Review Guideline for Renesas MCAL (RH850 X1x)

This application note describes topics to be checked as part of an MCAL configuration review. It applies to the RH850 X1x family (F1L/M/H, F1K, F1Kx, P1x, P1x-C and D1x) independent of the MCAL release version. The focus is on generic items that are not covered by checks of the configurator or generator tools. It does not address specific topics with close dependencies on the user application or potential issues by obvious misconfigurations.

Contents

1. Validation by Configurator Tool

Configurator tools like Vector DaVinci or Elektrobit Tresos perform a validation of the entered configuration values. This is based on the parameter definition files (BSWMD) that are delivered by Renesas. These files contain all available containers and parameters, the ranges of values or selectable options. The configurator tools typically mark any detected mismatches by detailed warning and error messages.

Validation is especially important to do when an existing configuration is upgraded to a newer release. An update of a release is done by importing the existing old configuration, based on a new parameter definition file. That new parameter definition file might have added or removed parameters, or the selectable values could have changed. Therefore, the configuration must be verified carefully after upgrade. All issues notified by the configurator tool must be checked and corrected if needed.

It is important to know that parameter mismatches are not explicitly checked and detected by the Renesas generator tools. Invalid parameters might have unexpected side effects.

2. Information and Warning Messages during Generation

The Renesas command line generator tools return three types of messages, errors, warnings, and information. Errors will abort the generation process and no source files are generated until the faults in the configuration are fixed.

Warnings might occur but the source files are created, nonetheless. It is important to check and justify the given warnings because they could indicate potential issues depending on the application and usage of MCAL functions. It is recommended to solve the warnings by configuration changes if possible.

Information messages like output of calculated frequencies can help the user to identify unintended settings.

3. Pin Usage and Assignment to Modules

During development or redesign phase the pin assignment might have to be changed. Finally, the pin configuration in the Port module should be compared with used pin functions in the peripheral modules with I/O functionality. This shall avoid double usage of a pin, assignment of a pin function to multiple pins or unused configurations.

Some peripherals like CAN and LIN interfaces have multiple options for the pin selection. According to the Hardware User Manuals a peripheral function must only be assigned to one pin at a time. This is especially valid for pins with combined function of CAN/LIN and interrupts (INTP). Double assignments are not checked or detected by the generator tools.

4. Module Parameters

This chapter lists some dedicated options that require attention independent of the application.

4.1 General for all Modules

These parameters are available for most MCAL modules and not described individually later.

Parameter Name / Function	Description
<SWC>DeviceName	The Device name specified in the module must match to the used microcontroller. When doing redesigns with reuse of existing configurations this setting remains at the original value.
<SWC>CriticalSectionProtection	Also called exclusive areas, shall be enabled for data protection of global variables, protected register accesses, to ensure correct timing etc. Dedicated analysis and risk assessment must be done in case this setting is disabled.
<SWC>DevErrorDetect	This option shall be enabled during development process to detect e.g. wrong usage of the API. In mass production application it shall be disabled (except FLS, here it depends on the concept).
TimeoutValue	Timeout checks are used to avoid endless loops in case of unexpected faults. The proper value depends on the application and the clock configuration. Some extra margin should be added.
API Enabling/Disabling	Many MCAL drivers offer optional APIs that are not needed by some applications. For RAM/ROM optimizations only those APIs should be enabled that are used by the application.

Explicit frequency values	If explicit frequency values are entered (no reference to MCU module), these must fit to the entered clock configuration. Verification by the generator tools is not possible in this case.

4.2 MCU Module

Parameter Name / Function	Description
McuMainOscStabilizationTime	The stabilization time must match to the value given by the crystal/oscillator vendor. If the time is too short, the further initialization might still be performed with an instable input frequency. If the time is too long, the initialization is delayed and the lifetime of the crystal might degrade, because a higher current is applied during stabilization time.
Mcu RAM Initialization	The RAM shall be cleared by the startup code before MCAL initialization. Therefore, the RAM initialization of MCAL module is optional. If used the RAM start address and length must match to the entries in the linker directive file.
McuClearPendingWakeupFactors McuWakeUpFactorsClearCentralised?	Pending wakeup factors can either be cleared centralized by MCU module or individually by the other modules (ICU, GPT, …)
McuAwoFout	FOUT is typically used as debug feature and shall be disabled in productive systems.
WakeupFactor<n>	If enabled a sub-container with selected wakeup factors should exist.
Option Bytes	Some devices (e.g. F1KM, F1KH) have flags controlling dividers for the clock configuration.

4.3 ADC Module

Parameter Name / Function	Description
AdcFunctionalityMode	This setting shall fit whether polling, interrupt, or both is used.
AdcSamplingTime	The ADC sampling time depends on the peripheral input clock. A wrong value might delay the conversion or cause imprecise conversion results.
ADC Frequency Ratio	Some controllers require the ratio of ADC peripheral clock and core clock to be within a defined range. Check the HW user manual for details.
AdcDmaMode	DMA shall only be enabled if groups have AdcResultAccessMode set to DMA.
AdcSgDmaChannel	A DMA channel resource must be used only by one module exclusively.
Group for Self-Diag	Group for Self-Diag shall be assigned to ADC group with highest priority.

4.4 GPT, ICU and PWM Modules

Parameter Name / Function	Description
TAU TaUnit Containers	The Timer Units (TAUB, TAUD and TAUJ) are shared between the modules GPT, PWM and ICU. The clocking prescalers must be configured for each used unit. It is recommended to configure the prescalers only in one of the three modules. (If same unit is configured multiple times, the timer unit configuration is also done multiple times during module initialization. If the parameters should differ the resulting configuration depends on order of <SWC>_Init calls.)
TAU Channels	A TAUx channel must only be used by one module, GPT or PMW or ICU. The 2nd channel internally reserved for duty cycle measurement must be considered. The generator tools of GPT, PWM and ICU do not check for double usage.

4.5 FLS Module

Parameter Name / Function	Description
FlsDmaChannel	A DMA channel resource for Fls write access must be used only by one module exclusively.

4.6 Port Module

Parameter Name / Function	Description
PortIpControl	Some peripherals require to set the PIPC flag for pins. This applies for example to data out and clock out of SPI interfaces (CSIH and CSIG modules).
PortInputBuffer	The input buffer must be enabled for DIO input pins to read the input level (instead of the programmed output level).
Analog and Digital filters	Analog and digital filters should be considered for all ICU module inputs.

4.7 SPI Module

Parameter Name / Function	Description
SpiLevelDelivered	This option selects either synchronous, asynchronous or both operation modes. To optimize code and memory consumption this should fit to the job settings.
SpiDmaMode	DMA shall only be enabled if SPI interfaces use DMA.
SpiRxDmaChannel SpiTxDmaChannel	A DMA channel resource must be used only by one module exclusively.

4.8 WDG Module

Parameter Name / Function	Description
Option Bytes	The settings of Wdg module must fit to the programmed Option Bytes. This cannot be verified by the configurator or generator or the driver software.

5. Build Environment

This chapter addresses the compiling, linking, and assembling process.

5.1 Proper Use of Compiler Options

Greenhills recommends using the builder for compiling, linking, and assembling instead of the standalone tools. Depending on the input files the builder selects the right operation.

The used builder options (for compiling, linking, and assembling) shall match to those given in the Release Notes of the MCAL package. Also, the MULTI version and patches must fit.

The options for MCAL are fixed but some options can be selected individually for other software modules:

- Allowed are different settings related to optimization.
- Not allowed are different data type related options, like --short_enum, -pack, --signed_chars, …
- Not allowed are different register mode settings: -registermode, -reserve_r2, -r21has65535, -notda, -fsoft/-fhard/-fnone, -sda=never/-sda=xxx
- Not allowed is a selection of different C language dialects (-ansi, …)

For details refer to the GHS user manuals.

5.2 Linker Directive File

Initialization of Data and BSS memory sections for variables is required for proper MCAL operation. This is done on system start-up by the start-up code and is controlled by the linker directive file. Following items should be checked:

- Data sections must have an entry in ROM with ROM() directive, that initial values are copied from RAM to ROM by _start function before the first usage by the driver code (MCAL Init functions).
- Data sections must not have the CLEAR attribute.
- BSS sections for not-initialized variables should have NOCLEAR attribute.

Note: After reset the content of RAM is undefined and no clearing is done by hardware. But a connected debugger typically clears the RAM after reset. This could hide issues of the linker directive file during development process.

5.3 Low Level Startup

After power on of the microcontroller the content of RAM is undefined, no initialization is done by the hardware. Therefore, the start-up code shall initialize the RAM to zero to have defined values and to initialize the ECC bits. The write accesses must be done 32-bit wise.

6. Revision History

Revision	Date	Description
1.0	Nov 24, 2021	Initial release.