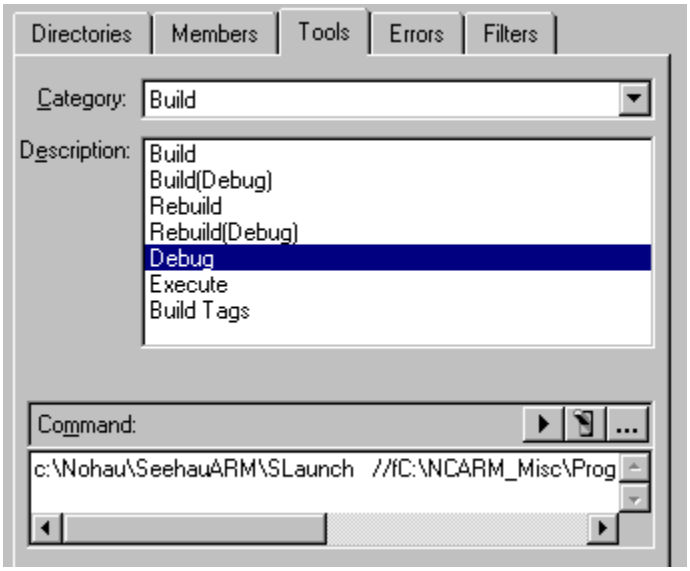


CodeWright

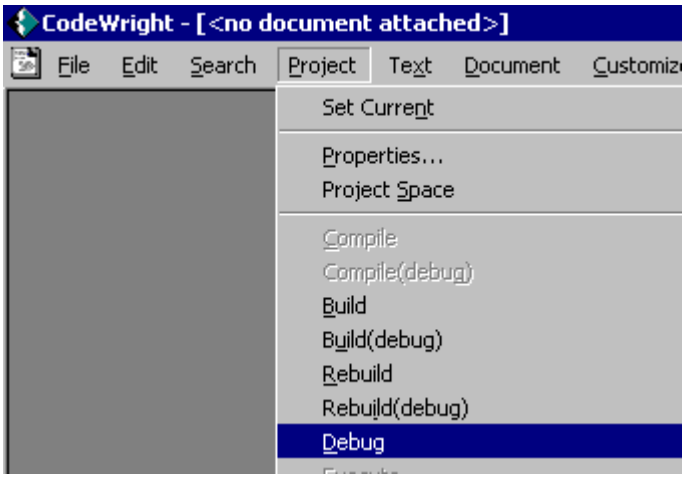
CodeWright is a popular code editor for the Windows environment. It is a complete IDE for code development – it can be customized for building, debugging and running programs. Select menu option “Project | Properties” to configure the project.



Then for the desired project enter information shown below. If desired, repeat for “Execute” with the “//go” command line option. Typical information could be c:\Nohau\SeehauARM\SLaunch //fC:\Timer-EB40A.elf

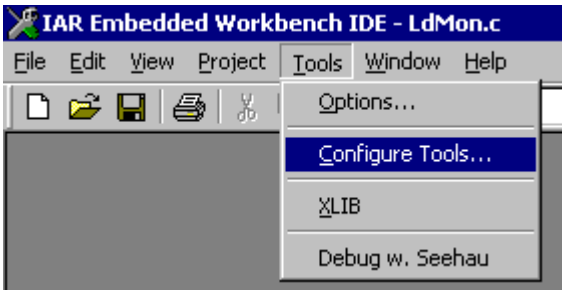


The debugger can now be invoked with the “Project | Debug “.

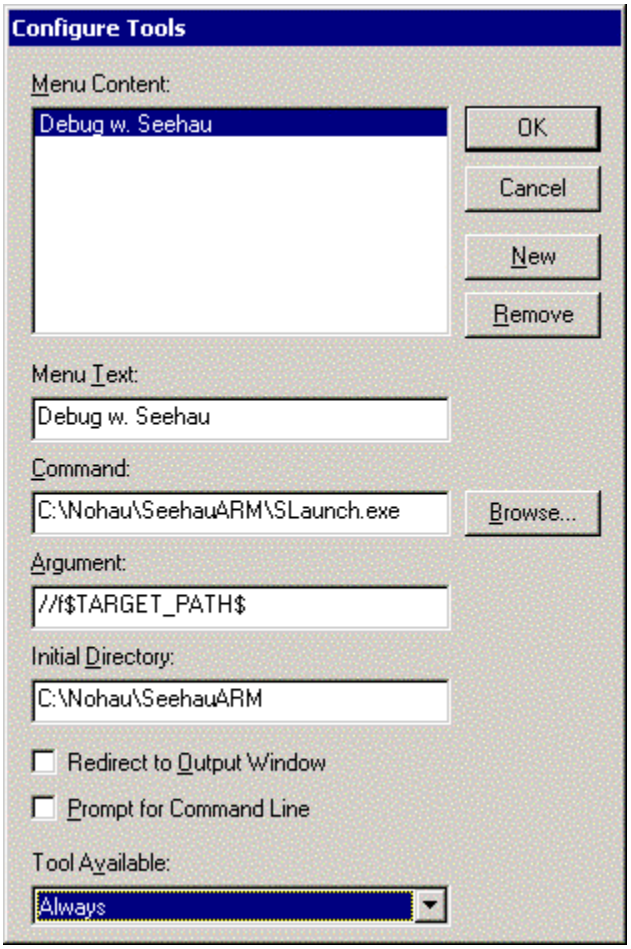


IAR Embedded Workbench

The IAR EWB (Embedded Workbench) has a tools menu that can be customized. To do this, select menu “Tools | Configure Tools…”. When added, Seehau is started using “Tools | Debug w. Seehau” as shown in the picture.



In the popup that follows, enter information as shown. Note that the macro $TARGET_PATH$ is built into the IAR EWB, and is expanded into the current output file including full path. Enter the “Argument” as: //f% TARGET_PATH %



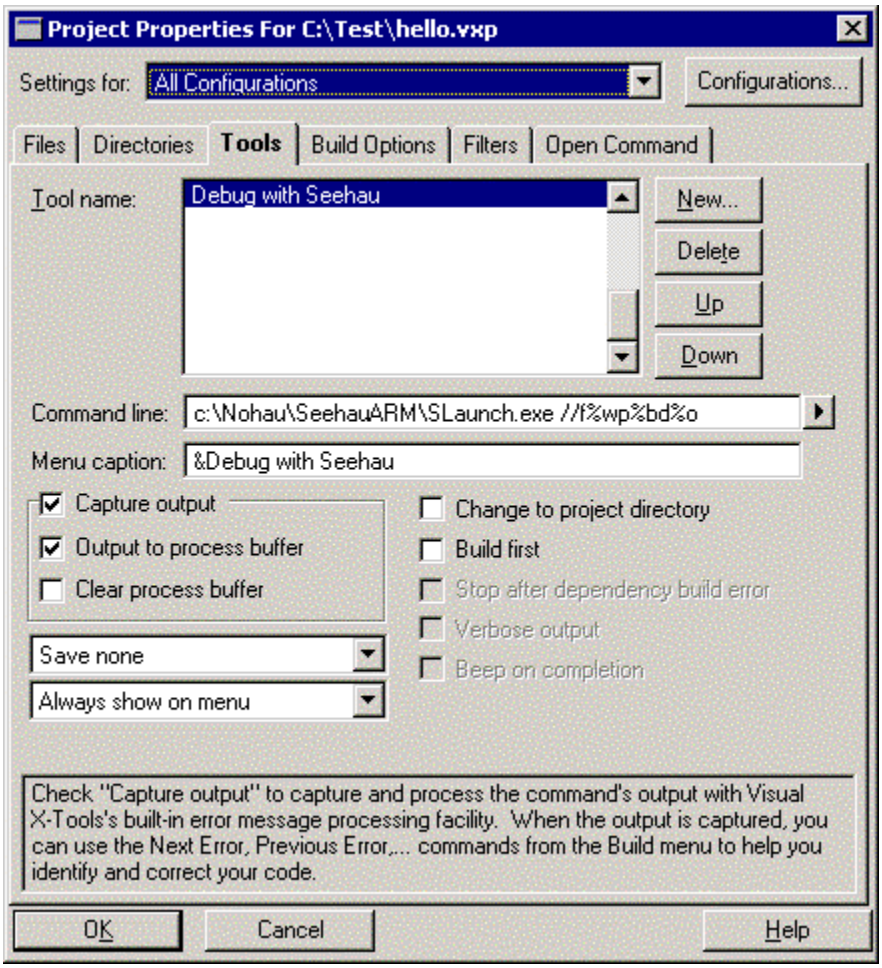
MicroCross Visual X-Tools (SlickEdit)

Visual X-Tools from MicroCross is an IDE for the GNU compiler supplied by MicroCross. It is based on the SlickEdit editor (http://www.slickedit.com/). So the procedure for Slicked should be similar. We have tested two different methods to integrate Seehau into Visual X-Tools as described in following sections.

MicroCross Visual X-Tools - Adding New Menu Option

Visual X-tools allows configuring the Debug menu per project. To do this for one project, select menu "**Project | Project Properties...**", and click on the "Tools" Tab, then click "New…". Enter the menu option to create – i.e. "Debug with Seehau" – click "OK" and you will get something like in the picture below.

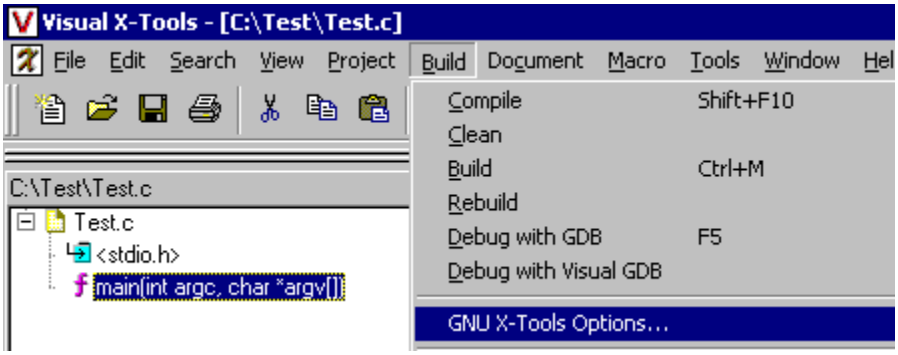
Note the ">" button on the command line entry field below – clicking it shows a list of macros that can be used to specify debug file. (Note that SLaunch translates all '/' to '\' in file names.) Below, we have entered command line: c:\Nohau\SeehauARM\SLaunch.exe //f%wp%bd%o



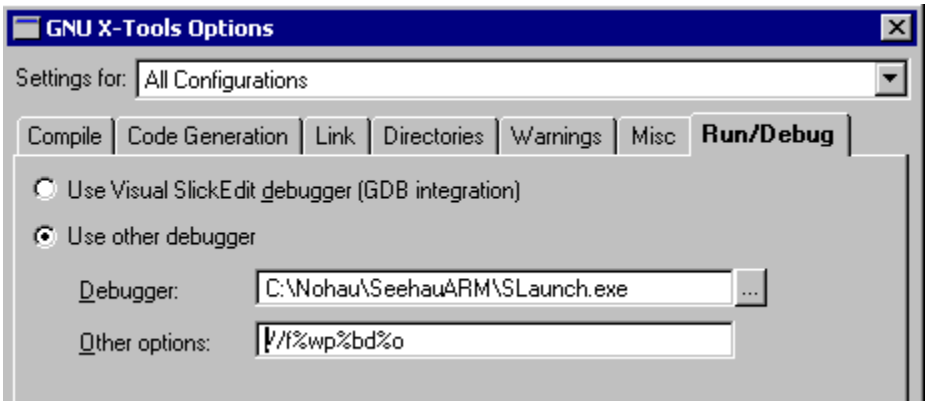
MicroCross Visual X-Tools – Modifying Existing Menu Option

Visual X-Tools has integrated support for GDB (GNU Debugger). If the debug button or "F5" key is pressed, GDB will start. This can be modified so that Seehau is started instead. (However, the menu option would still say, "Debug with GDB".)

To do this, open "**Build | GNU X-Tools Options...**" (Can also be accessed through "Project | Project Properties...",)



Then on the "Run/Debug" tab enter something like shown below. Note that the resulting command line will include the load by default, which means that it will appear twice. This is no problem since SLaunch ignores all command line parameters without



For a description of %wp etc, see previous section.