Integration Manual

for MPC574XG FR Driver

Document Number: IM35FRASR4.2 Rev002R1.0.0
Rev. 1.0.0

Contents

Section number **Title** **Page**

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0.0	09-Feb-2017	Nam Khuc	RTM 1.0.0 version

Chapter 2
Introduction

This integration manual describes the integration requirements for FR Driver for MPC574XG microcontrollers.

2.1 Acronyms and Definitions

Table 2-1. Acronyms and Definitions

Term	Definition
FR	FlexRay
FRIF	FlexRay Interface
BSW	Basic Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
ISR	Interrupt Service Routine
OS	Operating System
GUI	Graphical User Interface
API	Application Programming Interface
PB Variant	Post Build Variant
LT Variant	Link Time Variant
PC Variant	Pre Compile Variant

2.2 Reference List

Table 2-2. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002FR Driver Software Specification Document.	2.6.0 R4.2.1 Rev 1
2	MPC5748G Reference Manual	Rev. 5, 12/2016

Table continues on the next page...

Table 2-2. Reference List (continued)

#	Title	Version
3	MPC5746C Reference Manual	Rev. 4, 12/2016
4	MPC5748G_1N81M_Rev.2 (official document) (1N81M)	Jun-16
5	MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0 (internal document) (1N81M, 0N78S)	31.10.2016
6	MPC5746C_1N06M_Rev.4 (official document) (1N06M)	Jul-16
7	MPC5746C_cut1.1_cut2.0_cut2.1_comparison_v0 (internal document) (1N06M, 0N84S, 1N84S)	14-Sep-16
8	C3M_cut2.1_new_errata_20170113 (internal document) (1N84S)	13-Jan-17

Chapter 3
Building the Driver

This section describes the source files and various compilers, linker options used for building the Autosar FR driver for NXP SemiconductorMPC574XG . It also explains the EB Tresos Studio plugin setup procedure.

3.1 Build Options

The FR driver files are compiled using

- Windriver DIAB DIAB_5_9_6_2
- Green Hills Multi 7.1.4 / Compiler 2015.1.6

The compiler, linker flags used for building the driver are explained below:

Note

The TS_T2D35M10I0R0 plugin name is composed as follow:

TS_T = Target_Id

D = Derivative_Id

M = SW_Version_Major

I = SW_Version_Minor

R = Revision

(i.e. Target_Id = 2 identifies PA architecture and Derivative_Id = 35 identifies the MPC574XG)

3.2 GHS Compiler/Linker/Assembler Options

Table 3-1. Compiler Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-ansi	Specifies ANSI C with extensions. This mode extends the ANSI X3.159-1989 standard with certain useful and compatible constructs.
-noSPE	Disables the use of SPE and vector floating point instructions by the compiler.
-Ospace	Optimize for size.
-sda=0	Enables the Small Data Area optimization with a threshold of 0.
-vle	Enables VLE code generation
-dual_debug	Enables the generation of DWARF, COFF, or BSD debugging information in the object file
-G	Generates source level debugging information and allows procedure call from debugger's command line.
--no_exceptions	Disables support for exception handling
-Wundef	Generates warnings for undefined symbols in preprocessor expressions
-Wimplicit-int	Issues a warning if the return type of a function is not declared before it is called
-Wshadow	Issues a warning if the declaration of a local variable shadows the declaration of a variable of the same name declared at the global scope, or at an outer scope
-Wtrigraphs	Issues a warning for any use of trigraphs
--prototype_errors	Generates errors when functions referenced or called have no prototype
--incorrect_pragma_warnings	Valid #pragma directives with wrong syntax are treated as warnings
-noslashcomment	C++ like comments will generate a compilation error
-preprocess_assembly_files	Preprocesses assembly files
-nostartfile	Do not use Start files
--short_enum	Store enumerations in the smallest possible type
--diag_error 223	Sets the specified compiler diagnostic messages to the level of error
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DUSE_SW_VECTOR_MODE	-D defines a preprocessor symbol and optionally can set it to a value. USE_SW_VECTOR_MODE: By default in the package, drivers are compiled to be used with interrupt controller configured to be in hardware vector mode. In case of AUTOSAR_OS_NOT_USED, the compiler option "-DUSE_SW_VECTOR_MODE" must be added to the list of compiler options to be used with interrupt controller configured to be in software vector mode.
-DDISABLE_MCAL_INTERMODULE_ASR_CHECK	-D defines a preprocessor symbol to disable the inter-module version check for AR_RELEASE versions. DISABLE_MCAL_INTERMODULE_ASR_CHECK: By default in the package, drivers are compiled to perform the inter-module version check as per Autosar BSW004. When the inter-module version check needs to be disabled then the DISABLE_MCAL_INTERMODULE_ASR_CHECK global define must be added to the list of compiler options.
-DGHS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the GHS preprocessor symbol.
-c	Produces an object file (called input-file.o) for each source file.

Table 3-2. Assembler Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-G	Generates source level debugging information and allows procedure call from debugger's command line.
-list	Creates a listing by using the name of the object file with the .lst extension

Table 3-3. Linker Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-nostartfiles	Do not use Start files.
-vle	Enables VLE code generation
--nocpp	Do not Generate Constructors/Destructors
-Mn	sort numerically the MAP file
-delete	The -delete option instructs the linker to remove functions that are not referenced in the final executable.
-ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete. DWARF debug information will contain references to deleted functions that may break some third-party debuggers.
-keepmap	keeps the MAP file in case of link error

3.3 DIAB Compiler/Linker/Assembler Options

Table 3-4. Compiler Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-Xdialect-ansi	Follow the ANSI C standard with some additions
-XO	Enables extra optimizations to produce highly optimized code
-g3	Generate symbolic debugger information and do all optimizations.
-Xsize-opt	Optimize for size rather than speed when there is a choice
-Xsmall-data=0	Set Size Limit for 'small data' Variables to zero.
-Xsmall-const=0	Set Size Limit for “small const” Variables to zero.

Table continues on the next page...

Table 3-4. Compiler Options (continued)

Option	Description
-Xaddr-sconst=0x11	Specify addressing for constant static and global variables with size less than or equal to -Xsmall-const to far-absolute.
-Xaddr-sdata=0x11	Specify addressing for non-constant static and global variables with size less than or equal to -Xsmall-data in size to far-absolute.
-Xno-common	Disable use of the 'COMMON' feature so that the compiler or assembler will allocate each uninitialized public variable in the .bss section for the module defining it, and the linker will require exactly one definition of each public variable
-Xnested-interrupts	Allow nested interrupts
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xdebug-local-all	Force generation of type information for all local variables
-Xdebug-local-cie	Create common information entry per module
-Xdebug-struct-all	Force generation of type information for all typedefs, struct, union and class types
-Xforce-declarations	Generates warnings if a function is used without a previous declaration
-ee1481	Generate an error when the function was used before it has been declared
-Xmacro-undefined-warn	Generates a warning when an undefined macro name occurs in a #if preprocessor directive
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files
-W:as:,-l	Pass the option '-l' (lower case letter L) to the assembler to get an assembler listing file
-Wa,-Xisa-vle	Instruct the assembler to expect and assemble VLE (Variable Length Encoding) instructions rather than BookE instructions.
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DUSE_SW_VECTOR_MODE	-D defines a preprocessor symbol and optionally can set it to a value. USE_SW_VECTOR_MODE: By default in the package, drivers are compiled to be used with interrupt controller configured to be in hardware vector mode. In case of AUTOSAR_OS_NOT_USED, the compiler option "-DUSE_SW_VECTOR_MODE" must be added to the list of compiler options to be used with interrupt controller configured to be in software vector mode.
-DDIAB	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the DIAB preprocessor symbol.
-DDISABLE_MCAL_INTERMODULE_ASR_CHECK	-D defines a preprocessor symbol to disable the inter-module version check for AR_RELEASE versions. DISABLE_MCAL_INTERMODULE_ASR_CHECK: By default in the package, drivers are compiled to perform the inter-module version check as per Autosar BSW004. When the inter-module version check needs to be disabled then the DISABLE_MCAL_INTERMODULE_ASR_CHECK global define must be added to the list of compiler options.
-c	Stop after assembly, produce object file.

Table 3-5. Assembler Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries

Table continues on the next page...

Table 3-5. Assembler Options (continued)

Option	Description
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-g	Dump the symbols in the global symbol table in each archive file.
-Xisa-vle	Expect and assemble VLE (Variable Length Encoding) instructions rather than Book E instructions. The default code section is named .text_vle instead of .text, and the default code section fill "character" is set to 0x44444444 instead of 0. The .text_vle code section will have ELF section header flags marking it as VLE code, not Book E code.
-Xasm-debug-on	Generate debug line and file information
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xsemi-is-newline	Treat the semicolon (;) as a statement separator instead of a comment character.

Table 3-6. Linker Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to tPPCE200Z4204N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-tPPCE200Z210N3VEN:simple	Sets target processor to tPPCE200Z210N3VEN, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries
-Xelf	Generates ELF object format for output file
-m6	Generates a detailed link map and cross reference table
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files

3.4 Files Required for the Compilation

This section describes the include files required to compile, assemble (if assembler code) and link the AUTOSAR FR driver for MPC574XG microcontrollers

To avoid integration of incompatible files, all the include files from other modules shall have the same AR_MAJOR_VERSION and AR_MINOR_VERSION, i.e. only files with the same AUTOSAR major and minor versions can be compiled.

FR Files

- ..\Fr_TS_T2D35M10I0R0\src\Fr.c
- ..\Fr_TS_T2D35M10I0R0\src\Fr_IPW.c
- ..\Fr_TS_T2D35M10I0R0\src\Fr_Flexray.c
- ..\Fr_TS_T2D35M10I0R0\include\Fr.h

- ..\Fr_TS_T2D35M10I0R0\include\Fr_IPW.h
- ..\Fr_TS_T2D35M10I0R0\include\Fr_Flexray.h

- Fr_[VariantName]_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, PB). For driver compilation, this file should be generated by the user using a configuration tool.The file contains the definition of the init pointer for the respective variant.
- Fr_Cfg.h - For driver compilation, this file should be generated by the user using a configuration tool

Files from Base common folder

- ..\Base_TS_T2D35M10I0R0\include\Cer.h
- ..\Base_TS_T2D35M10I0R0\include\Compiler.h
- ..\Base_TS_T2D35M10I0R0\include\Compiler_Cfg.h
- ..\Base_TS_T2D35M10I0R0\include\ComStack_Types.h
- ..\Base_TS_T2D35M10I0R0\include\Mcal.h
- ..\Base_TS_T2D35M10I0R0\include\Fr_MemMap.h
- ..\Base_TS_T2D35M10I0R0\include\Platform_Types.h
- ..\Base_TS_T2D35M10I0R0\include\Reg_eSys.h
- ..\Base_TS_T2D35M10I0R0\include\Reg_Macros.h
- ..\Base_TS_T2D35M10I0R0\include\Soc_Ips.h
- ..\Base_TS_T2D35M10I0R0\include\Std_Types.h

Files from FrIf folder:

- ..\FrIf_TS_T2D35M10I0R0\include\Fr_GeneralTypes.h

Files from Dem folder:

- ..\Dem_TS_T2D35M10I0R0\Dem.h
- ..\Dem_TS_T2D35M10I0R0\Dem_IntErrId.h
- ..\Dem_TS_T2D35M10I0R0\Dem_Types.h

Files from Det folder:

- ..\Det_TS_T2D35M10I0R0\Det.h

3.5 Setting up the Plugins

The FR driver was designed to be configured by using the EB tresos Studio (version EB tresos Studio 21.0.0 b160607-0933 or later). Since Autosar FR Driver uses also the configuration data given by Autosar FrIf plugin, the FrIf files are required for correct configuration generation.

Location of various files inside the FR module folder:

- VSMD (Vendor Specific Module Definition) file in EB tresos Studio XDM format:
  - ..\Fr_TS_T2D35M10I0R0\config\Fr.xdm
  - ..\FrIf_TS_T2D35M10I0R0\config\FrIf.xdm
- VSMD (Vendor Specific Module Definition) file(s) in AUTOSAR compliant EPD format:
  - ..\Fr_TS_T2D35M10I0R0\autosar\Fr.epd
  - ..\FrIf_TS_T2D35M10I0R0\autosar\FrIf.epd
- • Code Generation Templates for parameters without variation points
  - ..\Fr_TS_T2D35M10I0R0\generate_PC\include_PC\Fr_Cfg.h
  - ..\Fr_TS_T2D35M10I0R0\generate_PC\include_PC\Fr_DeviceMacros.m
  - ..\Fr_TS_T2D35M10I0R0\generate_PC\include_PC\Fr_VersionCheck.m
- Code Generation Templates for variant aware parameters
  - ..\Fr_TS_T2D35M10I0R0\generate_PC\include_PB\Fr_PBcfg.h
  - ..\Fr_TS_T2D35M10I0R0\generate_PC\include_PB\Fr_VersionCheck.m
  - ..\Fr_TS_T2D35M10I0R0\generate_PC\src_PB\Fr_PBcfg.c
  - ..\Fr_TS_T2D35M10I0R0\generate_PB\src_PB\Fr_DeviceMacros.m
  - ..\Fr_TS_T2D35M10I0R0\generate_PB\src_PB\Fr_VersionCheck.m

Steps to generate the configuration:

1. Copy both FR module, FrIf module folders (Fr_TS_T2D35M10I0R0, FrIf_TS_T2D35M10I0R0) and freescale.tresos.flexray.jar file into EB tresos Studio installation path under '\plugins' folder.
2. Set the desired Tresos Output location folder for the generated sources and header files.
3. Use the EB tresos Studio GUI to modify ECU configuration parameters values.
4. Generate the configuration files.

NOTE

For using an alternative plug-ins directory with EB tresos, please follow the steps below:

1. Copy the plug-in(s) (named like "Fr_TS_T2D35M10I0R0") into a directory "<MyDirectoryPath>\<MyDirectoryName>" under the folders "<MyDirectoryPath>\<MyDirectoryName>\eclipse \plugins"
2. Create a text file with the extension ".link" into the EB tresos installation in a folder called "links" (e.g. <MyTresosIntallation>\links \<MyPluginPackageName>.link)

3. Put the following text in that "<MyPluginPackageName>.link" file: "path=<MyDirectoryPath>/<MyDirectoryName>", please make sure the path is described with forward slashes.

Chapter 4
Function calls to module

4.1 Function Calls during Start-up

This driver does not need OS Support. To initialize the FR module, the API Fr_Init() should be called.

4.2 Function Calls during Shutdown

In order to disable FR module, the user can call the Fr_DeInit function. But this function can not apply for platforms which have IpVault lower version 10.

4.3 Function Calls during Wake-up

None.

Chapter 5
Module requirements

5.1 Exclusive Areas to Be Defined in BSW Scheduler

None.

5.2 Peripheral Hardware Requirements

FR does not take care of configuring other components but requires that their initialization precede the FR initialization. The following components need to be configured:

- External pins - FR_A_RX, FR_A_TX, FR_A_TX_EN, FR_B_RX, FR_B_TX and FR_B_TX_EN pins should be initialized
- The clock for FR - Please refer to the chapter Protocol Engine Clocking in MPC5748G Reference Manual [Reference List] for the clock configuration (clock source, frequency value)
- FR Memory - FR Memory is located in the system memory of the MCU. The FlexRay block does not provide a memory protection scheme for the FR Memory. The FR Memory is a contiguous region where data like, frame headers, physical message buffers payload data, and synchronization tables is stored; the size is maximum 64Kbytes and starts at a 16 byte boundary. It should be configured as cache-inhibited, please see Data Cache Restriction chapter [Data Cache Restriction].

5.3 ISR to Configure within OS - Dependencies

FR does not use interrupt vector for process each message. In order to monitor the timing of protocol, the AutoSar also provide some APIs to process timer interrupt follow:

- Fr_SetAbsoluteTimer: setup timer API.

- Fr_EnableAbsoluteTimerIRQ: enable timer interrupt.
- Fr_GetAbsoluteTimerIRQStatus: get the status of timer interrupt.
- Fr_AckAbsoluteTimerIRQ: clear the timer interrupt flag.
- Fr_DisableAbsoluteTimerIRQ: disable timer interrupt.
- Fr_CancelAbsoluteTimer: disable timer.

On the vector tabel of this platform, the interrupt vector for timer interrupt is 458 (PRIF). To use this feature, users have to create their own ISR.

5.4 Other AUTOSAR Modules - Dependencies

- **Port:** This module shall configure the Port pins which are to be used for the FR operation.
- **Mcu:** This module shall be initialized before using FR. This module is required for setting the system clock frequency (clock for FR).
- **Det** (only if FrDevErrorDetect=true): This module is necessary for enabling Development error detection. The API function used is Det_ReportError(). The activation/deactivation of Development error detection is configurable using 'FrDevErrorDetect' configuration parameter.
- **Dem:** This module is necessary for enabling reporting of production relevant error status. The API function used is Dem_ReportErrorStatus().

5.5 XBAR Configuration

To prevent system bus failures during FlexRay operation (SBCF_EF is set) it must be ensured that FlexRay as the bus master has assigned sufficient priority to be able to access SRAM in any time without contention with another bus master. This can be ensured by increasing the FlexRay module priority on the crossbar switch (XBAR). In case of there are two SRAM blocks with its own slave ports it is highly recommended to assign one slave port to FlexRay only to prevent bus collision with another master.

5.6 Data Cache Restriction

To ensure that FR driver always reads latest data updated by the FlexRay CC, FR Memory area has to be configured as cache-inhibited. This is ensured by the Fr_MemoryArea[] array which size and start address corresponds with the FR Memory area and which is located in a non-cacheable section of the MCU system memory.

5.7 User mode support

On this platform, the Flexray module can run with both User mode and Suppervisor mode.

Chapter 6
Main API Requirements

6.1 Main functions calls within BSW scheduler

None.

6.2 API Requirements

Not Applicable.

6.3 Calls to Notification Functions, Callbacks, Callouts

Call-back Notifications

- None

User Notification

- None

Chapter 7
Memory Allocation

7.1 Sections to Be Defined in Fr_MemMap.h

Table 7-1. Sections to be defined in Fr_MemMap.h

Section Name	Type Of Section	Description
FR_START_SEC_CONFIG_DATA_UNSPECIFIED	Configuration Data	Start of Memory Section for Config Data
FR_STOP_SEC_CONFIG_DATA_UNSPECIFIED	Configuration Data	End of Memory Section for Config Data
FR_START_SEC_CODE	Code	Start of memory Section for Code
FR_STOP_SEC_CODE	Code	End of memory Section for Code
FR_START_SEC_VAR_NO_INIT_BOOLEAN	Variables	Used for variables which have to be aligned to boolean type. These variables are never cleared and never initialized by start-up code.
FR_STOP_SEC_VAR_NO_INIT_BOOLEAN	Variables	End of above section.
FR_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE	Non-Cacheable Variables	Used for variables, structures, arrays when the SIZE (alignment) does not fit the criteria of 8,16 or 32 bit, and that have to be stored in a non-cacheable memory section. These variables are never cleared and never initialized by start-up code.
FR_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE	Non-Cacheable Variables	End of above section.
FR_START_SEC_VAR_NO_INIT_8	Variables	Used for variables which have to be aligned to 8 bit. For instance used for variables of size 8 bit or used for composite data types: arrays, structs containing elements of maximum 8 bits. These variables are never cleared and never initialized by start-up code.
FR_STOP_SEC_VAR_NO_INIT_8	Variables	End of above section.
FR_START_SEC_VAR_NO_INIT_UNSPECIFIED	Variables	Used for variables, structures, arrays when the SIZE (alignment) does not fit the criteria of 8,16 or 32 bit. These variables are never cleared and never initialized by start-up code.
FR_STOP_SEC_VAR_NO_INIT_UNSPECIFIED	Variables	End of above section.

Table continues on the next page...

Table 7-1. Sections to be defined in Fr_MemMap.h (continued)

Section Name	Type Of Section	Description
FR_START_SEC_VAR_INIT_BOOLEAN	Variables	Used for variables which have to be aligned to boolean type. These variables are initialized with values after every reset.
FR_STOP_SEC_VAR_INIT_BOOLEAN	Variables	End of above section.
FR_START_SEC_CONST_UNSPECIFIED	Constant Data	Used for constants that does not fit the criteria of 8,16 or 32 bit.
FR_STOP_SEC_CONST_UNSPECIFIED	Constant Data	End of above section.

7.2 Linker Command File

Memory shall be allocated for every section defined in MemMap.h.

Moreover it has to be ensured that cache inhibited section is allocated for the FlexRay module in the linker command file, please see Data Cache Restriction chapter [Data Cache Restriction]. The maximum size of the FlexRay memory partition is 64 Kbytes and it starts at a 16 byte boundary.

Chapter 8
Configuration Parameters

Configuration parameter class for Autosar FR and FRIF drivers fall into the following variants as defined below:

Table 8-1. Configuration Parameters

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
FrGeneral			
	FrPrepareLPduSupport	PC	PC
	FrReconfigLPduSupport	PC	PC
	FrDisableLPduSupport	PC	PC
	FrCtrlTestCount	PC	PC
	FrDevErrorDetect	PC	PC
	FrIndex	PC	PC
	FrNumCtrlSupported	PC	PC
	FrRxStringentCheck	PC	PC
	FrRxStringentLengthCheck	PC	PC
	FrVersionInfoApi	PC	PC
FrGeneral/VendorSpecific			
	BufferReconfigurationEnable	PC	PC
	NetworkManagementVectorEnable	PC	PC
FrGeneral/VendorSpecific			
	ConnectedCC	PC	PC
	FrUnusedBitValue	PC	PC
	FrDisableDemReportErrorStatus	PC	PC
FrMultipleConfiguration/FrController			
	FrCtrlIdx	PB/PC	PB/PC
	pAllowHaltDueToClock	PB/PC	PB/PC
	pAllowPassiveToActive	PB/PC	PB/PC
	pChannels	PB/PC	PB/PC
	pClusterDriftDamping	PB/PC	PB/PC
	pDecodingCorrection	PB/PC	PB/PC

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
	pDelayCompensationA	PB/PC	PB/PC
	pDelayCompensationB	PB/PC	PB/PC
	pExternalSync	PB/PC	PB/PC
	pFallBackInternal	PB/PC	PB/PC
	pKeySlotId	PB/PC	PB/PC
	pKeySlotOnlyEnabled	PB/PC	PB/PC
	pKeySlotUsedForStartup	PB/PC	PB/PC
	pKeySlotUsedForSync	PB/PC	PB/PC
	pLatestTx	PB/PC	PB/PC
	pMacroInitialOffsetA	PB/PC	PB/PC
	pMacroInitialOffsetB	PB/PC	PB/PC
	pMicroInitialOffsetA	PB/PC	PB/PC
	FpMicroInitialOffsetB	PB/PC	PB/PC
	pMicroPerCycle	PB/PC	PB/PC
	pNmVectorEarlyUpdate	PB/PC	PB/PC
	pOffsetCorrectionOut	PB/PC	PB/PC
	pOffsetCorrectionStart	PB/PC	PB/PC
	pPayloadLengthDynMax	PB/PC	PB/PC
	pRateCorrectionOut	PB/PC	PB/PC
	pSamplesPerMicrotick	PB/PC	PB/PC
	pSecondKeySlotId	PB/PC	PB/PC
	pTwoKeySlotMode	PB/PC	PB/PC
	pWakeupChannel	PB/PC	PB/PC
	pWakeupPattern	PB/PC	PB/PC
	pdAcceptedStartupRange	PB/PC	PB/PC
	pdAcceptedStartupRange	PB/PC	PB/PC
	pdListenTimeout	PB/PC	PB/PC
	pdMicrotick	PB/PC	PB/PC
FrMultipleConfiguration/FrController/FrAbsoluteTimer			
	FrAbsTimerIdx		
FrMultipleConfiguration/FrController/ FrControllerDemEventParameterRefs			
	FR_E_CTRL_TESTRESULT		
FrController/FrFifo			
	FrAdmitWithoutMessageId		
	FrBaseCycle		
	FrChannels		
	FrCycleRepetition		

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
	FrFifoDepth		
	FrMsgIdMask		
	FrIdMatch		
FrController/FrFifo/FrRange			
	FrRangeMax		
	FrRangeMin		
FrMultipleConfiguration/FrController/VendorSpecific			
	CCBaseAddress	PB/PC	PB/PC
	CCFlexRayMemoryBaseAddress	PB/PC	PB/PC
	SystemMemoryAccessTimeOut	PB/PC	PB/PC
	ClockSource	PB/PC	PB/PC
	ChannelBitrate	PB/PC	PB/PC
	SingleChannelModeEnabled	PB/PC	PB/PC
	ForceReconfiguration	PB/PC	PB/PC
FrIfConfig/FrIfCluster			
	FrIfClstIdx	PB/PC	PB/PC
	FrIfClstIdx	PB/PC	PB/PC
	FrIfDetectNITError	PB/PC	PB/PC
	gChannels	PB/PC	PB/PC
	gColdStartAttempts	PB/PC	PB/PC
	gColdStartAttempts	PB/PC	PB/PC
	gCycleCountMax	PB/PC	PB/PC
	gListenNoise	PB/PC	PB/PC
	gMacroPerCycle	PB/PC	PB/PC
	gMaxWithoutClockCorrectionFatal	PB/PC	PB/PC
	gMaxWithoutClockCorrectionPassive	PB/PC	PB/PC
	gNetworkManagementVectorLength	PB/PC	PB/PC
	gNumberOfMinislots	PB/PC	PB/PC
	gNumberOfStaticSlots	PB/PC	PB/PC
	gPayloadLengthStatic	PB/PC	PB/PC
	gSyncFrameIDCountMax	PB/PC	PB/PC
	gdActionPointOffset	PB/PC	PB/PC
	gdBit	PB/PC	PB/PC
	gdCASRxLowMax	PB/PC	PB/PC
	gdCycle	PB/PC	PB/PC
	gdDynamicSlotIdlePhase	PB/PC	PB/PC
	gdIgnoreAfterTx	PB/PC	PB/PC
	gdMacrotick	PB/PC	PB/PC

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
	gdMiniSlotActionPointOffset	PB/PC	PB/PC
	gdMinislot	PB/PC	PB/PC
	gdNIT	PB/PC	PB/PC
	gdSampleClockPeriod	PB/PC	PB/PC
	gdStaticSlot	PB/PC	PB/PC
	gdSymbolWindow	PB/PC	PB/PC
	gdSymbolWindowActionPointOffset	PB/PC	PB/PC
	gdTSSTansmitter	PB/PC	PB/PC
	gdWakeupSymbolRxIdle	PB/PC	PB/PC
	gdWakeupSymbolRxLow	PB/PC	PB/PC
	gdWakeupSymbolRxWindow	PB/PC	PB/PC
	gdWakeupsymbolTxLow	PB/PC	PB/PC
	gdWakeupSymbolTxIdle	PB/PC	PB/PC
	FrIfMainFunctionPeriod	PB/PC	PB/PC
	FrIfMaxIsrDelay	PB/PC	PB/PC
	FrIfSafetyMargin	PB/PC	PB/PC
FrIfConfig/FrIfFrameStructure			
	FrIfByteOrder	PB/PC	PB/PC
FrIfConfig/FrIfFrameStructure/FrIfPdusInFrame			
	FrIfByteOrder	PB/PC	PB/PC
	FrIfPduOffset	PB/PC	PB/PC
	FrIfPduReference	PB/PC	PB/PC
FrIfConfig/FrIfPdu			
	FrIfPduDirection	PB/PC	PB/PC
FrIfConfig/FrIfController/FrIfLPdu			
	FrIfLPduIdx	PB/PC	PB/PC
	FrIfLPduReconfigurable	PB/PC	PB/PC
	FrIfLPduTriggeringReference	PB/PC	PB/PC
FrIfCluster/FrIfController/FrIfFrameTriggering			
	FrIfAllowDynamicLSduLength	PB/PC	PB/PC
	FrIfBaseCycle	PB/PC	PB/PC
	FrIfChannel	PB/PC	PB/PC
	FrIfCycleRepetition	PB/PC	PB/PC
	FrIfSduLength	PB/PC	PB/PC
	FrIfPayloadPreamble	PB/PC	PB/PC
	FrIfSlotID	PB/PC	PB/PC
	FrIfFrameStructureReference	PB/PC	PB/PC
FrIfConfig/FrIfPdu			

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
	FrIfTxPdu	PB/PC	PB/PC
	FrIfRxPdu	PB/PC	PB/PC

Chapter 9
Integration Steps

This section gives a brief overview of the steps needed for integrating FlexRay :

- Generate the required FR configurations. For more details refer to section
- Allocate proper memory sections in FR_MemMap.h and linker command file. For more details refer to section
- Compile & build the FR with all the dependent modules. For more details refer to section Building the Driver

Chapter 10
External Assumptions for FR driver

The section presents requirements that must be complied with when integrating FR driver into the application.

[SMCAL_CPR_EXT163]

<< If interrupts are locked a centralized function pair to lock and unlock interrupts shall be used. >>

[SMCAL_CPR_EXT176]

<< The integrator shall assure that <MSN>_Init() and <MSN>_DeInit() functions do not interrupt each other. >>

[SWS_Fr_00117]

<< Fr_GeneralTypes.h shall contain all types and constants that are shared among the AUTOSAR FlexRay modules Fr, FrIf and FrTrcv. >>

NOTE

Must be ensured by integrator. FR driver does not provide this file.

[SWS_Fr_00498]

<< Error Name: FR_E_CTRL_TESTRESULT [_<Fr_CtrlIdx>]

Short Description: FlexRay Controller hardware error

Long Description: This extended production error indicates hardware errors of the FlexRay communication controller. Please note that this extended production error does not address Flexray protocol errors detected on the network.

Detection Criteria: Fail Every API function accessing hardware registers of the FlexRay controller might detect a missbehavior of the device compared to the device specification.

For details see requirements:

SWS_Fr_00147, SWS_Fr_00176, SWS_Fr_00181, SWS_Fr_00520, SWS_Fr_00186, SWS_Fr_00190, SWS_Fr_00195, SWS_Fr_00201, SWS_Fr_00216, SWS_Fr_00223, SWS_Fr_00613, SWS_Fr_00232, SWS_Fr_00243, SWS_Fr_00248, SWS_Fr_00529, SWS_Fr_00543, SWS_Fr_00255, SWS_Fr_00261, SWS_Fr_00552, SWS_Fr_00560, SWS_Fr_00568, SWS_Fr_00580, SWS_Fr_00589, SWS_Fr_00272, SWS_Fr_00286, SWS_Fr_00297, SWS_Fr_00308, SWS_Fr_00319, SWS_Fr_00331, SWS_Fr_00652

Pass During FlexRay communication controller initialization (function Fr_ControllerInit()) the proper operation of the hardware registers is successfully verified.

For details see requirements:

SWS_Fr_00598,

Secondary Parameters: In case of successful device operation verification (function Fr_ControllerInit()) report PASS.

In case of an error always report FAIL.

Time Required: If a FlexRay Controller hardware error occurs it shall be immediately reported as error.

Monitor Frequency continuous >>

NOTE

Value for this production code Event Id is assigned externally by the configuration of the Dem.

[SWS_Fr_00600]

<< Error Name: FR_E_LPDU_SLOTSTATUS [_<LPduIdx>]

Short Description: FlexRay Protocol communication error

Long Description: This production error indicates Flexray protocol communication errors on the network for a particular LPdu.

Detection Criteria: Fail Each time FlexRay slot status with at least one of the following FlexRay protocol errors active:

- vSS!SyntaxError

- vSS!ContentError

- vSS!Bviolation

For details see requirements:

SWS_Fr_00605, SWS_Fr_00606,

Pass Each time FlexRay slot status with none one of the following FlexRay protocol errors active:

- vSS!SyntaxError

- vSS!ContentError

- vSS!Bviolation

For details see requirements:

SWS_Fr_00627, SWS_Fr_00629,

Secondary Parameters: Detection of production error events is active only during ongoing FlexRay communication.

Time Required: The time for detecting an evident FlexRay protocol error in a particular slot strongly depends on the period of the actual FlexRay slot and thus on the FlexRay protocol configuration parameters.

Monitor Frequency continuous >>

NOTE

Value for this production code Event Id is assigned externally by the configuration of the Dem.

[SWS_Fr_00657]

<< The macros listed in this chapter shall be defined in Fr_GeneralTypes.h. >>

NOTE

Must be ensured by integrator.

[SWS_Fr_00110]

<< The types specified in SWS_Fr_00505, SWS_Fr_00506, SWS_Fr_00507, SWS_Fr_00508, SWS_Fr_00509, SWS_Fr_00510, SWS_Fr_00511, SWS_Fr_00512 and SWS_Fr_00514 shall be declared in Fr_GeneralTypes.h. >>

NOTE

Must be ensured by integrator. FR driver does not provide this file.

[SWS_Fr_00499]

<< The content of Fr_GeneralTypes.h shall be protected by a FR_GENERAL_TYPES define. >>

NOTE

Must be ensured by integrator.

[SWS_Fr_00500]

<< If different FlexRay drivers are used, only one instance of this file has to be included in the source tree. For implementation all Fr_GeneralTypes.h related types in the documents mentioned before shall be considered. >>

NOTE

Must be ensured by integrator.

[SWS_Fr_00505]

<< Name: Fr_POCStateType

Type: Enumeration

Range: FR_POCSTATE_CONFIG (=0) Represents literal CONFIG of formal type definition T_POCState.

FR_POCSTATE_DEFAULT_CONFIG Represents literal DEFAULT_CONFIG of formal type definition T_POCState.

FR_POCSTATE_HALT Represents literal HALT of formal type definition T_POCState.

FR_POCSTATE_NORMAL_ACTIVE Represents literal NORMAL_ACTIVE of formal type definition T_POCState.

FR_POCSTATE_NORMAL_PASSIVE Represents literal NORMAL_PASSIVE of formal type definition T_POCState.

FR_POCSTATE_READY Represents literal READY of formal type definition T_POCState.

FR_POCSTATE_STARTUP Represents literal STARTUP of formal type definition T_POCState.

FR_POCSTATE_WAKEUP Represents literal WAKEUP of formal type definition T_POCState.

Description: This formal definition refers to the description of type T_POCState in chapter 2.2.1.3 POC status of [12].

>>

NOTE

Fr_POCStateType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00506]

<< Name: Fr_SlotModeType

Type: Enumeration

Range: FR_SLOTMODE_KEYSLOT Represents literal KEYSLOT of formal type definition T_SlotMode.

FR_SLOTMODE_ALL_PENDING Represents literal ALL_PENDING of formal type definition T_SlotMode.

FR_SLOTMODE_ALL Represents literal ALL of formal type definition T_SlotMode.

Description: This formal definition refers to the description of type T_SlotMode in chapter 2.2.1.3 POC status of [12].

Name: Fr_SlotModeType

Type: Enumeration

Range: FR_SLOTMODE_KEYSLOT Represents literal KEYSLOT of formal type definition T_SlotMode.

FR_SLOTMODE_ALL_PENDING Represents literal ALL_PENDING of formal type definition T_SlotMode.

FR_SLOTMODE_ALL Represents literal ALL of formal type definition T_SlotMode.

Description: This formal definition refers to the description of type T_SlotMode in chapter 2.2.1.3 POC status of [12].

>>

NOTE

Fr_SlotModeType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00599]

<< There shall be a preprocessor macro with name FR_SLOTMODE_SINGLE that maps to value FR_SLOTMODE_KEYSLOT in file Fr_GeneralTypes.h. >>

NOTE

Must be ensured by integrator.

[SWS_Fr_00507]

<< Name: Fr_ErrorModeType

Type: Enumeration

Range: FR_ERRORMODE_ACTIVE (=0) Represents literal ACTIVE of formal type definition T_ErrorMode.

FR_ERRORMODE_PASSIVE Represents literal PASSIVE of formal type definition T_ErrorMode.

FR_ERRORMODE_COMM_HALT Represents literal COMM_HALT of formal type definition T_ErrorMode.

Description: This formal definition refers to the description of type T_ErrorMode in chapter 2.2.1.3 POC status of [12].

>>

NOTE

Fr_ErrorModeType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00508]

<< Name: Fr_WakeupStatusType

Type: Enumeration

Range: FR_WAKEUP_UNDEFINED (=0) Represents literal UNDEFINED of formal type definition T_WakeupStatus.

FR_WAKEUP_RECEIVED_HEADER Represents literal RECEIVED_HEADER of formal type definition T_WakeupStatus.

FR_WAKEUP_RECEIVED_WUP Represents literal RECEIVED_WUP of formal type definition T_WakeupStatus.

FR_WAKEUP_COLLISION_HEADER Represents literal COLLISION_HEADER of formal type definition T_WakeupStatus.

FR_WAKEUP_COLLISION_WUP Represents literal COLLISION_WUP of formal type definition T_WakeupStatus.

FR_WAKEUP_COLLISION_UNKNOWN Represents literal COLLISION_UNKNOWN of formal type definition T_WakeupStatus.

FR_WAKEUP_TRANSMITTED Represents literal TRANSMITTED of formal type definition T_WakeupStatus.

Description: This formal definition refers to the description of type T_WakeupStatus in chapter 2.2.1.3 POC status of [12].

>>

NOTE

Fr_WakeupStatusType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00509]

<< Name: Fr_StartupStateType

Type: Enumeration

Range: FR_STARTUP_UNDEFINED (=0) Represents literal UNDEFINED of formal type definition T_StartupState.

FR_STARTUP_COLDSTART_LISTEN Represents literal COLDSTART_LISTEN of formal type definition T_StartupState.

FR_STARTUP_INTEGRATION_COLDSTART_CHECK Represents literal INTEGRATION_COLDSTART_CHECK of formal type definition T_StartupState.

FR_STARTUP_COLDSTART_JOIN Represents literal COLDSTART_JOIN of formal type definition T_StartupState.

FR_STARTUP_COLDSTART_COLLISION_RESOLUTION Represents literal COLDSTART_COLLISION_RESOLUTION of formal type definition T_StartupState.

FR_STARTUP_COLDSTART_CONSISTENCY_CHECK Represents literal COLDSTART_CONSISTENCY_CHECK of formal type definition T_StartupState.

FR_STARTUP_INTEGRATION_LISTEN Represents literal INTEGRATION_LISTEN of formal type definition T_StartupState.

FR_STARTUP_INITIALIZE_SCHEDULE Represents literal INITIALIZE_SCHEDULE of formal type definition T_StartupState.

FR_STARTUP_INTEGRATION_CONSISTENCY_CHECK Represents literal INTEGRATION_CONSISTENCY_CHECK of formal type definition T_StartupState.

FR_STARTUP_COLDSTART_GAP Represents literal COLDSTART_GAP of formal type definition T_StartupState.

FR_STARTUP_EXTERNAL_STARTUP Represents literal EXTERNAL_STARTUP of formal type definition T_StartupState.

Description: This formal definition refers to the description of type T_StartupState in chapter 2.2.1.3 POC status of [12].

>>

NOTE

Fr_StartupStateType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00510]

<< Name: Fr_POCStatusType

Type: Structure

Element: boolean CHIHaltRequest --

boolean ColdstartNoise --

Fr_ErrorModeType ErrorMode --

boolean Freeze --

Fr_SlotModeType SlotMode --

Fr_StartupStateType StartupState --

Fr_POCStateType State --

Fr_WakeupStatusType WakeupStatus --

boolean CHIReadyRequest --

Description: This formal definition refers to the description of type T_POCStatus in chapter 2.2.1.3 POC status of [12].

>>

NOTE

Fr_POCStatusType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00511]

<< Name: Fr_TxLPduStatusType

Type: Enumeration

Range: FR_TRANSMITTED (=0) LPdu has been transmitted

FR_NOT_TRANSMITTED LPdu has not been transmitted

Description: These values are used to determine whether a LPdu has been transmitted or not.

>>

NOTE

Fr_TxLPduStatusType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00512]

<< Name: Fr_RxLPduStatusType

Type: Enumeration

Range: FR_RECEIVED (=0) LPdu has been received

FR_NOT_RECEIVED LPdu has not been received

FR_RECEIVED_MORE_DATA_AVAILABLE LPdu has been received. More instances of this LPdu are available (FIFO usage).

Description: These values are used to determine if a LPdu has been received or not.

>>

NOTE

Fr_RxLPduStatusType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00514]

<< Name: Fr_ChannelType

Type: Enumeration

Range: FR_CHANNEL_A (=0) Refers to channel A of a CC.

FR_CHANNEL_B Refers to channel B of a CC.

FR_CHANNEL_AB Refers to both channels (A and B) of a CC.

Description: The values are used to reference channels on a CC.

>>

NOTE

Fr_ChannelType shall be provided by the Fr_GeneralTypes.h file.

[SWS_Fr_00390]

<< API function: Dem_ReportErrorStatus

Description: Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function. OBD Events Suppression shall be ignored for this computation. >>

NOTE

This requirement defines interface which is required to fulfill core functionality of the module.

[SWS_Fr_00391]

<< API function: Det_ReportError

Description: Service to report development errors.

API Function: SchM_Enter_Fr

API Function: SchM_Exit_Fr >>

NOTE

This requirement defines interfaces that are required to fulfill an optional functionality of the module. SchM functionality is not used by the FR driver.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number IM35FRASR4.2 Rev002R1.0.0
Revision 1.0.0