MICROSAR Fr ERay

Technical Reference

Communication Controller E-Ray V850

Version 1.41

Authors	Juergen Schaeffer, Mario Kunz, Sebastian Gärtner, Sebastian Schmar, Oliver Reineke, Roland Hocke, Matthias Müller
Status	Released

1 Document Information

1.1 History

Author	Date	Version	Remarks
Roland Hocke	2013-05-14	1.34	Creation and content copy from MSR3 document
	2013-06-26	1.35	Remove obsolete MTS
Roland Hocke	2014-01-29	1.37	Buffer alignment hints
Matthias Müller	2015-02-20	1.38	Renaming of document name
Matthias Müller	2015-09-28	1.39	Description of Endinit and Protected Register Access
Matthias Müller	2015-11-27	1.40	ESCAN00080370 The usage of the Appl_TricoreAurixInit() function is not described in the TechRef SafeBSW limitations
Matthias Müller	2017-02-01	1.41	Added hint about the specifics of RH850 P1X

Table 1-1 History of the document

1.2 Reference Documents

No.	Title	Version
[1]	AUTOSAR_SWS_DevelopmentErrorTracer.pdf	3.2.0
[2]	AUTOSAR_SWS_DiagnosticEventManager.pdf	4.2.0
[3]	TechnicalReference_Fr.pdf	1.0 or later
[4]	E-Ray_Errata_Sheet_20100215.pdf and later	REL20100215 and later
[5]	TechnicalReference_SchM.pdf	2.5 or later

Table 1-2 Reference documents

1.3 Scope of the Document

This technical reference describes the specific use of the FlexRay ERay driver software. It supplements the general FlexRay driver technical reference [3].

Please note
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Illustrations

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Tables

3 Component History

Component Version	New Features
3.0	Initial version
3.3	ISR routines in separate file
3.9	Fifo added
3.11	Read and verification of the FlexRay configuration
3.13	Added Fr_GetChannelStatus,Fr_GetClockCorrection, Fr_GetSyncFrameList, Fr_DisableLPdu and Fr_ReconfigLPdu
3.14	Optimization

Table 3-1 Component history

4 Introduction

This document describes the functionality, API and configuration of the AUTOSAR BSW module Fr ERay that is specific for the communication controller E-Ray and the used micro controller. The basic functionality API architecture is already described within the document [3].

5 Functional Description

Please refer to [3].

6 Integration

This chapter gives necessary information for the integration of the MICROSAR Fr ERay into an application environment of an ECU. It describes ERay specific issues in addition to [3].

6.1 Scope of Delivery

The delivery of the Fr ERay contains the files which are described in the chapters 6.1.1 and 6.1.2:

6.1.1 Static Files

File Name	Description
Fr_ERay.h	This file replaces the file Fr_<CC>.h listed in the document [3].

Table 6-1 Static files

6.1.2 Dynamic Files

No additional dynamic files. Please refer to [3].

6.2 Compiler Abstraction and Memory Mapping

The following table contains the memory section names and the compiler abstraction definitions defined for the Fr ERay and illustrates their assignment among each other.

Compiler Abstraction Definitions / Memory Mapping Sections	[Abstraction Name]	FR_CODE	FR_VAR_NOINIT	FR_VAR_NOINIT_FAST	FR_VAR_FRM	FR_CONST	FR_PBCFG	FR_APPL_CODE	FR_CODE_ISR
[Section Name]									
FR_START_SEC_CODE FR_STOP_SEC_CODE		■							
FR_START_SEC_VAR_NOINIT_UNSPECIFIED FR_STOP_SEC_VAR_NOINIT_UNSPECIFIED			■						
FR_START_SEC_VAR_FAST_NOINIT_UNSPECIFIED FR_STOP_SEC_VAR_FAST_NOINIT_UNSPECIFIED				■					

FR_START_SEC_CONST_UNSPECIFIED FR_STOP_SEC_CONST_UNSPECIFIED						■	■		
FR_START_SEC_PBCFG_ROOT FR_ STOP _SEC_PBCFG_ROOT							■		
FR_START_SEC_PBCFG FR_STOP_SEC_PBCFG							■		
FR_START_SEC_CONST_32BIT FR_STOP_SEC_CONST_32BIT						■	■		
FR_START_SEC_CONST_16BIT FR_STOP_SEC_CONST_16BIT						■	■		
FR_START_SEC_CONST_8BIT FR_STOP_SEC_CONST_8BIT						■	■		
FR_APPL_START_SEC_CODE FR_APPL_STOP_SEC_CODE								■	
FR_START_SEC_CODE_ISR FR_STOP_SEC_CODE_ISR									■

Table 6-2 Compiler abstraction and memory mapping

Caution
Please ensure that the define FR_PBCFG get the same value as FR_ CONST in configuration type “Pre-compile Configuration” or “Link-time Configuration”

6.3 Critical Sections

To ensure data consistency and a correct function of the Fr ERay the exclusive area `FR_EXCLUSIVE_AREA_0` has to be provided during the integration.

Considering the timing behavior of your system (e.g. depending on the CPU load of your system, priorities and interruptibility of interrupts and OS tasks and their jitter and delay times) the integrator has to choose and configure a critical section solution in such way that it is ensured that the API functions do not interrupt each other. You can find a set of rules below which describes whether an exclusive area is needed by the Fr ERay or not.

The `FR_EXCLUSIVE_AREA_0` has to be used if at least one of the following rules hold true:

- It is possible that the function `Fr_TransmitTxLPdu` is interrupted by the function `Fr_TransmitTxLPdu` itself. E.g. at execution of method `Fr_TransmitTxLPdu`, that was triggered by upper layer because the attribute `FrIfImmediate` is set, is interrupted by the job list execution of FrIf that contains a call of the method `Fr_TransmitTxLPdu`, or vice versa.
- At enabled feature “Reconfig LPdu Support” it is possible that the functions `Fr_DisableLPdu` or `Fr_ReconfigLPdu` are interrupted by the function `Fr_TransmitTxLPdu` and vice versa. E.g. at execution of the method `Fr_DisableLPdu` or `Fr_ReconfigLPdu` is interrupted by the job list execution of FrIf that contains a call of the method `Fr_TransmitTxLPdu`.

The recommended implementation for the `FR_EXCLUSIVE_AREA_0` of the component Fr ERay depends on the integration context of the job list execution.

The `FR_EXCLUSIVE_AREA_0` shall disable/enable the FlexRay timer interrupt or call `SuspendAllInterrupts()` and `ResumeAllInterrupts()` in case the job list execution is done at interrupt context to ensure data consistency.

The `FR_EXCLUSIVE_AREA_0` shall call `SuspendOSInterrupts()` and `ResumeOSInterrupts()` in case the case the job list execution is done at task context to ensure data consistency. Alternative the undesired task activation can be prevent by implement OS resource lock at the involved tasks.

The Fr ERay supports one of the following two alternatives as implementation for these exclusive areas depending on your questionnaire:

- The BSW Scheduler (refer to [5] for a detailed description)
- the Vector Standard Library (VStdLib)

The VStdLib offers the possibility of mapping the interrupt handling to OS services or to user defined functions. In the first case interrupt handling is done by the OS, in the second case the user has to take care by providing corresponding functions.

6.4 General Integration notes

A workaround for errata #27 in [4] is implemented in module FrSm. If no FrSm is used, the application has to implement a workaround.

6.4.1 Calculation of Timeout Loops

Please refer to [3] for generically description of this feature. The following text describes the worst case value for given platform.

The worst case time value depends on the “BCLK“ speed. The time value can be calculated as follow:

Worst case time = 1/BCLK * 1329

e.g.

Worst case time = 1/64MHz * 1329 = 20,7 micro seconds

The worst case time wait is reached with the “Timeout Duration Factor” that can be set within configuration. The “Timeout Duration Factor” value for the given example worst case time of 20,7 micro seconds is 380 in case the internal system clock of the micro controller is set to 128MHz.

Info

The timeout expiration is controlled by simply while loop. You can adapt the “Timeout Duration Factor” in case you use other BCLK or internal system clock values. The new value of the “Timeout Duration Factor” can be find out by measuring the execution time of the while loop.

```
while (timeoutCounter < "Timeout Duration Factor")
{
  timeoutCounter++;
}
```

6.5 Integration notes for NEC V850

6.5.1 ERay Base Address

The base address for V85x CC depends on the device and can be found in Table 6-3.

Device name	CC base address
µPD70F3441	0x0F800000
µPD70F3461(A)	0x0F800000
µPD70F3501	0xFF580000
µPD70F4010	0xFF580000

Table 6-3 CC base address

6.5.2 FlexRay Memory buffer number

The ERay CC at V85x can address a maximum number of 128 message buffers.

6.5.3 FlexRay Memory Buffer size

The available FlexRay Message RAM for V85x CC depends on the device and can be found in Table 6-4.

Device name	Message RAM
µPD70F3441	6 kB
µPD70F3461(A)	8 kB
µPD70F3501	8 kB
µPD70F4010	8 kB

Table 6-4 Message RAM

6.5.4 Interrupt routines

The Fr ERay module configures the CC with its interrupt registers. But the FlexRay interrupts are not enabled on V850 interrupt controller by Fr ERay.

The application has to enable the FlexRay interrupts on the V850 interrupt controller. As an example a small function is given within the next lines:

Example

```
void EnableInterrupts(void)
{
  ECU_FRIC0 = ECU_FRIC0 & (~0x40);
  ECU_FRIC1 = ECU_FRIC1 & (~0x40);
  ECU_FRIC2 = ECU_FRIC2 & (~0x40);
}
```

The interrupt function can be called directly from OS or integration of V850. Therefore the interrupt routines

- Fr_IrqLine0 or Fr_IrqLine1 line selected at optional parameter “Line select for status interrupts”. By default Fr_IrqLine0 is used. This interrupt is only necessary if configuration attribute “Cycle Start Interrupt” or ECUC Parameter FrCycleStartInterrup is enabled. Otherwise this interrupt does need not be enabled.
- Fr_IrqTimer0

are implemented in file Fr_Irq.c. They call the corresponding interrupt function out of which the application callout functions are called. To save resources, Fr_IrqTimer0 directly calls the application callout function or FrIf, according to configuration.

6.5.5 Timer interrupt

At V850 with ERay it is not possible to switch off FlexRay timer interrupts on ERay. The configuration of Eray allows disconnecting the timer interrupt from line 0 or line 1 interrupt. This is the default configuration of FR. But there is an additional interrupt line for timer interrupt which is not configurable by Eray registers. This additional interrupt line is connected directly to V850 interrupt controller.

So enabling or disabling the timer interrupt on V850 interrupt controller has to be done by the application by configuration of V850 interrupt controller.

6.5.6 Buffer alignment

The used FlexRay buffers shall be aligned to 16 bit addresses. This is important for message buffers which are sent with FrIf mode immediately or no FrIf is used. If FrIf is used, FrIf automatically aligns the buffers to 32 bit. The used FlexRay buffers shall be aligned to 16 bit addresses.

In case this is not possible, FlexRay driver can align unaligned message itself. This functionality needs more runtime to transmit unaligned messages and needs to check every message for its alignment. It can be activated with a define switch in userconfig-file.

Example

```
#define FR_NONALIGNED_SRCMEMORY
```

6.6 Integration notes for RH850

The ERay CC at RH850 can address a maximum number of 128 message buffers. The maximum amount of usable message buffers depends heavily on the size of the frames, and thus, on the specific configuration.

6.6.1 Buffer alignment

The used FlexRay buffers shall be aligned to 16 bit addresses. This is important for message buffers which are sent with FrIf mode immediately or no FrIf is used. If FrIf is used, FrIf automatically aligns the buffers to 32 bit. The used FlexRay buffers shall be aligned to 16 bit addresses.

6.6.2 Specifics of RH850 P1x

We discovered a problem with RH850 derivatives of type P1x. With these chips our FlexRay might not work as expected. We observed an abortion of the initialization sequence. The result is that no FlexRay communication can be established (ESCAN00093807).

We observed this behavior on the following P1x:

- P1H-C 1331
- P1H-C 1372

> P1M 1312

There is a workaround that can be controlled by defining FR_WAIT_STATES via a user config file. If FR_WAIT_STATES is not defined its default value will be 5 which means 5 ms. A user config file can look like that:

```
#define FR_WAIT_STATES 3 /* number means time in ms */
```

7 API Description

7.1 Type Definitions

The types defined by the Fr ERay are described in this chapter.

The Fr ERay does not define specific type definitions.

7.2 Interrupt Service Routines provided by Fr ERay

For general description please refer to [3] and the respective chapters under “Functional Description”.

Prototype	
`FUNC(void, FR_CODE) ()`	
Parameter	

Return code	

Functional Description	
Executes the status interrupts of ERay. By default cycle start interrupt are used by Fr ERay to synchronize Application with callback function `ApplFr_ISR_CycleStart`.	
Particularities and Limitations	
> Particularities, limitations, post-conditions, pre-conditions > An interrupt splitting is used. Fr ERay is initialized and interrupts are enabled. > Configuration attribute “Cycle Start Interrupt” or ECUC Parameter FrCycleStartInterrupt is enabled.	

Table 7-1 Fr_IrqLine0/Fr_IrqLine1

Prototype	
`FUNC(void, FR_CODE) ()`	
Parameter	

Return code	

Functional Description
Executes the Timer0 interrupt of ERay. By default this interrupt is used by Fr ERay. It also disables the interrupt.
Particularities and Limitations
> Particularities, limitations, post-conditions, pre-conditions > A separate interrupt for Timer0 is called by hardware and interrupt splitting is used. Fr ERay is initialized and interrupts are enabled.

Table 7-2 Fr_IrqTimer0

7.3

7.4 Services provided by Fr ERay and differs from standard

The Fr ERay API consists of services, which are realized by function calls. General services are described in [3]. The following services differ from standard behavior.

7.4.1 Fr_EnableAbsoluteTimerIRQ

Fr ERay module does not enable or disable ERay timer interrupts or assign them to an ERay interrupt line. The application has to enable or disable the ERay timer interrupt at the controller specific interrupt controller.

7.4.2 Fr_DisableAbsoluteTimerIRQ

Please refer to 7.4.1

7.5 Services used by Fr ERay

General services are described in [3].

Depending on configuration services used by Fr ERay are listed in next table. For details about prototype and functionality refer to the documentation of the providing component.

Configuration/Precondition	Component	API
Application callback at timer 0 interrupt and BSM FrIf is used	FrIf	FrIf_JobListExec_0

Table 7-3 Configuration dependent services used by Fr ERay

7.6 Callback Functions

There is no Fr ERay specific callback function available.

7.7 Configurable Interfaces

7.7.1 Notifications

There is no Fr ERay specific notification function available.

7.7.2 Callout Functions

There is no Fr ERay specific callout function available.

8 Configuration

In the Fr ERay the attributes can be configured with the following methods:

> Configuration in DaVinci Configurator 5; for a detailed description see 8.2.

8.1 Hardware Fifo

This platform does not support Hardware Fifo support. Therefore it cannot be configured.

8.2 Configuration with DaVinci Configurator 5

The Fr ERay is configured with the help of the configuration tool DaVinci Configurator 5. For general configuration information please refer to [3].

9 AUTOSAR Standard Compliance

9.1 Deviations

For general deviations please refer to [3] and [4].

9.1.1 Fr_GetSyncFrameList

Eray CC always receives a transmitted SYNC frame on both channels, even if a channel is not used. The transmission of a SYNC frame depends on configuration.

9.1.2 Fr_GetPOCStatus

The value for Fr_POCStatusPtr->SlotMode is derived out of the Eray bit CCSV.SLT.

At Eray versions higher than 1.0.1 the Eray Bit CCSV.SLT (single slot mode) can not be read in Poc states: “Config, Default Config, Halt” and always return 0. In all other POC states the Value is according configuration. For a detailed description please refer to [4].

9.2 Additions/ Extensions

Please refer to [3].

9.3 Limitations

Please refer to [3].

For the hardware limitations described in [4], workarounds are implemented or configuration warnings with suggestions for solutions are displayed at configuration time.

10 Glossary and Abbreviations

10.1 Glossary

Term	Description
EAD	Embedded Architecture Designer; generation tool for MICROSAR components
Cfg 5	DaVinci Configurator 5. Configuration tool for Microsar 4

Table 10-1 Glossary

10.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CC	FlexRay Communication Controller
E-Ray	Specific implementation of a FlexRay Communication Controller
DBA	Direct Buffer Access
DEM	Diagnostic Event Manager
DET	Development Error Tracer
EAD	Embedded Architecture Designer
ECU	Electronic Control Unit
HIS	Hersteller Initiative Software
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
PPort	Provide Port
RPort	Require Port
RTE	Runtime Environment
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification

Table 10-2 Abbreviations

11 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector-informatik.com