Review Guideline for Renesas MCAL (RH850/X2x)

This application note describes topics to be checked as part of an MCAL configuration review. It applies to the RH850/X2x family (RH850/E2x and RH850/U2x) 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.

1. Configuration and Generation 2
1.1 Validation by Configurator Tool 2
1.2 Generation Tool Usage 2
1.3 Information and Warning Messages during Generation 2
2. Pin Usage and Assignment to Modules 2
3. MCAL Usage Guidance 2
3.1 Module Parameters 2
3.1.1. General for all Modules 3
3.1.2. MCU Module 3
3.1.3. CAN Module 3
3.1.4. ADC Module 4
3.1.5. GPT, ICU and PWM Modules 4
3.1.6. FLS Module 5
3.1.7. PORT Module 5
3.1.8. ETH Module 5
3.1.9. SPI Module 5
3.1.10. WDG Module 5
3.2 Clock Usage 6
3.3 Interrupt Handling 6
3.4 Module Standby Register Usage 6
3.5 Multi-instance Implementation 6
4. Build Environment 6
4.1 Proper Use of Compiler Options 6
4.2 Build Process 6
4.3 Linker Directive File 6
4.4 Low Level Startup 7
5. Revision History 8

1. Configuration and Generation

1.1 Validation by Configurator Tool

Configurator tool like Vector DaVinci performs 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 tool typically marks 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.

1.2 Generation Tool Usage

Generation tool is a command line tool that accepts ECU Configuration Description File(s), BSWMDT file, Translation XML file and Configuration XML file as input, and generates the C source and header files based on the configuration of the module.

The Generation Tool should not be executed more than five times in parallel. In case the user provides multiple Translation XML then Generation Tool will work with only last Translation XML file path via command line or Configuration XML File.

1.3 Information and Warning Messages during Generation

The Renesas command line generator tools return three types of messages which are 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.

2. 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 tool.

3. MCAL Usage Guidance

3.1 Module Parameters

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

3.1.1. 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 the microcontroller used. 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 the development process to detect e.g., wrong usage of the API.
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.

3.1.2. MCU Module

Parameter Name / Function	Description
McuClockResponseWaitCount	Parameter contains the Clock source change and Division ratio change response wait count. To satisfy all stabilization time among MOSC and PLLO, wait count value must be configured with maximum value.
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.
McuClearAndSetWakeupFactors McuClearAndSetWakeupFactorApi	Devices such as U2x have pending wakeup factors which can either be set or cleared for transition to Power-Down mode.
WakeupFactor<n>	Always ensure to configure a sub container with selected wakeup factors if this parameter is enabled in U2x.

3.1.3. CAN Module

Parameter Name / Function	Description
CanControllerPclkClock	This parameter represents the peripheral clock (pclk) which refer to the MCU container McuHighSpeedPeriClk.
CanControllerPplClock	This parameter represents the on-chip peripheral clock (clkc) which is set in the MCU driver configuration. This clock is considered only when the parameter CanClockSourceSelect (i.e., DCS bit) is configured as CLKC. E2x:

	This reference shall refer to the MCU container McuLowSpeedPeriClk. U2Ax, U2Bx: This reference shall refer to the MCU container McuHighSpeedPeriClk.
CanControllerMainClock	This parameter represents the crystal clock (clk_xincan) which refers to the MCU container McuRCANClk. This clock is considered only when the parameter CanClockSourceSelect (i.e. DCS bit) is configured as CLK_XINCAN.

3.1.4. ADC Module

Parameter Name / Function	Description
AdcFunctionalityMode	This setting shall fit whether polling, interrupt, or both is used.
AdcEnableDma	Enables / disables the DMA functionality of the ADC driver
AdcSgDmaChannel	A DMA channel resource must be used only by one module exclusively.
Group for Self-Diag	The group for Self-Diag shall be assigned to ADC group with highest priority.
AdcClkLsbRef	This parameter specifies the reference to the MCU container McuLowSpeedPeriClk
AdcClkAdcRef	This reference shall refer to the MCU container McuADCClk.
AdcClkAAdcRef	This reference shall refer to the MCU container McuADCClkA.

3.1.5. GPT, ICU and PWM Modules

Parameter Name / Function	Description
TAU TauUnit Containers/ Atu5 Timer Block Select Containers	The Timer Units (TAUD and TAUJ) are shared between the modules GPT, PWM and ICU for U2x. Similarly, the ATU-V Timer sub blocks (ATU-V Timer A, ATU-V Timer C, ATU-V Timer D and ATU-V Timer G) are shared between the modules GPT, PWM and ICU for E2x. The clocking prescalers must be configured for each units used. 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 depending on the resulting configuration, then it depends on the order of <SWC>_Init invocation.)
TAU / ATU Channels	A TAUx or ATU-V Timer sub block channel must be used only by a single module i.e., GPT or PMW or ICU. The generator tool of GPT, PWM and ICU do not check for double usage.
GptClockReference/ PwmMcuClockReferencePoint	This reference shall refer to the MCU container McuClockSettingConfig (McuLowSpeedPeriClk / McuHighSpeedPeriClk / McuTAUJClk / /McuClockReferencePoint).

3.1.6. FLS Module

Parameter Name / Function	Description
FlsCpuFrequency	Reference to the CPU subsystem frequency configuration, which is set in the MCU driver configuration. This reference from MCU informs the programming HW frequency to the programming Firmware. The clock value in MCU must be according to the clock supply setting of the CPU Subsystem. Please make sure to set the correct clock value in MCU configuration to avoid damage to the flash hardware.

3.1.7. PORT Module

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

3.1.8. ETH Module

Parameter Name / Function	Description
Initialization of GRAM/RRAM EthRamSize	The EthRamSize parameter is the size of the heap memory used in this module. Tx buffers, Rx buffers and the descriptors are dynamically allocated using heap memory. Hence, the user shall make sure to configure enough memory to allocate the buffers and descriptors.
EthInputClockRef	HBUS clock (AXI/AHB) McuHBusClk set directly for the ENTB Macro.

3.1.9. 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.
SpiClockFrequencyRef	This reference contains assignment of the MCU frequency McuHighSpeedPeriClk.

3.1.10. 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.

WdgClockRef	This reference shall refer to the MCU container McuWDTClk / McuWDTClkA.

3.2 Clock Usage

Clock settings should be taken care across the whole system when configuring each software component. For the E2x Devices, the clock must be geared up for peripheral functions. The external clock must be configured after clock gear up is complete.

3.3 Interrupt Handling

The MCAL driver does not set some bits (EITBn, EIPn) of EIC Register. Therefore, it is necessary to set such bits according to the interrupt method used by the user before initializing the MCAL driver. The MCAL driver for U2Ax devices does not consider the Interrupt Overflow bit in EIC register. Missing interrupt request can be detected by "Periodic check of not executed interrupt requests" in Safety Application Note. In U2Bx, it is necessary to pay attention when using interrupt source between modules.

For details refer to the HW user manual.

3.4 Module Standby Register Usage

Module Standby Registers are used to enable/disable all clocks connected to the module.

3.5 Multi-instance Implementation

When MCAL driver is used for multi-instance, interface name will change, and the user should update/create BSWMDT file as per user's configuration to avoid redundant information.

4. Build Environment

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

4.1 Proper Use of Compiler Options

Greenhills recommends the use of 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 Compiler version 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 data type related option, like --short_enum

For details refer to the GHS user manuals.

4.2 Build Process

The Make file consists of the GNU Make (version 3.81) compatible script to build the Driver Component in case of any change in the configuration.

4.3 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 (E2 Emulator) typically clears the RAM after reset. This could hide issues of the linker directive file during development process.

4.4 Low Level Startup

After powering 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.

5. Revision History

Revision	Date	Description
1.0	April 20,2023	Initial release.