Flash Bootloader

Delivery Information - CBD1500635 D03

Version 1.1

Authors	Joe Benczarski
Status	Released

Contents

1 General

Check **DeliveryDescription_CBD1500635.html** Documentation.

> **i** **Note**
> Documentation concerning the Flash Bootloader can be found in the **Doc** folder of your delivery.

File	Description
2493.2_ADD_Nexteer_FBL_GM_SLP6_Renesas RH850-CBD1500635.D03.pdf	Legal document
DeliveryDescription_CBD1500635.html	Version information for delivered components
IssueReport_CBD1500635.pdf	Document describing known issues at the time of delivery
Readme_CBD1500635.html	This document
TechnicalReference_FBL_DrvSPI.pdf	Technical reference for the SPI Driver
TechnicalReference_FBL_GM_CMPR.pdf	Secure Fbl Compression handling
TechnicalReference_FBL_GM_Containers.pdf	Secure Fbl Container handling
TechnicalReference_FBL_GM_SLP6.pdf	Technical Reference for the Flash Bootloader
TechnicalReference_FBL_RH850.pdf	Hardware specific Flash Bootloader reference guide
TechnicalReference_FBL_Updater.pdf	Technical reference for the Updater component.
TechnicalReference_FBL_Updater_GM.pdf	OEM specific reference guide for the Updater
TechnicalReference_NvWrapper.pdf	Manual to describe possible Eeprom partitioning.
TechnicalReference_FBL_GM_CmprLzma.pdf	Secure Fbl Compression handling
UserManual_FlashBootloader.pdf	Getting started with the Flash Bootloader

Table 1-1   Flash Bootloader Documentation

2 Software Tools

The following Vector tools are included in your Flash Bootloader package.

File	Description
GENy	Code generation tool which auto generates parameter files for the Flash Bootloader based on customer input.
vFlash	Flash tool which is capable of downloading application as well as calibration data to an ECU running the Flash Bootloader. The tool itself is not a standard part of the delivery. The delivery comes with a GM specific template to be used with the vFlash tool.
Hexview	Viewer and editor of container files. May be used to edit & create required containers manually. A script solution that generates the required header formats automatically will be provided, check TechnicalReference_FBL_GM_Containers.pdf

Table 2-1   Vector Tools

3 Build

The Bootloader supports the v6.1.6 of the Green Hills Compiler command line tools.

3.1 Compiler and Linker Options

The Compiler and Linker Options used in this Bootloader can be found in DeliveryDescription_CBD1500635.html.

3.2 Compiler Warnings

Vector develops software that leads to as few compiler warnings as possible. It is unfortunately impossible to write platform-independent software that is functional on small 8-bit micro-controllers and powerful 32-bit micro-controllers and does not lead to compiler warnings. Changes to reduce the number of warnings for one compiler sometimes increase the number of warnings for another compiler.

The following compiler warnings occurred during integration and do no harm:

"../../../demo/demo_tja1145/_common/Compiler.h", line 92: warning #47-D:

   incompatible redefinition of macro "LOCAL_INLINE" (declared at line

   920 of "../../../bsw/_common/v_def.h")

 # define LOCAL_INLINE static inline

4 GENy configuration

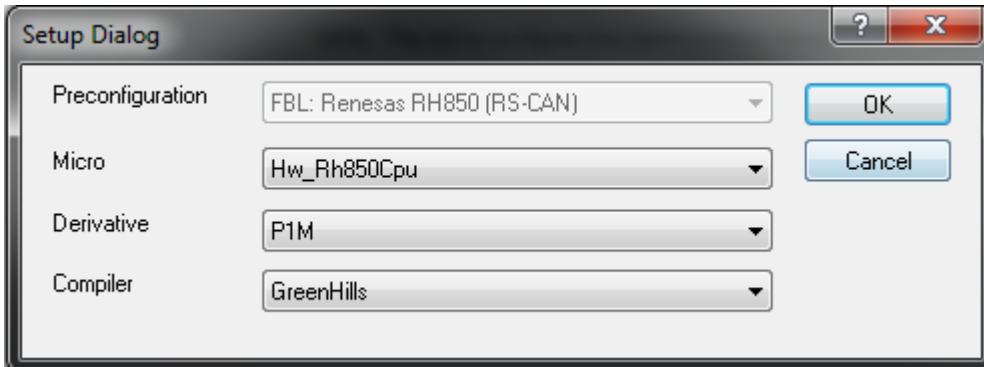When creating your own GENy configuration please use the following settings.

Figure 4-1   GENy Setup Dialog

4.1 User Configuration File

A user configuration file is needed to be included in the GENy configuration in order to replicate the environment used at Vector. The "MandatoryDeliveryPreconfig.cfg" file can be found in the Demo\DemoFbl\Config\ directory of the SIP.

There are also switches which need to be enabled in order to use FEE for the NVM elements used by GM. By default these switches are commented and basic NVM handling is used. These are described below:

Enable Fee Wrapper Functions

    #define FBL_APNV_USE_FUNCTIONS

Enable Fee NVM Handling

    #define FBL_HDR_DISABLE_BASIC_NVM_HANDLING

5 Demo Application

The provided DemoAppl project was used for test purpose by Vector and is provided to you to give you an example on a possible mapping and interfacing to the Bootloader. The project itself is a modified Bootloader which is remapped and interfacing the "real" Bootloader project **DemoFbl** like a real Application could do it.

Usually you are not able to successfully execute DemoAppl on your hardware. Our intention when providing it is, to enable you exploring the .map file, the linker file and **fbl_jmp_to_fbl.c** in order to understand better what you will have to do in your application to prepare for the Bootloader use case.

The jump from the application into the flash Bootloader is prepared and executed in fbl_jmp_to_fbl.c. This contains also an example how to implement the transition to CANdesc (commented out).

Please check the hw-dependent Technical Reference for a correct Mapping. This was used in the DemoAppl-project:

> applvect.c will overwrite the Bootloader Dummy Vector table (fbl_applvect) located at 0x20300 to provide the entry point to the application

> Note that the Demo Application does not use interrupts, so correct interrupt routing is not checked for that configuration

6 MSR Component Integration

6.1 Tja1145 CAN Transceiver

Example generated code from DaVinci is included in this delivery as a reference. The MSR BSW code for the CanTrcv component is not included. You can find all the files which were needed listed in Makefile.project.part.defines. The Vector SPI driver is included in the BSW source code and can be found in the BSW\FblSpi_RenesasCsih\ directory.

There is also a very basic Tja1145 driver which can put the transceiver in normal mode, but does not handle transitions. This is helpful for getting basic functionality working. The MSR CanTrcv BSW code was used for transceiver mode transitions.

6.2 Flash Emulated Eeprom (FEE)

Example generated code from DaVinci is included in this delivery as a reference. The MSR BSW code for the FEE / Fls components is not included. You can find all the files which were needed listed in Makefile.project.part.defines.

The file 'fbl_apnv_fee.c' is included in the delivery and provides a wrapper interface to the FEE. There are functions provided for initializing, reading, and writing the FEE. The other modifications which may be necessary would be to modify the ApplFblNVM* callback functions for each of the NVM elements to use the FEE wrapper functions in order to read/write the correct data.

7 Known Issues

See **IssueReport_CBD1500635.pdf**.

8 Contact

Any questions concerning the Flash Bootloader package should be sent to the following e-mail address:

fblsupport@us.vector.com

Visit our website for more information on

> News

> Products

> Demo software

> Support

> Training data

> Addresses

www.vector.com