TC1766

32-Bit Single-Chip Microcontroller

Microcontrollers

Never stop thinking

Edition 2008-04

**Published by Infineon Technologies AG,
81726 München, Germany**

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”). With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

TC1766

32-Bit Single-Chip Microcontroller

Microcontrollers

Never stop thinking

TC1766 Documentation Addendum

Revision History: V2.0 2008-04

Previous Versions: V1.0, V1.1, V1.2	
Page	Subjects (major changes since last revision)
–	This is the first release that refers to the TC1766 User's Manual **V2.0**, July 2007

Trademarks

TriCore® is a trademark of Infineon Technologies AG.

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

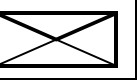

Table of Contents

Table of Contents

1 Introduction

This document describes corrections, changes, and improvements for the TC1766 User's Manual V2.0 2007-07, the System Units (Volume 1) and the Peripheral Units (Volume 2). These changes will be included in the next update of the User's Manual.

The referenced documents to this addendum are located at the Internet page:

- www.infineon.com/tc1766
- TC1766 User's Manual System and Peripheral Units , V2.0, July 2007

2 User's Manual - System Units Part (Volume 1)

This section describes the updates for the System Units of the User's Manual.

Page 3-19

Section 3.2.2.5 "Setting up the PLL after Reset" must be updated. Points 1. to 7. of this section must be replaced by the following 9 points (red text indicates the changes):

1. Wait until the oscillator is running (OSC_CON.OSCR = 1)
2. Selection of the VCO Bypass Mode (PLL_CLC.VCOBYP = 1)
3. Selecting the VCO band by programming PLL_CON.VCOSEL
4. Program the desired P, N and K values (PDIV, NDIV, and KDIV bit fields of register PLL_CLC) **to get a temporary f_{CPU} value which is lower than the target frequency.**[1)]
5. Connect the oscillator to the PLL (PLL_CLC.OSCDISC = 0)
6. Wait until the PLL becomes locked (PLL_CLC.LOCK = 1)
7. Disable the VCO Bypass Mode (PLL_CLC.VCOBYP = 0)
8. **Wait for typically 5ms until supply ripple caused by increased supply current is faded away.**
9. **Decrease K value step by step, with wait phases in between, until the targeted f_{CPU} is reached.**

Page 6-12

The first line in the description for bit field LEDAT[31:0] must be corrected into "LMB **Data** Bits [31:0]" instead of "LMB Bus Address Bits [31:0]".

The first line in the description for bit field LEDAT[63:32] must be corrected into "LMB **Data** Bits [**63:32**]" instead of "LMB Bus Address Bits [31:0]".

Page 7-34

The note at the end of the page must be erased: "Note: After the detection …".

Page 7-35

The second sentence of the first paragraph must be erased: "With this features, problematic …"

Paragraphs 3 to 6 must be erased: "Since problematic Flash array bits …" until "… of Flash cells is close to the zero state".

1) K value selection should result in a small change of f_{CPU} when bypass mode is left to reduce supply ripple

Page 7-36

The following section must be added to the top of page:

7.2.8.3 Application Hints Flash Error Handling

The previous sections described shortly the functionality of "error indicating" bits in the flash status register FSR. This section gives recommendations how these should be handled by customer software.

PFOPER/DFOPER "Operation Error"

Fault conditions:

ECC double-bit error detected in Flash microcode SRAM during a program or erase operation in PFlash or DFlash. This can be a transient event due to alpha-particles or illegal operating conditions or it is a permanent error due to a hardware defect. This situation will practically not occur. These bits can also be set in case of uncritical errors that don't affect Flash operation. This case is much more likely than real operation errors.

Attention: these bits can also be set during startup.

New state:

If triggered by Flash operation this is aborted, the BUSY flag is cleared and read mode is entered.

Proposed handling by software:

The DFOPER and PFOPER flags should be ignored for the possibility of flagging an error which is not related to a Flash write or erase operation. By checking the result of an operation (e.g. checking programmed data) a real operation error can be easily determined. When a real operation error is determined a reset should be applied before trying the operation again.

Note: Even when the flag is ignored it is recommended to clear it. Otherwise all following operations — including "sleep" — could trigger an interrupt even when they are successful.

VER "Verification Error"

Fault conditions:

This flag is a warning indication and not an error. It is set when a program or erase operation was completed but with a suboptimal result. This bit is already set when only a single bit is left over-erased or weakly programmed which would be corrected by the ECC anyhow.

However excessive VER occurrence can be caused by operating the Flash out of the specified limits, e.g. incorrect voltage or temperature. A VER after programming can also be caused by programming a page whose sector was not erased correctly (e.g. aborted erase due to power failure).

Under correct operating conditions a VER after programming will practically not occur. A VER after erasing is not unusual.

New state:

No state change. Just the bit is set.

Proposed handling by software:

This bit can be ignored. It should be cleared with "Clear Status" or "Reset to Read". In-spec operation of the Flash memory must be ensured.

If the application allows (timing and data logistics), a more elaborate procedure can be used to get rid of the VER situation:

- VER after program: erase the sector and program the data again. This is only recommended when there are more than 3 program VERs in the same sector. When programming the DFlash in field (EEPROM emulation) ignoring program VER is normally the best solution because its most likely cause are violated operating conditions. Take care that never a sector is programmed in which the erase was aborted. In the EEPROM emulation the algorithm must ensure this e.g. by programming a marker after finishing successfully the erase.
- VER after erase: the erase operation can be repeated until VER disappears. Repeating the erase more than 3 times consecutively for the same sector is not recommended. After that it is better to ignore the VER, program the data and check its readability. Again for EEPROM emulation its most likely cause are violated operating conditions. Therefore it is recommended to repeat the erase at most once or ignore it altogether.

For optimizing the quality of Flash programming see the following section about handling single-bit ECC errors.

Note: Even when this flag is ignored it is recommended to clear it. Otherwise all following operations — including "sleep" — could trigger an interrupt even when they are successful.

PFSBER/DFSBER "Single-Bit Error"

Fault conditions:

When reading data or fetching code from PFlash or DFlash the ECC evaluation detected a single-bit error ("SBE") which was corrected.

This flag is a warning indication and not an error. A certain amount of single-bit errors must be expected because of known physical effects.

New state:

No state change. Just the bit is set.

Proposed handling by software:

This flag can be used to analyze the state of the Flash memory. During normal operation it should be ignored. In order to count single-bit errors it must be cleared by "Clear Status" or "Reset to Read" after each occurrence.

Usually it is sufficient after programming data to compare the programmed data with its reference values ignoring the SBE bits. When there is a comparison error the sector is erased and programmed again.

When programming the PFlash (end-of-line programming or SW updates) customers can further reduce the probability of future read errors by performing the following check after programming:

- Change the read margin to "high margin 0".
- Verify the data and count the number of SBEs.
- When the number of SBEs exceeds a certain limit (e.g. 10 in 2 MByte) the affected sectors could be erased and programmed again.
- Repeat the check for "high margin 1".
- Each sector should be reprogrammed at most once, afterwards SBEs can be ignored.

In case of EEPROM emulation using DFlash the verification of programmed data should be done with the normal read level and SBEs should be ignored. When a comparison error is found the sector can usually not be erased because it contains active data in other pages. The emulation algorithm can mark the affected page as invalid and program the data to a following page. As always the number of consecutive repetitions should be limited (e.g. to 3) as protection against violated operating conditions.

To keep the EEPROM emulation alive even when wordline (two consecutive pages, even followed by odd pages) oriented fails occur (e.g. due to over-cycling) the algorithm can implement the following scheme for highest possible robustness:

- Before programming a page save the content of the other page on the same wordline in SRAM.
- Program the new page and compare the content with the reference data. This can be done with normal read margins. Ignore SBEs.
- If the data comparison fails program this page and the saved content of the other page to a different wordline.
- This procedure can be repeated if the data comparison fails again. The number of repetitions should be limited (e.g. to 3) in case the programming fails because of out-of-spec operating conditions.

Due to the specificity of each application the appropriate usage and implementation of these measures (together with the more elaborate VER handling) must be chosen according to the context of the application.

Page 8-17

With the addition of possible memory access for OVRAM, Table 8-5 must be replaced by the below table:

Table 8-5 Possible Memory Accesses

Memory		Bit	Byte		Half-word		Word		Double-word	
		rmw	r	w	r	w	r	w	r	w
PMI[1]	SPRAM	✔	✔	–	✔	✔	✔	✔	✔	✔
DMI[1]	LDRAM	✔	✔	✔	✔	✔	✔	✔	✔	✔
PMU	ROM	–	✔	–	✔	–	✔	–	✔	–
	PFLASH	–	✔	–	✔	–	✔	✔	✔	✔
	DFLASH	–	✔	–	✔	–	✔	✔	✔	✔
	OVRAM[1]	–	✔	✔	✔	✔	✔	✔	✔	✔
PCP[2]	CRAM	–	–	–	–	–	✔	✔	✔	✔
	PRAM	–	–	–	–	–	✔	✔	✔	✔

1) The module also supports LMB 2-Word and 4-Word Block read and write accesses.

2) The module also supports FPI 4-Word and 8-Word Block read and write accesses.

Page 9-47

Table 9-13 should be replaced as below:

Table 9-13 Port 3 Functions

Port Pin	I/O	Pin Functionality	Associated Reg./ I/O Line	Port I/O Control Select.	
				Reg./Bit Field	Value
P3.0	I	General-purpose input	P3_IN.P0	P3_IOCR0.PC0	$0XXX_B$
		ASC0 input	RXD0A		
	O	General-purpose output	P3_OUT.P0		$1X00_B$
		ASC0 output (Synchronous Mode only)[1]	RXD0A		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.1	I	General-purpose input	P3_IN.P1	P3_IOCR0.PC1	$0XXX_B$
		SCU input	OSCBYP		
	O	General-purpose output	P3_OUT.P1		$1X00_B$
		ASC0 output[1]	TXD0A		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.2	I	General-purpose input	P3_IN.P2	P3_IOCR0.PC2	$0XXX_B$
		SSC0 input (Slave Mode)	SCLK0		
	O	General-purpose output	P3_OUT.P2		$1X00_B$
		SSC0 output (Master Mode)[1]	SCLK0		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.3	I	General-purpose input	P3_IN.P3	P3_IOCR0.PC3	$0XXX_B$
		SSC0 input (Master Mode)	MRST0		
	O	General-purpose output	P3_OUT.P3		$1X00_B$
		SSC0 output (Slave Mode)[1]	MRST0		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$

Table 9-13 Port 3 Functions (cont'd)

Port Pin	I/O	Pin Functionality	Associated Reg./ I/O Line	Port I/O Control Select.	
				Reg./Bit Field	Value
P3.4	I	General-purpose input	P3_IN.P4	P3_IOCR4.PC4	$0XXX_B$
		SSC0 input (Slave Mode)	MTSR0		
	O	General-purpose output	P3_OUT.P4		$1X00_B$
		SSC0 output (Master Mode)[1]	MTSR0		$1X01_B$
					$1X10_B$
		Reserved [2]	–		$1X11_B$
P3.5	I	General-purpose input	P3_IN.P5	P3_IOCR4.PC5	$0XXX_B$
	O	General-purpose output	P3_OUT.P5		$1X00_B$
		SSC0 output	SLSO00		$1X01_B$
		SSC1 output	SLSO10		$1X10_B$
		SSC0 and SSC1 output	SLSO00 AND SLSO10[3]		$1X11_B$
P3.6	I	General-purpose input	P3_IN.P6	P3_IOCR4.PC11 x	$0XXX_B$
	O	General-purpose output	P3_OUT.P6		$1X00_B$
		SSC0 output	SLSO01		$1X01_B$
		SSC1 output	SLSO11		$1X10_B$
		SSC0 and SSC1 output	SLSO01 AND SLSO11[3]		$1X11_B$
P3.7	I	General-purpose input	P3_IN.P7	P3_IOCR4.PC7	$0XXX_B$
		SSC0 input	$\overline{\text{SLSI0}}$		
	O	General-purpose output	P3_OUT.P7		$1X00_B$
		SSC0 output	SLSO02		$1X01_B$
		SSC1 output	SLSO12		$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.8	I	General-purpose input	P3_IN.P8	P3_IOCR8.PC8	$0XXX_B$
	O	General-purpose output	P3_OUT.P8		$1X00_B$
		SSC0 output	SLSO06		$1X01_B$
		ASC1 output	TXD1A		$1X10_B$
		Reserved[2]	–		$1X11_B$

Table 9-13 Port 3 Functions (cont'd)

Port Pin	I/O	Pin Functionality	Associated Reg./ I/O Line	Port I/O Control Select.	
				Reg./Bit Field	Value
P3.9	I	General-purpose input	P3_IN.P9	P3_IOCR8.PC9	$0XXX_B$
		ASC1 input	RXD1A		
	O	General-purpose output	P3_OUT.P9		$1X00_B$
		ASC1 output (Synchronous Mode only)[1]	RXD1A		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.10	I	General-purpose input	P3_IN.P10	P3_IOCR8.PC10	$0XXX_B$
		SCU input	REQ0		
	O	General-purpose output	P3_OUT.P10		$1X00_B$
		Reserved[2]	–		$1X01_B$
					$1X10_B$
					$1X11_B$
P3.11	I	General-purpose input	P3_IN.P11	P3_IOCR8.PC11	$0XXX_B$
		SCU input	REQ1		
	O	General-purpose output	P3_OUT.P11		$1X00_B$
		Reserved[2]	–		$1X01_B$
					$1X10_B$
					$1X11_B$
P3.12	I	General-purpose input	P3_IN.P12	P3_IOCR12.PC12	$0XXX_B$
		CAN node 0 receive input 0 CAN node 1 receive input 1	RXDCAN0		
		ASC0 input	RXD0B		
	O	General-purpose output	P3_OUT.P12		$1X00_B$
		ASC0 output (Synchronous Mode only)[1]	RXD0B		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$

Table 9-13 Port 3 Functions (cont'd)

Port Pin	I/O	Pin Functionality	Associated Reg./ I/O Line	Port I/O Control Select.	
				Reg./Bit Field	Value
P3.13	I	General-purpose input	P3_IN.P13	P3_IOCR12.PC13	$0XXX_B$
	O	General-purpose output	P3_OUT.P13		$1X00_B$
		CAN node 0 output	TXDCAN0		$1X01_B$
		ASC0 output	TXD0B		$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.14	I	General-purpose input	P3_IN.P14	P3_IOCR12.PC14	$0XXX_B$
		CAN node 1 receive input 0 CAN node 0 receive input 1	RXDCAN1		
		ASC1 output	RXD1B		
	O	General-purpose output	P3_OUT.P14		$1X00_B$
		ASC1 output (Synchronous Mode only)[1]	RXD1B		$1X01_B$
					$1X10_B$
		Reserved[2]	–		$1X11_B$
P3.15	I	General-purpose input	P3_IN.P15	P3_IOCR12.PC15	$0XXX_B$
	O	General-purpose output	P3_OUT.P15		$1X00_B$
		CAN node 1output	RXDCAN1		$1X01_B$
		ASC1 output	TXD1B		$1X10_B$
		Reserved[2]	–		$1X11_B$

1) The ALT1 and ALT2 for this pin are connected together. There are no dependencies. Either one can be chosen.

2) The port I/O control values P3_IOCRx.Py that are assigned to this reserved alternate output control selection should not be used. Otherwise, unpredictable output port line behavior may occur.

3) The AND-gate of ALT3 is located in the GPIO module.

Page 10-109

In Table 10-15, the number of clock cycles for the stated instructions must be corrected as below:

Table 10-15 Instruction Timing

Instruction	Number of Clock Cycles	Comments	Notes
PRAM Access			
MCLR.PI	**6**	–	–
MSET.PI	**6**	–	–
ST.PI	**4**	–	–
XCH.PI	**5**	–	–
Complex Maths			
MSETP.U	**11**	–	–

On Page 10-111, the last row of footnote 7 must be changed to "32 x 32 bit multiply requires instruction MINIT + 4 x MSTEP.U = 1+4 x **11** = **45** cycles".

Page 11-35

The first bulleted point must be changed to "The activation of the interrupt corresponding to the current active channel 0n using the Interrupt Pointer defined in CHICR0n.**INTP**.".

3 User's Manual - Peripheral Units Part (Volume 2)

This section describes the updates for the Peripheral Units of the User's Manual.

Page 17-37

Table 17-8 should be replaced as below:

Table 17-8 ASC0/ASC1 I/O Control Selection and Setup

Module	Port Lines	PISEL Register	Input/Output Control Register Bits[1)]	I/O
ASC0	P3.0/RXD0A	ASC0_PISEL.RIS = 0	P3_IOCR0.PC0 = $0XXX_B$	Input
	P3.0/RXD0A	–	P3_IOCR0.PC0 = $1X01_B$ or $1X10_B$	Output [2)]
	P3.12/RXD0B	ASC0_PISEL.RIS = 1	P3_IOCR12.PC12 = $0XXX_B$	Input
	P3.12/RXD0B	–	P3_IOCR12.PC12 = $1X01_B$ or $1X10_B$	Output [2)]
	P3.1/TXD0A	–	P3_IOCR0.PC1 = $1X01_B$or $1X10_B$	Output
	P3.13/TXD0B	–	P3_IOCR12.PC13 = $1X10_B$	Output
ASC1	P3.9/RXD1A	ASC1_PISEL.RIS = 0	P3_IOCR8.PC9 = $0XXX_B$	Input
	P3.9/RXD1A	–	P3_IOCR8.PC9 = $1X01_B$or $1X10_B$	Output [2)]
	P3.14/RXD1B	ASC1_PISEL.RIS = 1	P3_IOCR12.PC14 = $0XXX_B$	Input
	P3.14/RXD1B	–	P3_IOCR12.PC14 = $1X01_B$ or $1X10_B$	Output [2)]
	P3.8/TXD1A	–	P3_IOCR8.PC8 = $1X10_B$	Output
	P3.15/TXD1B	–	P3_IOCR12.PC15 = $1X10_B$	Output

1) For possible PCx bit field combinations, see Table 17-9.

2) Applicable in Synchronous Mode only.

Page 18-16

The expression "SLSOx" must be replaced three times by "SLSO**n**".

In the numbered paragraph "3." the expression "SS**S**OTC.INACT" must be changed to "SSOTC.INACT".

Page 18-24

The description for the Module Revision Number bit field for the ID register must be changed to "MODREV defines the module revision number. The value of a module revision starts with $\mathbf{10}_H$(first revision)."

Page 18-34

The bit field description for bit TB_VALUE must begin with "Register **TB** stores the data value ....".

Page 18-42

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed to "**r**".

Page 19-4

The item "- Programmable upstream data frame length (16 or 12 bits)" must be moved from the second bullet as third item under the third bullet "Low-speed asynchronous serial reception on upstream channel".

Page 19-13

in the last bullet line on the bottom of the page "DDL" must be changed to "D**C**L".

Page 19-18

"DSS.DC" must be changed to "DSS.**PFC**" at the first sentence of the Passive Frame Counter in Data Repetition Mode section.

Page 19-19

In the paragraph above Figure 19-12 "(ENSELL=0)" must be changed to "(ENSEL**H**=0)".

Page 19-25

In Table 19-6 column USR.URR fourth line "010_B" must be changed to "$01\mathbf{1}_B$".

Page 19-26

In the first paragraph, "OCSR.URR" must be changed to "**USR**.URR".

Page 19-34

In the paragraph above Figure 19-26 "ISC.SRDI and ISC.CRDI" must be changed to "ISC.S**U**RDI and ISC.C**U**RDI".

Figure 19-26 must be updated with the following figure (output signal on the right side):

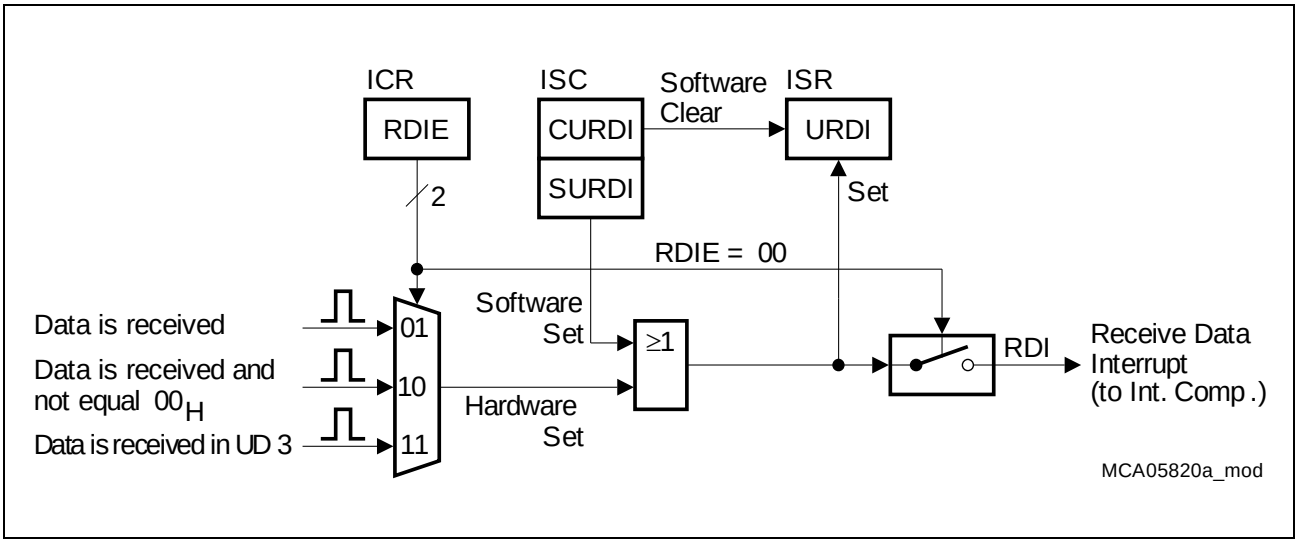

Figure 19-26 Receive Data Interrupt Control

Page 19-47

The paragraph on the top of the page must be changed to:
"The bit fields of the Downstream Select Data Source High Register DSDS**H** determine the data source for each bit in shift register SR**H**."

The register description table must be changed as follows:

SHx (x = 0-15)	[2*x+1: 2*x]	rw	**Select Source for SRH** SHx determines which data source is used for the shift register bit SR**H**[x] during data frame transmission. 00_B SR**H**[x] is taken from data register DD.DD**H**[x]. 01_B Reserved. 10_B SR**H**[x] is taken from the ALTINH input line x. 11_B SR**H**[x] is taken from the ALTINH input line x in inverted state.

Page 19-57

n the bit description of bit CSH "SRL" must be changed to "SR**H**".

Page 19-65

The below five formulas must be changed as follows ("MSC0_FDR.STEP"):

$$f_{MSC0} = f_{SYS} \times \frac{1}{n} \text{ with n = 1024 - MSC0\_FDR.STEP} \quad (19.3)$$

$$\text{Baud rate}_{MSC0} = f_{SYS} \times \frac{1}{2 \times (1024 - \text{MSC0\_FDR.STEP})} \quad (19.5)$$

$$\text{Baud rate}_{MSC0} = f_{SYS} \times \frac{\text{MSC0\_FDR.STEP}}{2 \times 1024} \quad (19.6)$$

$$\text{Baud rate}_{MSC0} = f_{SYS} \times \frac{1}{DF \times (1024 - \text{MSC0\_FDR.STEP})} \quad (19.7)$$

$$\text{Baud rate}_{MSC0} = f_{SYS} \times \frac{\text{MSC0\_FDR.STEP}}{DF \times 1024} \quad (19.8)$$

Page 19-68

In the bit description for bit SUSACK in the register description table "Indicates state of SPNDACK signal." must be changed to "Indicates state of **SPNDACK** signal.

Page 19-69

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed to "**r**".

Page 20-39

The bulleted points of Allocation Case 1 must be changed to:

- The upper three bits of **MOIPRn**.MPN (MPN[7:5]) select the number k of a Message Pending Register MSPNDk in which the pending bit will be set.".
- The lower five bits of **MOIPRn**.MPN (MPN[4:0]) select the bit position (0-31) in MSPNDk for the pending bit to be set.

Page 20-50

The second sentence of paragraph 6 must be changed to "Transmit acceptance filtering evaluates TXEN1 for each message object and a message object can win transmit acceptance filtering only if its TXEN**1** bit is set."

Page 20-64

The first sentence of the page must be changed to "Each of the **two** CAN nodes has a list that determines the allocated message objects."

Page 20-66

The first sentence of the page must be changed to "When a message object n generates an interrupt request upon the transmission or reception of a message, then the request is routed to the interrupt output line selected by the bit field MOIPRn.**TXINP** or MOIPRn.**RXINP** of the message object n."

Page 20-68

The bit field description of Message Index Mask must be replaced with "Only those bits in MSPND**k** for which the corresponding Index Mask bits are set contribute to the calculation of the Message Index.

Page 20-113

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed to "**r**".

Page 21-29, 21-34

At both pages, the first bullet paragraphs from the top of the pages should be extended at its end by the following text: "... are not taken into account, **assuming the buffer size is configured correctly (see Page 21-102).**"

Page 21-75

In Figure 21-50, the TCDMR text within the Transmission Status/Control Registers block must be corrected to "**TCMDR**".

Page 21-80

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed into "**r**".

Page 21-102

The description of bit field BS should be extended in the following way:

1. Adding the bit field combination "**1101_B 14-bit offset address of Remote Window**"
2. Adding the following text after bit combination 1111_B: "**Do not use the values 1101_B, 1110_B, and 1111_B as buffer size BS for Small Transfer Windows.**"

Page 21-103

The bit description of AOFF should be extended at its end by the following text: "... are not taken into account for further actions, **assuming the buffer size is configured correctly (see Page 21-102).**"

Page 22-164

The bit description of bit PLLCTR.AEN must be changed as follows:

AEN	2	rw	**Automatic End Mode Enable** With the Automatic End Mode the compensation of input signal's period length variation (acceleration, deceleration) is requested. 0_B Automatic End Mode is disabled. 1_B Automatic End Mode is enabled.

Page 22-228

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed to "**r**".

Page 23-91

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed to "**r**".

Page 24-7

The ACRx.GAIN setting which corresponds to the FAINxP and FAINxN values in **Table 24-1** for Differential Measurement Mode (Configuration 3) must be replaced as below:

Table 24-1 Conversion Results in the Different Measurement Modes

Measurements	ACRx. ENP	ACRx. ENN	FAINxP	FAINxN	ACRx. GAIN	Conversion Results
Single-ended Measurement Mode (Configuration 1)	0	1	"don't care"	0	00_B	768
				3.3		256
				0	01_B	1023
				3.3		0
Single-ended Measurement Mode (Configuration 2)	1	0	0	"don't care"	00_B	256
			3.3			768
			0		01_B	0
			3.3			1023
Differential Measurement Mode (Configuration 3)	1	1	0	1.65	00_B	256
			0	3.3		0
			1.65	0		768
			3.3	0		1023
			3.3	1.65		768
			0	1.65	01_B	0
			1.65	0		1023
			3.3	1.65		1023

Page 24-37

The bit field description CRPRIO must be changed as follows:

CRPRIO	[17:16]	rwh	**Conversion Request Priority** This bit field determines the priority of the conversion requests if more than one channel is requested. If the dynamic priority assignment is enabled, the priority is automatically changed as a function of the gating inputs. The priority of the channels is: 00_B Channel 0 before channel 1 01_B Channel 1 before channel 0 10_B Reserved 11_B Reserved

Page 24-50, 24-51

The bit field description of CRR0.AC and CRR1.AC must be changed as follows:

AC	[26:24]	rh	**Addition Count** With the Automatic End Mode the compensation of input signal's period length variation (acceleration, deceleration) is requested.xxThis bit field indicates the number of additions of filter input values with remain to be executed before the next intermediate result register transfer occurs. AC is loaded with the value of FCRn.ADDL for a new addition sequence, **also when writing GCR.RSTFn = 1**.

Page 24-60

The content "rw" of column "Type" in the last row of the table on top of the page for the "Reserved" bits must be changed to "**r**".

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG