UserGuide QSPI-Flasher-speedup

|  |  |
| --- | --- |
| Document Control: | |
| Project: | MFC4T0, Toyota Mono |
| Revision: | 1.6 |
| Last Change: | 23.03.2015 |
| Confidence Level: | 🞏 Public 🗷 Confidential |

Printed copy is uncontrolled!  
  
CONFIDENTIAL AND PROPRIETARY PROPERTY OF A.D.C. GmbH - ALL RIGHTS RESERVED

|  |  |
| --- | --- |
| Document State: | |
| State: | Draft |
| Author: | Harald Heim-Ext, Shinjith Unithrattil, Magnus Baur |
| Reviewed by: | - |
| Released by: |  |

Table of Contents

[1 Introduction 3](#_Toc409472594)

[2 Change Log 3](#_Toc409472595)

[3 Userguide 4](#_Toc409472596)

[3.1 Target Configuration Set Up 4](#_Toc409472597)

[3.1.1 Fetch Updated Configuration 4](#_Toc409472598)

[3.1.2 Import target Configuration 4](#_Toc409472599)

[3.1.3 Launch target Configuration 5](#_Toc409472600)

[3.2 Script mode 6](#_Toc409472601)

[3.2.1 Pre-requisite 6](#_Toc409472602)

[3.2.2 Installation of Script 6](#_Toc409472603)

[3.2.3 Using Script 6](#_Toc409472604)

[3.3 Manual Mode 8](#_Toc409472605)

[4 Known Issues 12](#_Toc409472606)

# Introduction

This document is the user guide of the reworked high-speed VH28 QSPI-Flash writer. The speed of file to RAM transfer has been 10times increased.  
Flashing needed 15min in the former, now is done within 1½min.

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Editor** | **Description** | **Pages /**  **Chapter** |
| 1.1 | 17.10.2013 | Heim | Initial revision | All |
| 1.2 | 23.10.2013 | Heim | Wipe SBL and Script mode added | All |
| 1.3 | 24.10.2013 | Heim | Guide to adapt your own Target Configuration, wrong Position of labels m30 and m31 corrected. | 3.1.1 |
| 1.4 | 19.01.2015 | Shinjith | Changes done according to the new directory structure and flashing procedure | All |
| 1.5 | 11.02.2015 | Heim | To copy DRA7xx.xml added | 3.1.1 |
| 1.6 | 23.03.2015 | Baur | To copy DRA7xx folder added | 3.1.1 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Userguide

The package is found in the folder

{yoursandbox}\S00\_ALL\04\_Engineering\02\_Development\_Tools\ti\_Tools\qspi-flash-writer-reloaded\

Within the folder you find now a complete CCS-Project with source-files (see src\).

The old main.c and so on has been dropped.

* Start CCS and Import existing project

## Target Configuration Set Up

First step is to get the latest target configuration and gel files from sandbox

### Fetch Updated Configuration

Copy the {yoursandbox} S00\_ALL\04\_Engineering\03\_Workspace\sw\target\_configs\Modules\DRA7xx Folder to {CCS\_INSTALL\_DIR}\ccsv6\ccs\_base\common\targetdb\Modules\

Copy {yoursandbox}\S00\_ALL\04\_Engineering\03\_Workspace\sw\target\_configs\vhe\_single\_minigel\_usb560.ccxml

To your CCS installation

{CCS\_INSTALL\_DIR}\ccsv5\ccs\_base\emulation\gel\

Copy complete folder {yoursandbox}\S00\_ALL\04\_Engineering\03\_Workspace\sw\target\_configs\gel\_files

To your CCS installation

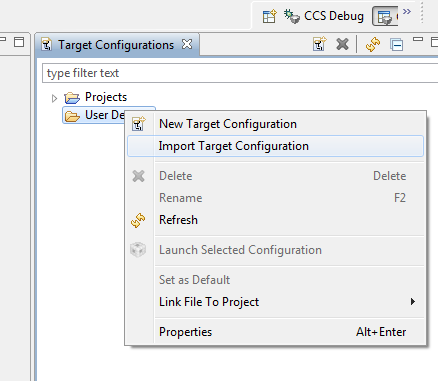
{CCS\_INSTALL\_DIR}\ccsv5\ccs\_base\emulation\gel\

also copy the xml-file {yoursandbox}\S00\_ALL\04\_Engineering\03\_Workspace\sw\target\_configs\devices\**DRA7xx.xml**To your CCS installation {CCS\_INSTALL\_DIR}\\ccsv5\ccs\_base\common\targetdb\devices\

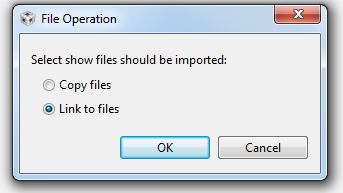
### Import target Configuration

Right click and import target configuration “vhe\_single\_minigel\_usb560.ccxml” from below path

{CCS\_INSTALL\_DIR}\ccsv5\ccs\_base\emulation\gel\vhe\_single\_minigel\_usb560.ccxml

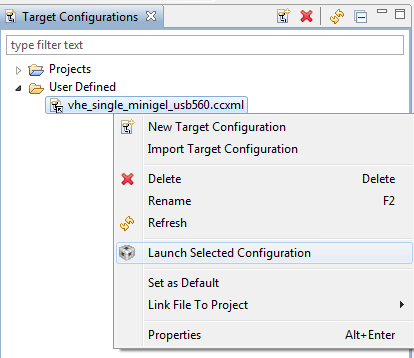


Select Link to files and ok

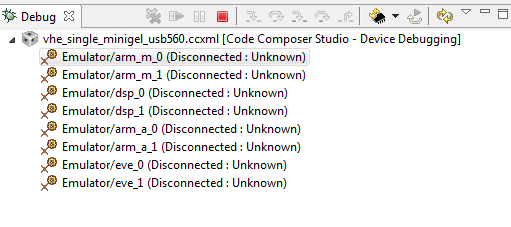


### Launch target Configuration

Right click on \*.ccxml file and launch



It may take a while during first launch. Following debug window or similar should pop up after successful launch



## Script mode

Automated flashing can be done using Debug server scripting. This flashes the combined image “soc\_allcores.bin” generated by the build system, which is available in below location after successful build

{yoursandbox}\S00\_ALL\04\_Engineering\04\_Build\mot\soc\_allcores.bin

### Pre-requisite

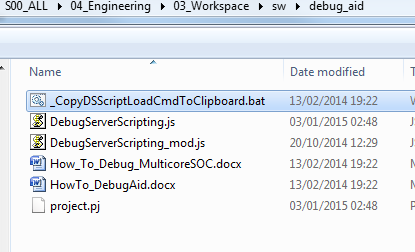
Launch appropriate target configuration “vhe\_single\_minigel\_usb560.ccxml”. Refer section 3.1 for detailed steps

### Installation of Script

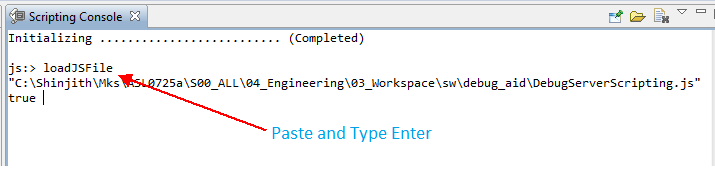
Browse to Scripts which is available in below sandbox location

{yoursandbox}\S00\_ALL\04\_Engineering\03\_Workspace\sw\debug\_aid

Double click on the \_CopyDSScriptLoadCmdToClipboard.bat file which copies the command and script path



Go to Scripting console and paste it



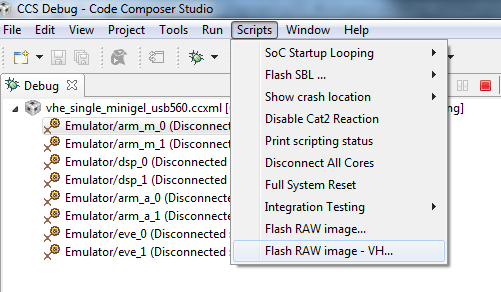
Now script is ready to use

### Using Script

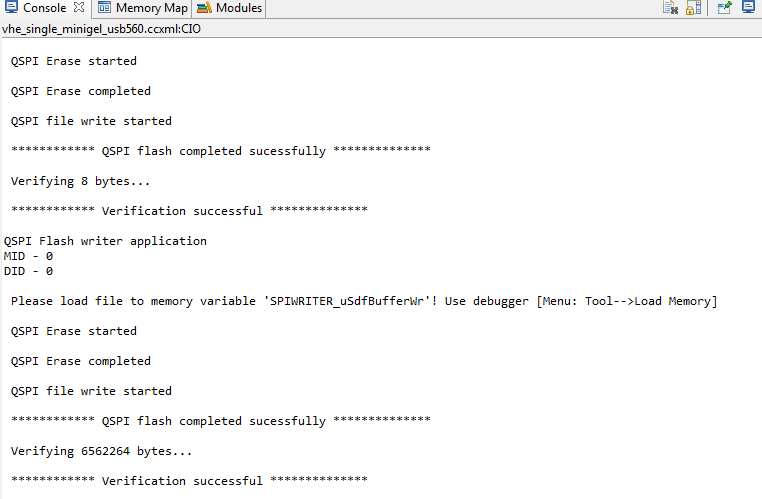
#### Flashing Raw Image

Power ON the ECU and run the script menu Flash RAW image - VH ...

Script flashes below binary file generated from the build system automatically. {yoursandbox}\S00\_ALL\04\_Engineering\04\_Build\mot\soc\_allcores.bin



Watch out the console for flashing progress and wait until verification success message appears in the std-out console window.



#### Not Supported

Other script features are not supported at the moment

## Manual Mode

Step 1 :

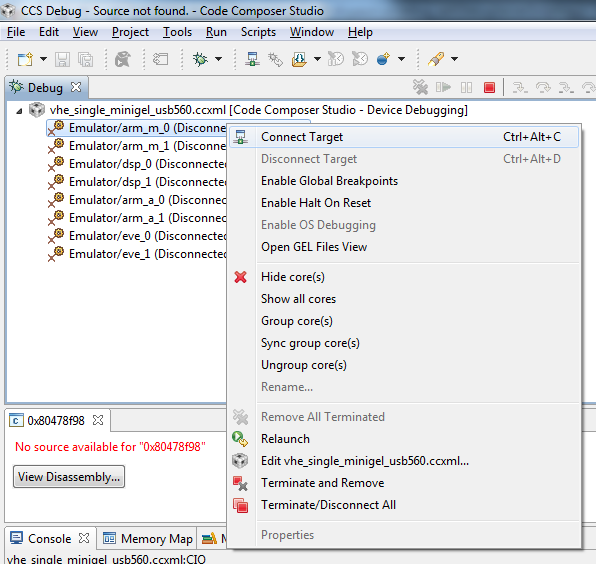
Launch valid target configuration from the below path

{CCS\_INSTALL\_DIR}\ccsv5\ccs\_base\emulation\gel\vhe\_single\_minigel\_usb560.ccxml

Step2 :

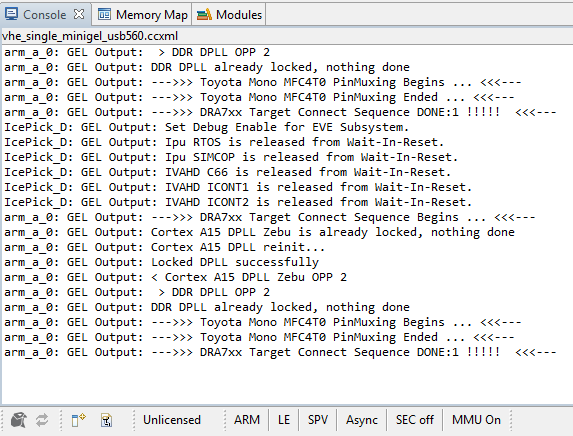
Right click and connect arm\_m\_0 core. This is required to prevent from SOC to execute the already flashed image which causes trouble during flash.

This step may fail if either SOC doesn’t have valid image or core is not out of reset. Both cases nothing to worry and should proceed to step3



Step3 :

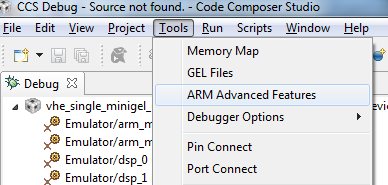
Similarly right click and connect arm\_a\_0 core. This core is intended to run the flash utility, which runs the gel file and do the basic initialization required for flashing. After successful execution of gel following messages should appear on console window



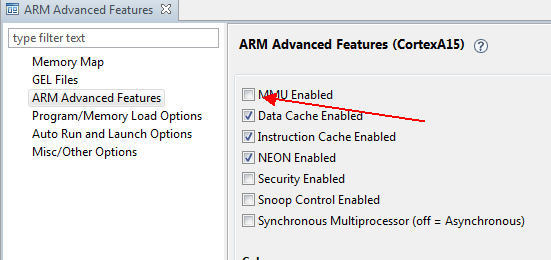
If any error pops out during gel execution flashing cannot be continued. Should reset the board and try again.

Step4 :

Turn off the MMU switch if it’s already enabled, otherwise .out loading may fail. MMU option can be switched off using Arm Advanced Feature options available in CCS tools menu



Un-tick the below option



Step5 :

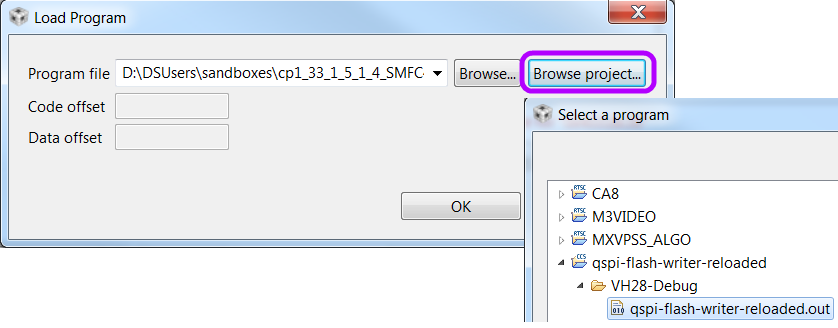
Load the qspi-flash-writer-reloaded.out.

Pre-built binary is available in sandbox

{yoursandbox}\S00\_ALL\04\_Engineering\02\_Development\_Tools\ti\_tools\qspi-flash-writer-reloaded\VH28\_Debug\qspi-flash-writer-reloaded.out

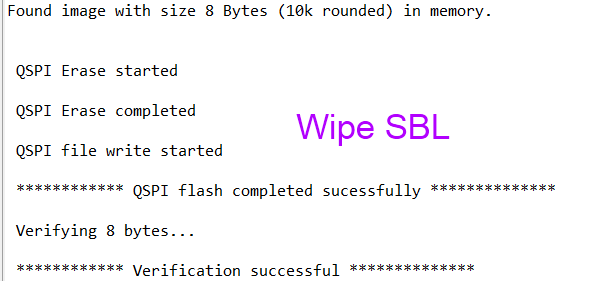
Alternately import and build the CCS project from below sand box path to get the proper symbols for breakpoint

{yoursandbox}\S00\_ALL\04\_Engineering\02\_Development\_Tools\ti\_Tools\qspi-flash-writer-reloaded\



Step6 :

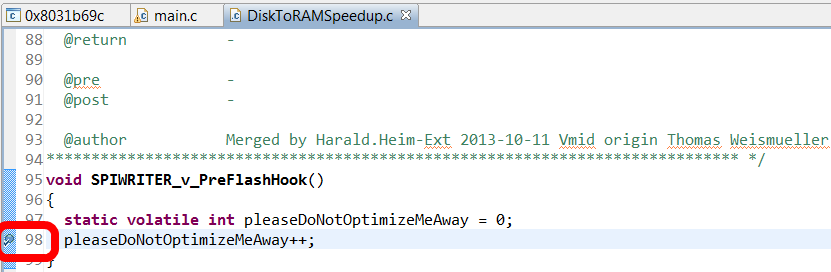
Wipe the SBL - Run the Program without breakpoints, and wait until verification successful.  

Step7 :

Halt the target, if it’s not in halted state (Press Alt+F8), then do a restart, which again brings application back to main() function  

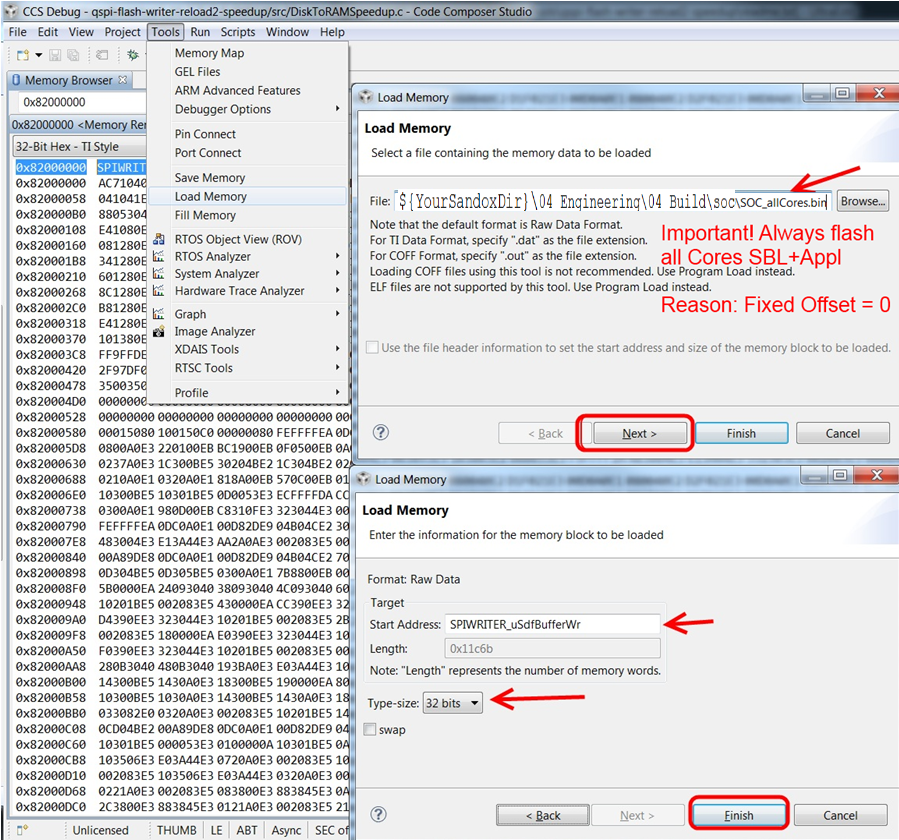

Step8 :

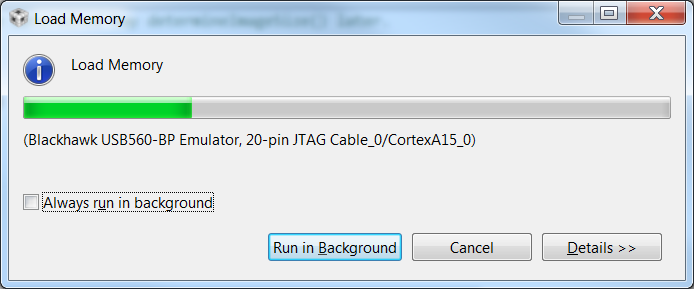
Open source file DiskToRAMSpeedup.c and set breakpoint in SPIWRITER\_v\_PreFlashHook() function  


Step9 :

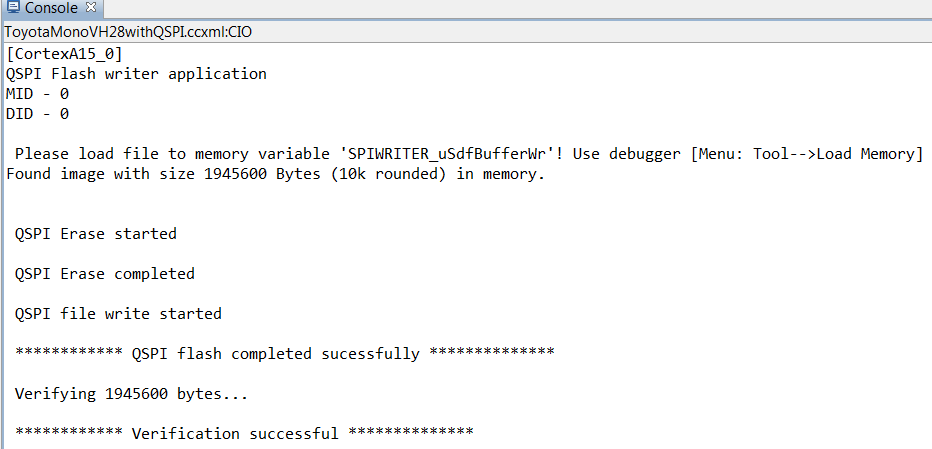
Run target again press F8, which halts at above set breakpoint  


Step10 :

Load {yoursandbox}\S00\_ALL\04\_Engineering\04\_Build\mot\soc\_allcores.bin file to memory variable 'SPIWRITER\_uSdfBufferWr'!   
Use debugger [Menu: Tool-->Load Memory]:  




Step11 :

After loading into memory run the target again press F8  
  
  
Watch out the std-out console window and wait until verification is successful  


# Known Issues

* Target should be connected as early as possible once board is powered ON. Target connection may fail once already existing application in flash is up and running.
* If release build application already exists in flash, board boots up faster and chances of failure is more