Note: The rising edge detection at the event trigger is enabled with the start of each basic cycle. The first rising edge triggers the compare of the actual cycle time with TTGTP0.CTP. All further edges until the beginning of the next basic cycle are ignored.

40.4 Registers

This section describes the registers within the MCMCAN module. All the registers are prefixed as CAN0_, CAN1_ or CAN2_ for MCMCAN modules, if the corresponding module exist for a particular product variant (Refer Appendix). The registers are grouped as three different sections:

- General Configuration Registers
- User Interface Registers
- Registers within M_CAN

The registers listed in **Figure 604** are not included in the MCMCAN module kernel (part of general module IP blocks), some registers must be programmed for proper operation of the MCMCAN module.

The additional ACCEN registers, are MCMCAN specific.

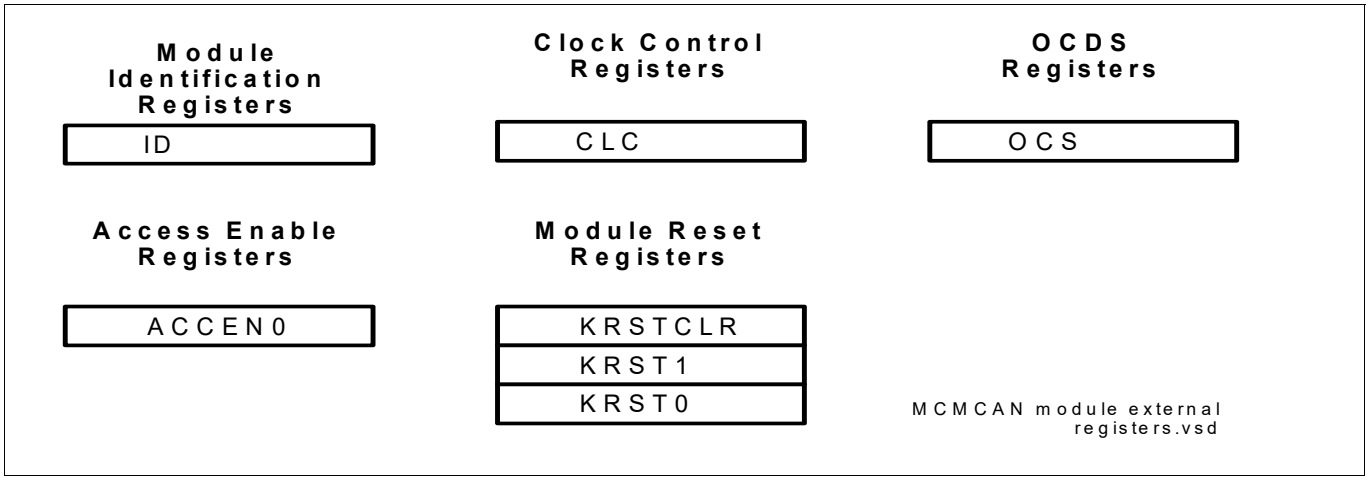


Figure 604 CAN Implementation-specific Special Function Registers

40.4.1 MCMCAN RAM address space

Table 378 Register Address Space - MCMCAN RAM

Module	Base Address	End Address	Note
CANRAM0	F020 0000$_H$	F020 7FFF$_H$	RAM Area CAN0
CANRAM1	F021 0000$_H$	F021 3FFF$_H$	RAM Area CAN1
CANRAM2	F022 0000$_H$	F022 3FFF$_H$	RAM Area CAN2

Note: Refer Appendix of a product variant for the register address space of MCMCAN and available RAM modules.

40.4.2 MCMCAN register overview

Table 379 Register Overview - CAN (ascending Offset Address)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
RAM	Embedded SRAM for messages (008000_H Byte)	000000_H				
CLC	CAN Clock Control Register	008000_H	U,SV	SV,E,P	Application Reset	83
ID	Module Identification Register	008008_H	U,SV	nBE	Application Reset	70
MCR	Module Control Register	008030_H	U,SV	SV,U,P	Application Reset	70
BUFADR	Buffer receive address and transmit address	008034_H	U,SV	SV,U,P	Application Reset	73
MECR	Measure Control Register	008040_H	U,SV	SV,U,P	Application Reset	74
MESTAT	Measure Status Register	008044_H	U,SV	SV,U,P	Application Reset	75
ACCENCTR0	Access Enable Register Control 0	$0080DC_H$	U,SV	SV,SE,P	Application Reset	79
OCS	OCDS Control and Status	$0080E8_H$	U,SV	SV,P,OEN	Debug Reset	76
KRSTCLR	Kernel Reset Status Clear Register	$0080EC_H$	U,SV	SV,E,P	Application Reset	81
KRST1	Kernel Reset Register 1	$0080F0_H$	U,SV	SV,E,P	Application Reset	81
KRST0	Kernel Reset Register 0	$0080F4_H$	U,SV	SV,E,P	Application Reset	80
ACCEN0	Access Enable Register 0	$0080FC_H$	U,SV	SV,SE	Application Reset	78
ACCENNODEi0	Access Enable Register CAN Node i 0	$008100_H + i*400_H$	U,SV	SV,SE,P	Application Reset	79
STARTADRi	Start Address Node i	$008108_H + i*400_H$	U,SV	SV,SE,P	Application Reset	89
ENDADRi	End Address Node i	$00810C_H + i*400_H$	U,SV	SV,SE,P	Application Reset	89
ISREGi	Interrupt Signalling Register i	$008110_H + i*400_H$	U,SV	nBE	Application Reset	86
GRINT1i	Interrupt routing for Groups 1 i	$008114_H + i*400_H$	U,SV	SV,U,P	Application Reset	83
GRINT2i	Interrupt routing for Groups 2 i	$008118_H + i*400_H$	U,SV	SV,U,P	Application Reset	85

Table 379 Register Overview - CAN (ascending Offset Address) (cont'd)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
NTCCRi	Node i Timer Clock Control Register	008120_H +i*400_H	U,SV	SV,U,P	Application Reset	90
NTATTRi	Node i Timer A Transmit Trigger Register	008124_H +i*400_H	U,SV	SV,U,P	Application Reset	91
NTBTTRi	Node i Timer B Transmit Trigger Register	008128_H +i*400_H	U,SV	SV,U,P	Application Reset	92
NTCTTRi	Node i Timer C Transmit Trigger Register	$00812C_H$ +i*400_H	U,SV	SV,U,P	Application Reset	92
NTRTRi	Node i Timer Receive Timeout Register	008130_H +i*400_H	U,SV	SV,U,P	Application Reset	93
NPCRi	Node i Port Control Register	008140_H +i*400_H	U,SV	SV,U,P	Application Reset	87
TTCRi	Time Trigger Control Register	$0081F0_H$	U,SV	SV,U,P	Application Reset	88
CRELi	Core Release Register i	008200_H +i*400_H	U,SV	nBE	See page 95	95
ENDNi	Endian Register i	008204_H +i*400_H	U,SV	nBE	Application Reset	95
DBTPi	Data Bit Timing & Prescaler Register i	$00820C_H$ +i*400_H	U,SV	SV,U,P	Application Reset	96
TESTi	Test Register i	008210_H +i*400_H	U,SV	SV,U,P	Application Reset	97
RWDi	RAM Watchdog i	008214_H +i*400_H	U,SV	SV,U,P	Application Reset	98
CCCRi	CC Control Register i	008218_H +i*400_H	U,SV	SV,U,P	Application Reset	99
NBTPi	Nominal Bit Timing & Prescaler Register i	$00821C_H$ +i*400_H	U,SV	SV,U,P	Application Reset	101
TSCCi	Timestamp Counter Configuration i	008220_H +i*400_H	U,SV	SV,U,P	Application Reset	103
TSCVi	Timestamp Counter Value i	008224_H +i*400_H	U,SV	SV,U,P	Application Reset	104
TOCCi	Timeout Counter Configuration i	008228_H +i*400_H	U,SV	SV,U,P	Application Reset	104
TOCVi	Timeout Counter Value i	$00822C_H$ +i*400_H	U,SV	SV,U,P	Application Reset	105
ECRi	Error Counter Register i	008240_H +i*400_H	U,SV	nBE	Application Reset	106
PSRi	Protocol Status Register i	008244_H +i*400_H	U,SV	nBE	Application Reset	107

Table 379 Register Overview - CAN (ascending Offset Address) (cont'd)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
TDCRi	Transmitter Delay Compensation Register i	008248_H +i*400_H	U,SV	SV,U,P	Application Reset	111
IRi	Interrupt Register i	008250_H +i*400_H	U,SV	SV,U,P	Application Reset	112
IEi	Interrupt Enable i	008254_H +i*400_H	U,SV	SV,U,P	Application Reset	115
GFCi	Global Filter Configuration i	008280_H +i*400_H	U,SV	SV,U,P	Application Reset	117
SIDFCi	Standard ID Filter Configuration i	008284_H +i*400_H	U,SV	SV,U,P	Application Reset	118
XIDFCi	Extended ID Filter Configuration i	008288_H +i*400_H	U,SV	SV,U,P	Application Reset	119
XIDAMi	Extended ID AND Mask i	008290_H +i*400_H	U,SV	SV,U,P	Application Reset	120
HPMSi	High Priority Message Status i	008294_H +i*400_H	U,SV	nBE	Application Reset	120
NDAT1i	New Data 1 i	008298_H +i*400_H	U,SV	SV,U,P	Application Reset	121
NDAT2i	New Data 2 i	$00829C_H$ +i*400_H	U,SV	SV,U,P	Application Reset	122
RXF0Ci	Rx FIFO 0 Configuration i	$0082A0_H$ +i*400_H	U,SV	SV,U,P	Application Reset	122
RXF0Si	Rx FIFO 0 Status i	$0082A4_H$ +i*400_H	U,SV	nBE	Application Reset	123
RXF0Ai	Rx FIFO 0 Acknowledge i	$0082A8_H$ +i*400_H	U,SV	SV,U,P	Application Reset	124
RXBCi	Rx Buffer Configuration i	$0082AC_H$ +i*400_H	U,SV	SV,U,P	Application Reset	125
RXF1Ci	Rx FIFO 1 Configuration i	$0082B0_H$ +i*400_H	U,SV	SV,U,P	Application Reset	125
RXF1Si	Rx FIFO 1 Status i	$0082B4_H$ +i*400_H	U,SV	nBE	Application Reset	126
RXF1Ai	Rx FIFO 1 Acknowledge i	$0082B8_H$ +i*400_H	U,SV	SV,U,P	Application Reset	127
RXESCi	Rx Buffer/FIFO Element Size Configuration i	$0082BC_H$ +i*400_H	U,SV	SV,U,P	Application Reset	127
TXBCi	Tx Buffer Configuration i	$0082C0_H$ +i*400_H	U,SV	SV,U,P	Application Reset	129
TXFQSi	Tx FIFO/Queue Status i	$0082C4_H$ +i*400_H	U,SV	nBE	Application Reset	130

Table 379 Register Overview - CAN (ascending Offset Address) (cont'd)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
TXESCi	Tx Buffer Element Size Configuration i	$0082C8_H$ +i*400_H	U,SV	SV,U,P	Application Reset	131
TXBRPi	Tx Buffer Request Pending i	$0082CC_H$ +i*400_H	U,SV	SV,U,P	Application Reset	132
TXBARi	Tx Buffer Add Request i	$0082D0_H$ +i*400_H	U,SV	SV,U,P	Application Reset	132
TXBCRi	Tx Buffer Cancellation Request i	$0082D4_H$ +i*400_H	U,SV	SV,U,P	Application Reset	133
TXBTOi	Tx Buffer Transmission Occurred i	$0082D8_H$ +i*400_H	U,SV	nBE	Application Reset	134
TXBCFi	Tx Buffer Cancellation Finished i	$0082DC_H$ +i*400_H	U,SV	nBE	Application Reset	134
TXBTIEi	Tx Buffer Transmission Interrupt Enable i	$0082E0_H$ +i*400_H	U,SV	SV,U,P	Application Reset	135
TXBCIEi	Tx Buffer Cancellation Finished Interrupt Enable i	$0082E4_H$ +i*400_H	U,SV	SV,U,P	Application Reset	135
TXEFCi	Tx Event FIFO Configuration i	$0082F0_H$ +i*400_H	U,SV	SV,U,P	Application Reset	136
TXEFSi	Tx Event FIFO Status i	$0082F4_H$ +i*400_H	U,SV	nBE	Application Reset	137
TXEFAi	Tx Event FIFO Acknowledge i	$0082F8_H$ +i*400_H	U,SV	SV,U,P	Application Reset	137
TTTMCi	TT Trigger Memory Configuration i	008300_H	U,SV	SV,U,P	Application Reset	138
TTRMCi	TT Reference Message Configuration i	008304_H	U,SV	SV,U,P	Application Reset	138
TTOCFi	TT Operation Configuration i	008308_H	U,SV	SV,U,P	Application Reset	139
TTMLMi	TT Matrix Limits i	$00830C_H$	U,SV	SV,U,P	Application Reset	141
TURCFi	TUR Configuration i	008310_H	U,SV	SV,U,P	Application Reset	142
TTOCNi	TT Operation Control i	008314_H	U,SV	SV,U,P	Application Reset	144
TTGTPi	TT Global Time Preset i	008318_H	U,SV	SV,U,P	Application Reset	146
TTTMKi	TT Time Mark i	$00831C_H$	U,SV	SV,U,P	Application Reset	147
TTIRi	TT Interrupt Register i	008320_H	U,SV	SV,U,P	Application Reset	148

Table 379 Register Overview - CAN (ascending Offset Address) (cont'd)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
TTIEi	TT Interrupt Enable i	008324_H	U,SV	SV,U,P	Application Reset	151
TTOSTi	TT Operation Status i	$00832C_H$	U,SV	nBE	Application Reset	153
TURNAi	TUR Numerator Actual i	008330_H	U,SV	nBE	Application Reset	155
TTLGTi	TT Local & Global Time i	008334_H	U,SV	nBE	Application Reset	156
TTCTCi	TT Cycle Time & Count i	008338_H	U,SV	nBE	Application Reset	156
TTCPTi	TT Capture Time i	$00833C_H$	U,SV	nBE	Application Reset	157
TTCSMi	TT Cycle Sync Mark i	008340_H	U,SV	nBE	Application Reset	157

40.4.3 General Configuration Registers

This section describes the global module registers, system registers, access enable registers and kernal reset registers.

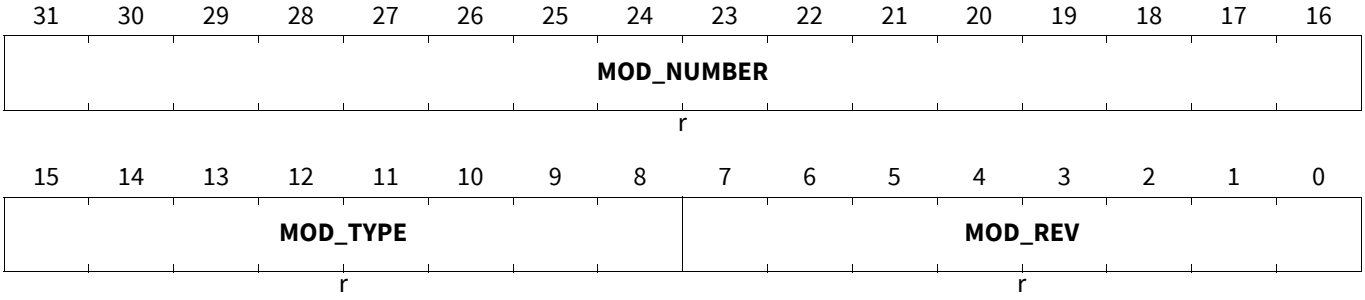
40.4.3.1 Global Module Registers

Global Module Registers, are registers, which are not part of M_CAN nodes.

Module Identification Register

The Module Identification Register ID contains read-only information about the module version.

ID
Module Identification Register **(008008_H)** **Application Reset Value: $00B8\ C0XX_H$**



Field	Bits	Type	Description
MOD_REV	7:0	r	**Module Revision Number** MOD_REV defines the revision number. The value of a module revision starts with 01_H (first revision).
MOD_TYPE	15:8	r	**Module Type** $C0_H$ Define the module as a 32-bit module.
MOD_NUMBER	31:16	r	**Module Number Value** This bit field defines the MCMCAN module identification number $00B8_H$.

Module Control Register

The Module Control Register MCR contains basic settings that determine the operation of the MCMCAN module.

The write access to the lowest byte of the MCR register becomes only valid, if and only if, MCR.CCCE and MCR.CI are already set during write access. To switch the clocks on or off, the bits of MCR.CCCE and MCR.CI have to be reset afterwards. Before this sequence hasn't taken place, no write access to the corresponding nodes, can be done.

Note: If the baud rate logic is supplied from an unstable clock source, or no clock at all, the CAN functionality is not guaranteed.

To be able to change the clock settings the following programming sequence needs to be met:

```
uwTemp = CANn_MCR.U;
uwTemp |= (0xC0000000 | CLKSELx);
CANn_MCR.U = uwTemp;
uwTemp &= ~0xC0000000;
```

CANn_MCR.U = uwTemp;
The clock settings for CAN nodes becomes active.
To be able to start the RAM initialization, the following programming sequence need to be met:
CANn_MCR |= 0xC0000000;
Wait until CANn_MCR.RBUSY is 0b
Set CANn_MCR.RINIT to 0b
Set CANn_MCR.RINIT to 1b
Dummy read CANn_MCR
Wait until CANn_MCR.RBUSY is 0b
Set CANn_MCR.RINIT to 0b
CANn_MCR &= ~0xC0000000;
RAM initialization is finished

MCR
Module Control Register **(008030_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCCE	**CI**	**RINIT**	**RBUSY**	**DXCM**	**NODE**			**0**							
rw	rw	rw	rh	rw		rw		r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								**CLKSEL3**		**CLKSEL2**		**CLKSEL1**		**CLKSEL0**	
r								rw		rw		rw		rw	

Field	Bits	Type	Description
CLKSEL0	1:0	rw	**Clock Select 0** This bitfield is MCR.CI and MCR.CCCE protected. 00_B No clock supplied 01_B The asynchronous clock source is switched on 10_B The synchronous clock source is switched on 11_B Both clock sources are switched on
CLKSEL1	3:2	rw	**Clock Select 1** This bitfield is MCR.CI and MCR.CCCE protected. 00_B No clock supplied 01_B The asynchronous clock source is switched on 10_B The synchronous clock source is switched on 11_B Both clock sources are switched on
CLKSEL2	5:4	rw	**Clock Select 2** This bitfield is MCR.CI and MCR.CCCE protected. 00_B No clock supplied 01_B The asynchronous clock source is switched on 10_B The synchronous clock source is switched on 11_B Both clock sources are switched on

Field	Bits	Type	Description
CLKSEL3	7:6	rw	**Clock Select 3** This bitfield is MCR.CI and MCR.CCCE protected. 00_B No clock supplied 01_B The asynchronous clock source is switched on 10_B The synchronous clock source is switched on 11_B Both clock sources are switched on
NODE	26:24	rw	**Node** This bit field determines the CAN node i which is used for debug over CAN. This bitfield only exists on CAN0. 000_B Node 0 … 011_B Node 3
DXCM	27	rw	**Debug Over CAN Messages Enable** This bit enables the debug over serial connections between DAP and CAN0 module. If enabled the lowest receive/transmit message buffer is reserved for debugger communication. DXCM is described in detail in the OCDS chapter. This bit only exists on CAN0. 0_B DXCM disabled 1_B DXCM enabled
RBUSY	28	rh	**RAM BUSY** This bit shows that the RAM Initialization is running. This bit is set back to 0b by hardware when the RAM intialization is completed.
RINIT	29	rw	**RAM Init** This bit is MCR.CI and MCR.CCCE protected. This bit starts the initialization of the RAM block to all 0x0. The RAM initialization is started only when this bit is changed from 0b to 1b and also RBUSY is 0b.
CI	30	rw	**Change Init** Needs to be set to enable and disable clocks. 0_B Change Init disabled 1_B Change Init enabled (takes effect with CCCE:=1)
CCCE	31	rw	**Clock and RAM Change Enable** Needs to be set to enable and disable the clocks. 0_B Clock and RAM Change disabled 1_B Clock and RAM Change enabled (takes effect with CI:=1)
0	23:8	r	**Reserved** Shall read 0; shall be written with 0.

Debug over CAN (DXCM feature)

The MCMCAN controller supports debugging using standard CAN tool access in parallel to regular CAN bus traffic. This is achieved by transmitting DAP telegrams and replies as regular CAN messages (DXCM DAP over CAN Messages). DXCM uses the lowest message buffers and it is strongly recommended to use also the same CAN pins as for DXCPL (DAP over CAN Physical Layer). DXCM is enabled with the MCR.DXCM bit. Please refer to the OCDS chapter for more information about DAP, DXCM and DXCPL.

Debug over CAN shall be only available on CAN0. TX Buffer 0 will be the sending transmit object. For receive at least one message buffer has to be configured. Meaning that RX Buffer 0 will be used for receiving DAP telegrams. The starting address of the message buffer and the receiving address of the message buffer have to be configured within BUFADR register.

Assigning the buffer start address

The following register assigns the start address to all features needing the message buffers inside the corresponding M_CAN, which are for receive and transmit.

Buffer receive address and transmit address

BUFADR

Buffer receive address and transmit address (008034$_H$) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		RXBUF													
r		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		TXBUF													
r		rw													

Field	Bits	Type	Description
TXBUF	13:0	rw	**Transmit Buffer start address** This is the start address of the first dedicated transmit buffer.
RXBUF	29:16	rw	**Receive Buffer start address** This is the start address of the first dedicated receive buffer.
0	15:14, 31:30	r	**Reserved** Shall read 0; shall be written with 0.

Pretended Networking

The registers above are intended to support Pretended Networking. As an application example, the SPB bus is clocked at 40MHz. The asynchronous module part is clocked either with 40MHz as well or even more power saving with direct drive from the oscillator. The cores are in idle mode.

Messages can be received and a receive interrupt can be generated. It is possible to trigger messages with or without changing the content by the timers provided. For example the network management message and two related messages can be triggered without any CPU interaction.

As mostly the operating system is still running, messages can be changed without any problem during the time, where the operating system is active.

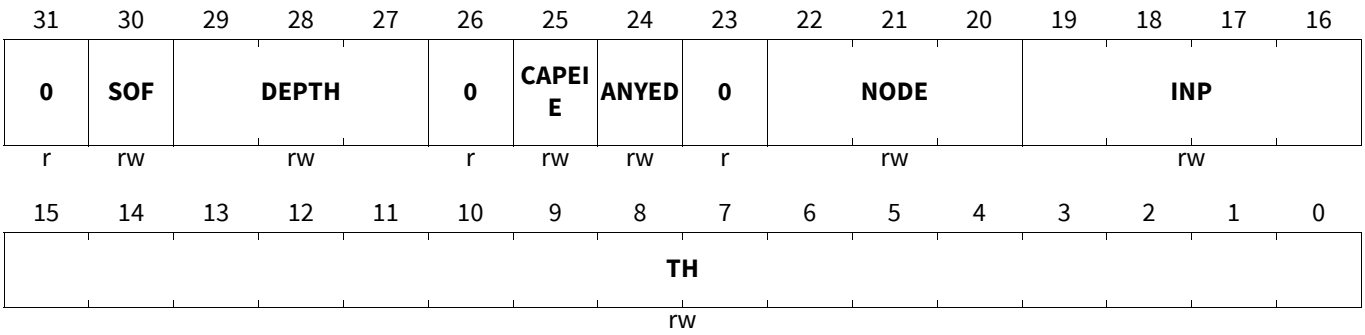
Oscillator calibration

The following registers support the oscillator calibration on which the decision is taken to increase or decrease the frequency. A detailed description will be provided as Application Node.

Measure Control Register

The Measure Control Register MECR controls the CAN edge timing measurement function for calibration purposes. This feature only exists on CAN0.

MECR
Measure Control Register **(008040_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
TH	15:0	rw	**Threshold** This bit field contains the threshold value for the measurement timer. If TH = 0000_H, the timer is stopped and the capture function is disabled.
INP	19:16	rw	**Interrupt Node Pointer** INP selects the interrupt output line INT_Om (m = 0-15) for a capture event interrupt. 0_H Interrupt output line INT_O0 is selected … F_H Interrupt output line INT_O15 is selected
NODE	22:20	rw	**Node** This bit field determines the CAN node i whose input line RXDCANi is used for start and capture of the measurement timer. 000_B Node 0 … 011_B Node 3
ANYED	24	rw	**Any Edge** This bit enables capture on any edge of CAN input line specified by NODE. 0_B Capture on falling (dominant) edge only 1_B Capture on rising (recessive) or falling (dominant) edge
CAPEIE	25	rw	**Capture Event Interrupt Enable** This bit enables the capture event interrupt. Bit field INP selects the interrupt output line which becomes activated at this type of interrupt. 0_B Capture event interrupt is disabled 1_B Capture event interrupt is enabled

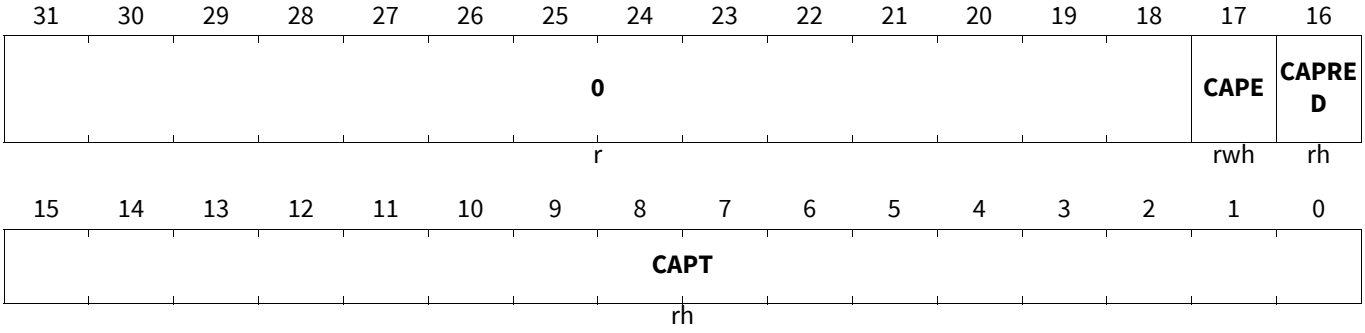
Field	Bits	Type	Description
DEPTH	29:27	rw	**Digital Glitch Filter Depth** DEPTH determines the number of input samples clocked with f_{SYNi} that are taken into account for the calculation of the floating average. The higher DEPTH is chosen to be, the longer the glitches that are suppressed and the longer the delay of the input signal introduced by this filter. 000_B off, default 001_B Filter depth of 8 cycles 010_B Filter depth of 16 cycles 011_B Filter depth of 32 cycles 100_B Filter depth of 64 cycles 101_B Filter depth of 128 cycles 110_B Filter depth of 255 cycles 111_B not allowed, reserved
SOF	30	rw	**Start Of Frame** This bit selects falling edge or any edge as measurement for start of frame detection. 0_B Measurement starts with any falling edge 1_B Measurement starts with falling Start of Frame edge. i.e any falling edge that occurs while the CAN node is in idle state
0	23, 26, 31	r	**Reserved** Shall read 0; shall be written with 0.

Measure Status Register

The Measure Status Register MESTAT contains the status information of the CAN edge timing measurement. This feature only exists on CAN0.

MESTAT

Measure Status Register **(008044_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
CAPT	15:0	rh	**Captured Timer** This bit field contains the captured measurement timer content. The timer itself is cleared and started by the first falling (dominant) edge of a CAN frame on the input line of the CAN node specified by MECR.NODE. The timer is incremented by the module control clock f_{SYNi} and will be stopped when $FFFF_H$ is reached. If MECR.TH = 0000_H, the timer is always stopped. A capture will take place if all the following conditions are met: 1. MECR.TH > 0000_H 2. Timer is cleared and started by new frame 3. Timer reaches MECR.TH 4. This node is not sending and first edge (as specified by MECR.ANYED) after 3. occurs on input line Capture will be repeated for the following CAN frames until MECR.TH is cleared.
CAPRED	16	rh	**Captured Rising Edge** This bit indicates the type of edge that caused the last capture event. 0_B Capture occurred on falling (dominant) edge 1_B Capture occurred on rising (recessive) edge
CAPE	17	rwh	**Capture Event** This flag is set on a capture event. It must be reset by software. An interrupt request is generated if MECR.CAPEIE = 1. If CAPE=1 then no further measurement results are posted to MESTAT.CAPT and MESTAT.CAPRED. CAPE bit has to be cleared to re-enable update of MESTAT.CAPT and MESTAT.CAPRED. 0_B No capture event has occurred since last flag reset 1_B Capture event has occurred since last flag reset
0	31:18	r	**Reserved** Shall read 0; shall be written with 0.

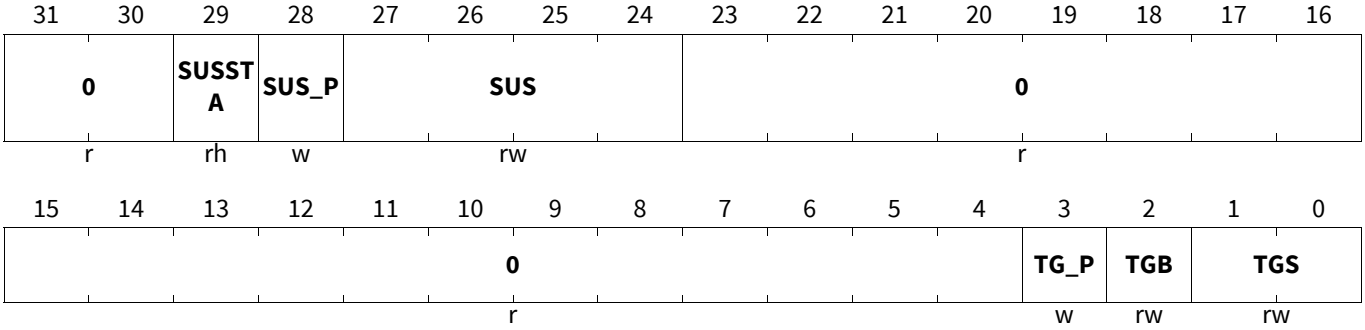
40.4.3.2 System Registers

OCDS Control and Status

OCDS Trigger Bus (OTGB) The OCDS Control and Status (OCS) register is cleared by Debug Reset. The OCS register can only be written when the OCDS is enabled. If OCDS is being disabled, the OCS register value will not change. When OCDS is disabled the OCS suspend control is ineffective. Write access requires Supervisor Mode.

OCS

OCDS Control and Status **(0080E8$_H$)** **Debug Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
TGS	1:0	rw	**Trigger Set for OTGB0/1** 00_B No Trigger Set output 01_B TS16_CAN **others**, reserved
TGB	2	rw	**OTGB0/1 Bus Select** 0_B Trigger Set is output on OTGB0 1_B Trigger Set is output on OTGB1
TG_P	3	w	**TGS, TGB Write Protection** TGS and TGB are only written when TG_P is 1, otherwise unchanged. Read as 0.
SUS	27:24	rw	**OCDS Suspend Control** Controls the sensitivity to the suspend signal coming from the OCDS Trigger Switch (OTGS) 0_H Will not suspend 1_H Hard suspend. Clock is off immediately. Do not use this mode in normal CAN applications, this mode is meant for debugging the peripheral IP. 2_H Soft suspend of CAN nodes. **others**, reserved
SUS_P	28	w	**SUS Write Protection** SUS is only written when SUS_P is 1, otherwise unchanged. Read as 0.
SUSSTA	29	rh	**Suspend State** 0_B CAN nodes are not (yet) suspended 1_B All CAN nodes are suspended
0	23:4, 31:30	r	**Reserved** Shall read 0; shall be written with 0.

OCDS Trigger Bus (OTGB) Interface

The MCMCAN Trigger Set is shown in **Table 380**. Its output is on OTGB0 or OTGB1 controlled by the **OCDS Control and Status** register. Links are only for CAN0, but the feature is available for all CAN modules.

Table 380 TS16_CAN Trigger Set MCMCAN

Values	Name	Description
i	AF	Acceptance filtering done for node i
i + 4	MR	Message successfully received on node i
i + 8	FDR	Fast Data Phase reception on node i
i + 12	FDT	Fast Data Phase transmission on node i.

40.4.3.3 Access Enable Registers ACCEN

The access enable bits control the access on the module itself.

Access Enable Register 0

The Access Enable Register 0 controls write access[1] for transactions with the on chip bus master TAG ID 000000_B to 011111_B (see On Chip Bus chapter for the products TAG ID <-> master peripheral mapping). The BPI_FPI is prepared for a 6-bit TAG ID. The registers ACCEN0 provides one enable bit for each possible 6-bit TAG ID encoding.

Mapping of TAG IDs to ACCEN0.ENy: EN0 -> TAG ID 000000_B, EN1 -> TAG ID 000001_B, … ,EN31 -> TAG ID 011111_B.

ACCEN0

Access Enable Register 0 ($0080FC_H$) Application Reset Value: $FFFF\ FFFF_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ENy (y=0-31)	y	rw	**Access Enable for Master TAG ID y** This bit enables write access to the module kernel addresses for transactions with the Master TAG ID y. 0_B Write access will not be executed 1_B Write access will be executed

40.4.3.4 Additional Access Enable Registers ACCENCTR and ACCENNODEi

The access enable bits help the application to control the access rights via bus master TAG ID. Inside the CAN node registers, an additional address range can be defined, called STARTADR and ENDADR.

The ACCENNODEi registers are protecting node i and the memory range defined in the STARTADRi and ENDADRi register. To disable the mechanism the STARTADRi has to be higher than the corresponding ENDADRi of the node.

1) The BPI_FPI Access Enable functionality controls only write transactions. Read transactions are not influenced. SW has to take care for destructive/modifying read functionality in kernel registers-

Access Enable Register Control 0

The Access Enable Register Control 0 controls write access for transactions with the on chip bus master TAG ID 000000_B to 011111_B (see On Chip Bus chapter for the products TAG ID ↔ master peripheral mapping). The BPI_FPI is prepared for a 6-bit TAG ID. The registers ACCENCTR0 provides one enable bit for each possible 6-bit TAG ID encoding. The control registers (address range 8020_H to $804F_H$) are protected by this register.

Mapping of TAG IDs to ACCENCTR0.ENy: EN0 → TAG ID 000000_B, EN1 → TAG ID 000001_B, … ,EN31 → TAG ID 011111_B.

ACCENCTR0
Access Enable Register Control 0 ($0080DC_H$) Application Reset Value: FFFF FFFF$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN31	**EN30**	**EN29**	**EN28**	**EN27**	**EN26**	**EN25**	**EN24**	**EN23**	**EN22**	**EN21**	**EN20**	**EN19**	**EN18**	**EN17**	**EN16**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	**EN14**	**EN13**	**EN12**	**EN11**	**EN10**	**EN9**	**EN8**	**EN7**	**EN6**	**EN5**	**EN4**	**EN3**	**EN2**	**EN1**	**EN0**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ENy (y=0-31)	y	rw	**Access Enable for Master TAG ID y** This bit enables write access to the module kernel addresses for transactions with the Master TAG ID y 0_B Write access will not be executed 1_B Write access will be executed

Access Enable Register CAN Node i 0

The Access Enable Register CAN Node i Control 0 controls write access for transactions with the on chip bus master TAG ID 000000_B to 011111_B (see On Chip Bus chapter for the products TAG ID ↔ master peripheral mapping). The BPI_FPI is prepared for a 6-bit TAG ID. The registers ACCENNODEi0 provides one enable bit for each possible 6-bit TAG ID encoding. This register controls node y registers, including the STARTADR0 and ENDADR0 register included in the node control area (8100_H+i*0400_H to 8400_H+i*0400_H-1). This means, that this area is including these registers and also the STARTADRi and ENDADRi. If the programming of the CPU leads to the fact, that it is disabling its own access, it will not be able to correct such a setting afterwards.

Mapping of TAG IDs to ACCENNODEi0.ENy: EN0 → TAG ID 000000_B, EN1 → TAG ID 000001_B, … , EN31 → TAG ID 011111_B.

ACCENNODEi0 (i=0-3)
Access Enable Register CAN Node i 0 (008100_H+i*400_H) Application Reset Value: FFFF FFFF$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN31	**EN30**	**EN29**	**EN28**	**EN27**	**EN26**	**EN25**	**EN24**	**EN23**	**EN22**	**EN21**	**EN20**	**EN19**	**EN18**	**EN17**	**EN16**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	**EN14**	**EN13**	**EN12**	**EN11**	**EN10**	**EN9**	**EN8**	**EN7**	**EN6**	**EN5**	**EN4**	**EN3**	**EN2**	**EN1**	**EN0**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ENy (y=0-31)	y	rw	**Access Enable for Master TAG ID y** This bit enables write access to the module kernel addresses for transactions with the Master TAG ID y 0_B Write access will not be executed 1_B Write access will be executed

40.4.3.5 Kernel Reset Registers

The Kernel Reset Registers give the user the possibility to reset the module without resetting the device.

Kernel Reset Register 0

The Kernel Reset Register 0 is used to reset the related module kernel. Kernel registers related to the Debug Reset (Class 1) are not influenced. To reset a module kernel it is necessary to set the RST bits by writing with ´1´ in both Kernel Reset Registers related to the module kernel reset.The RST bit will be reset by the BPI with the end of the BPI kernel reset sequence.

Kernel Reset Register 0 includes a kernel reset status bit that is set to ´1´ by the BPI_FPI in the same clock cycle the RST bit is reset by the BPI_FPI. This bit can be used to detect that a kernel reset was processed. The bit can be reset to ´0´ by writing to KRSTCLR.CLR with ´1´.

During the execution of the kernel reset until RSTSTAT is set, access to the kernel registers will result in an error acknowledge.

KRST0
Kernel Reset Register 0 **($0080F4_H$)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														RSTSTAT	RST
r														rh	rwh

Field	Bits	Type	Description
RST	0	rwh	**Kernel Reset** This reset bit can be used to request for a kernel reset. The kernel reset will be executed if the reset bits of both kernel registers are set. The RST bit will be cleared (reset to ´0´) by the BPI_FPI after the kernel reset was executed. 0_B No kernel reset was requested 1_B A kernel reset was requested

Field	Bits	Type	Description
RSTSTAT	1	rh	**Kernel Reset Status** This bit indicates wether a kernel reset was executed or not. This bit is set by the BPI_FPI after the execution of a kernel reset in the same clock cycle both reset bits. This bit can be cleared by writing with ´1´ to the CLR bit in the related KRSTCLR register. 0_B No kernel reset was executed 1_B Kernel reset was executed
0	31:2	r	**Reserved** Shall read 0; shall be written with 0.

Kernel Reset Register 1

The Kernel Reset Register 1 is used to reset the related module kernel. To reset a module kernel it is necessary to set the RST bits by writing with ´1´ in both Kernel Reset Registers (KRST1.RST and KRST0.RST) related to the module kernel reset. The RST bit will be reset (cleared to ‘0’) by the BPI with the end of the BPI kernel reset sequence.

KRST1

Kernel Reset Register 1 **($0080F0_H$)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															RST
r															rwh

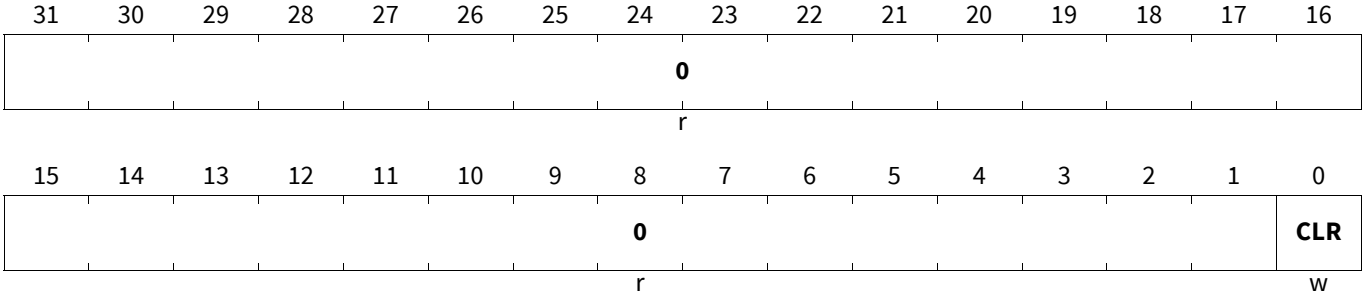
Field	Bits	Type	Description
RST	0	rwh	**Kernel Reset** This reset bit can be used to request for a kernel reset. The kernel reset will be executed if the reset bits of both kernel reset registers is set. The RST bit will be cleared (reset to ´0´) by the BPI_FPI after the kernel reset was executed. 0_B No kernel reset was requested 1_B A kernel reset was requested
0	31:1	r	**Reserved** Shall read 0; shall be written with 0.

Kernel Reset Status Clear Register

The Kernel Reset Clear Register is used to clear the Kernel Reset Status bit (KRST0.RSTSTAT).

KRSTCLR

Kernel Reset Status Clear Register **(0080EC$_H$)** **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
CLR	0	w	**Kernel Reset Status Clear** Read always as 0. 0_B No action 1_B Clear Kernel Reset Status KRST0.RSTSTAT
0	31:1	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4 MCMCAN User Interface Registers

This section describes the registers for clock control, port connections, interrupt control, and address decoding.

40.4.4.1 The Clock Control Register

CAN Clock Control Register

The Clock Control Register CLC allows the programmer to adapt the functionality and power consumption of the module to the requirements of the application. The description below shows the clock control register functionality which is implemented in the standard interface for the module. Where a module kernel is connected to the CLC clock control interface, CLC controls the f_{SYN} and f_{ASYN} module clock signal, sleep mode and fast shut-off mode for the module.

CLC

CAN Clock Control Register **(008000_H)** **Application Reset Value: $0000\ 0003_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												EDIS	0	DISS	DISR
r												rw	r	rh	rw

Field	Bits	Type	Description
DISR	0	rw	**Module Disable Request Bit** Used for enable/disable control of the module. The synchronous and asynchronous clock is switched on/off Note that no register access is possible to any register while module is disabled. A disable request is granted, if the M_CAN clock is disabled, or all M_CAN nodes acknowledge the disable request.
DISS	1	rh	**Module Disable Status Bit** Bit indicates the current status of the module.
EDIS	3	rw	**Sleep Mode Disable Control** Used to control module's sleep mode.
0	2, 31:4	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.2 Interrupt Grouping and Signalling Registers

Interrupt routing for Groups 1 i

GRINT1i is the first of two grouping registers. In this register, the interrupt line within the module is fixed. Please be reminded, that the interrupt sources need to be enabled to be mapped. The total module has 16 interrupts and the interrupt node can be chosen within GRINT1i and GRINT2i.

Meaning:

0000_B Interrupt output line INT_O0 is selected.

0001_B Interrupt output line INT_O1 is selected.

$\cdots_B \cdots$

1110_B Interrupt output line INT_O14 is selected.

1111_B Interrupt output line INT_O15 is selected.

GRINT1i (i=0-3)

Interrupt routing for Groups 1 i **(008114_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOI				BOFF				SAFE				MOER			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALRT				WATI				HPE				TEFIFO			
rw				rw				rw				rw			

Field	Bits	Type	Description
TEFIFO	3:0	rw	**Transmit Event FIFO Incidents** are mapped here. IR.TEFF (Transmit Event FIFO Full) and IR.TEFN (Transmit Event FIFO New Entry)
HPE	7:4	rw	**High Priority Events** are mapped here, giving IR.HPM an interrupt level
WATI	11:8	rw	**Watermark interrupts** are mapped here: IR.TEFW (Transmit FIFO warning interrupt reached), IR.RF1W (Receive FIFO 1 warning interrupt reached). IR.RF0W (Receive FIFO 0 warning interrupt reached)
ALRT	15:12	rw	**ALERTS** All kind of alerts are mapped here. IR.EW (warning status), IR.EP (error passive), IR.TSW (timestamp wrap around), IR.TEFL (Transmit Event FIFO Element Lost), IR.RF0L (Receive FIFO 0 Message Lost), IR.RF1L (Receive FIFO 1 Message Lost). The following TTCAN error messages and warnings are also shown here: TTIR. CER (Configuration Error), TTIR.AW Application Watchdog, TTIR.WT (Watch Trigger), TTIR.IWT Initialization Watch Trigger, TTIR.ELC (Error Level Changed), TTIR.SE2 (Scheduling Error 2), TTIR.SE1 (Scheduling Error), TTIR.TXO (Tx Count Overflow), TTIR.TXU (TX Count Underflow), TTIR.GTE (Global Time Error), TTIR.GTD (Global Time Discontinuity) and TTIR.GTW (Global Time Wrap)
MOER	19:16	rw	**Module errors** IR.WDI (watchdog interrupt) and IR.MRAF (message RAM access failure) are mapped here.
SAFE	23:20	rw	**Safety counter overflow** The interrupt node for IR.ELO showing a safety counter overflow
BOFF	27:24	rw	**Bus Off has been reached** Mapped to IRi.BO flag indication the change in Bus_Off status. To get out of bus off, the CCCRn.INIT bit has to be reset.

Field	Bits	Type	Description
LOI	31:28	rw	**Last Error Interrupts** The interrupt sources IR.PED (Protocol Error in Data Phase) and IR.PEA (Protocol Error in Arbitration Phase) are signalled here.

Interrupt routing for Groups 2 i

GRINT2i has the same functionality as GRINT1i, but for other interrupt sources. The interrupt sources need to be enabled to be mapped.

GRINT2i (i=0-3)
Interrupt routing for Groups 2 i **(008118_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRACO				**TRAQ**				**RETI**				**RxF0N**			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxF1N				**RxF0F**				**RxF1F**				**REINT**			
rw				rw				rw				rw			

Field	Bits	Type	Description
REINT	3:0	rw	**Message stored in dedicated receive buffer interrupt (IR.DRX)** is assigned to interrupt node.
RxF1F	7:4	rw	**IR.RF1F** Receive FIFO1 full interrupt assigned to an interrupt node
RxF0F	11:8	rw	**IR.RF0F** Receive FIFO0 full interrupt assigned to an interrupt node
RxF1N	15:12	rw	**IR.RF1N** Receive FIFO1 new message assigned to an interrupt node
RxF0N	19:16	rw	**IR.RF0N** Receive FIFO0 new message assigned to an interrupt node
RETI	23:20	rw	**Receive Timeouts** can be assigned here. IR.TOO (time-out event) and TE (Timer Event)
TRAQ	27:24	rw	**Transmission Queue Events** can be assigned here. IR.TFE Transmission FIFO Empty
TRACO	31:28	rw	**Interrupts of the transmission control** can be assigned here. IR.TCF (Transmission Cancellation Finished) and IR.TF (Transmission Completed). As an additional information the copy of a local time event is shown here with TTIR.SWT (Stop Watch Event). Further on the TTIR.TTMI Trigger Time Event Internal, TTIR.RTMI (Register Time Mark), TTIR.SOG (Start of Gap), TTIR.CSM (Change of Synchronization Mode), TTIR.SMC (Start Matrix Cycle) and TTIR.SBC (Start of Basic Cycle) are shown here.

Interrupt Signalling Register i

The groups by the GRINT registers are also shown inside the ISREG (interrupt signalling register) register. Inside the interrupt signalling register a 1 means, that one of the corresponding bits inside the interrupt (status) register of the corresponding M_CAN node, at least one group member is showing an interrupt. ISREG is purely ORing the interrupt status bits of the group to enable SW to have proper handling of the bits. Writing to ISREGi has no effect.

If ISREGi is written, this shall have no effect on the interrupt status inside the M_CAN nodes. The bits have to be reset inside the corresponding M_CAN nodes, see register CANn_IRi.

ISREGi (i=0-3)

Interrupt Signalling Register i **(008110_H+i*400_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOI	BOFF	SAFE	MOER	ALRT	WATI	HPE	TEFIFO	TRACO	TRAQ	RETI	RxF0N	RxF1N	RxF0F	RxF1F	REINT
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
REINT	0	rh	**A message stored in a receive buffer interrupt**
RxF1F	1	rh	**Receive FIFO1 is full interrupt**
RxF0F	2	rh	**Receive FIFO0 is full interrupt**
RxF1N	3	rh	**Receive FIFO1 got a new message interrupt**
RxF0N	4	rh	**Receive FIFO0 got a new message interrupt**
RETI	5	rh	**A receive timeout event interrupt**
TRAQ	6	rh	**A transmission queue event interrupt**
TRACO	7	rh	**A transmission control event interrupt**
TEFIFO	8	rh	**A Transmit Event FIFO Incident interrupt**
HPE	9	rh	**A high priority event interrupt**
WATI	10	rh	**A watermark interrupt has been reached**
ALRT	11	rh	**An alert interrupt**
MOER	12	rh	**Module error interrupt**
SAFE	13	rh	**The safety counter interrupt ELO**
BOFF	14	rh	**Bus Off Interrupt**
LOI	15	rh	**Last Error Interrupt**
0	31:16	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.3 Node Port Control Register

Node i Port Control Register

The Node Port Control Register NPCRi configures the CAN bus transmit/receive ports.

NPCRi (i=0-3)

Node i Port Control Register **(008140_H+i*400_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

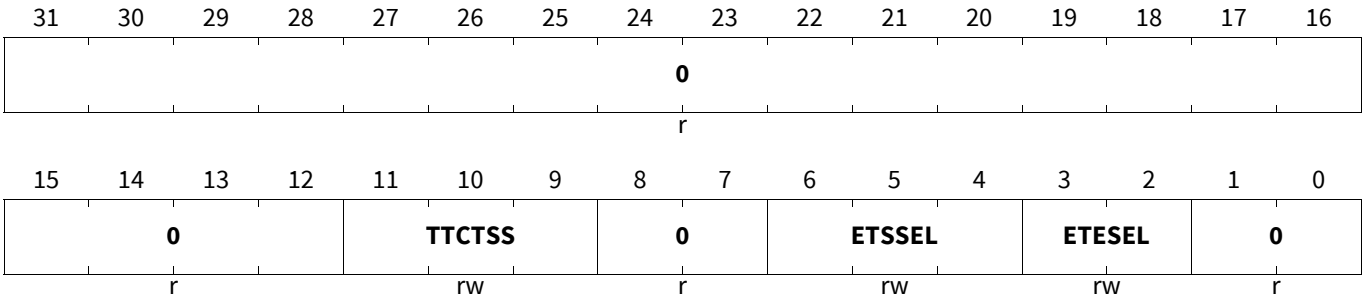
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					DELE	LOUT	LBM	0					RXSEL		
r					rw	rw	rw	r					rw		

Field	Bits	Type	Description
RXSEL	2:0	rw	**Receive Select** RXSEL selects one out of 8 possible receive inputs. The CAN receive signal is performed by the selected input. (see the device related chapter for RXSEL)
LBM	8	rw	**Loop-Back Mode** 0_B Loop-Back Mode is disabled. 1_B Loop-Back Mode is enabled. This node is connected to an internal (virtual) loop-back CAN bus. All CAN nodes which are in Loop-Back Mode are connected to this virtual CAN bus so that they can communicate with each other internally. The external transmit line is forced recessive in Loop-Back Mode.
LOUT	9	rw	**Loop Back Mode Out** The loop back bus is switched to the external CAN bus of the node.
DELE	10	rw	**Enable destructive read on ECRi.CEL** If this bit is set, the destructive read on ECRi.CEL and on the PSR register takes place. Meaning, that with read access on ECRi, the CEL is reset. The same is true for the PSR register, for the bits PXE, RFDF, RBRS, RESI, LEC and DLEC. After the destructive read it is advised to reset the bit again.
0	7:3, 31:11	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.4 Time Trigger Control Register

Time Trigger Control Register

TTCR0
Time Trigger Control Register **(0081F0$_H$)** **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
ETESEL	3:2	rw	**External Trigger Event Selection** This bit field defines the external trigger event that can be used to trigger the transmission of the reference message. The event causes the Event Trigger to be triggered. Control settings for this will not be influenced. 00_B The external event ECTTx does not trigger the transmission of the reference message. 01_B The reference message will be transmitted when a negative edge is detected at the selected input line ECTTx. 10_B The reference message will be transmitted when a positive edge is detected at the input line ECTTx. 11_B The reference message will be transmitted when a negative edge or a positive edge is detected at the input line ECTTx.
ETSSEL	6:4	rw	**External Trigger Source Selection** This bit fields selects the input source for the external reference message trigger. 000_B External trigger input line ECTT1 selected 001_B External trigger input line ECTT2 selected 010_B External trigger input line ECTT3 selected 011_B External trigger input line ECTT4 selected 100_B External trigger input line ECTT5 selected 101_B External trigger input line ECTT6 selected 110_B External trigger input line ECTT7 selected 111_B External trigger input line ECTT8 selected
TTCTSS	11:9	rw	**TTCapture Time Trigger Source Select** This bit selects the input source for the TT Capture Time (TTCPT) trigger. This register influences the stop watch event trigger 000_B No TTCPT trigger input allowed 001_B Local time register capture trigger input TTCPT_TRIG1 selected … 100_B Local time register capture trigger input TTCPT_TRIG4 selected **others**, Reserved; do not use this combination

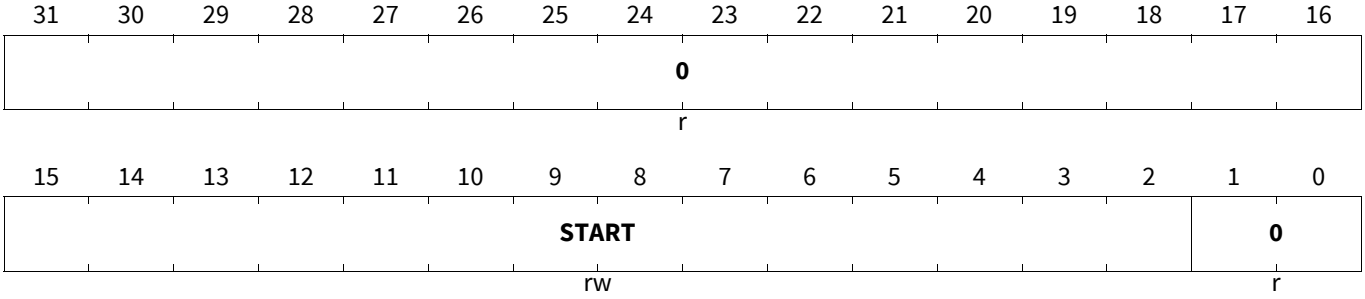
Field	Bits	Type	Description
0	1:0, 8:7, 31:12	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.5 Message RAM start address register

Start Address Node i

In case the RAM shall not be protected, the STARTADR has to be higher than the corresponding ENDADR of the node.

STARTADRi (i=0-3)
Start Address Node i **(008108_H+i*400_H)** **Application Reset Value: 0000 0000_H**

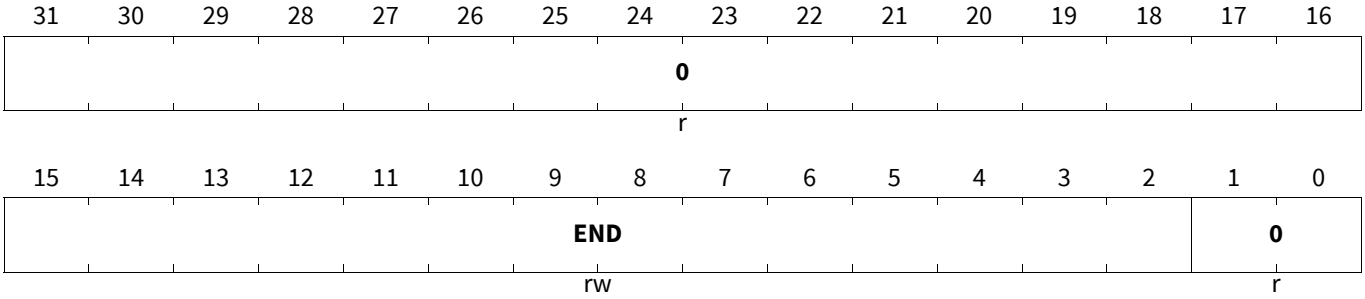


Field	Bits	Type	Description
START	15:2	rw	**Message RAM start** The address within the RAM area of the MCMCAN, of node i, where the message RAM to be protected starts
0	1:0, 31:16	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.6 Message RAM end address register

End Address Node i

ENDADRi (i=0-3)
End Address Node i **($00810C_H$+i*400_H)** **Application Reset Value: 0000 0000_H**



Field	Bits	Type	Description
END	15:2	rw	**Message RAM end** The address within the RAM area of the MCMCAN, of node i, where the message RAM to be protected ends
0	1:0, 31:16	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.7 NTCCR

The Node i Timer Clock Control and Node i Timer A/B/C Transmit Trigger Registers offer additional timing functions for the node.

Node i Timer Clock Control Register

The Node i Timer Clock Control Register NTCCRi controls the functions of the node timer.

NTCCRi (i=0-3)
Node i Timer Clock Control Register **(008120_H+i*400_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0											TRIGSRC			0	
r											rw			r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STSTART	STRESET	0		TPSC				0							
rw	rw	r		rw				r							

Field	Bits	Type	Description
TPSC	11:8	rw	**Timer Prescaler** The duration of one timer clock is given by (TPSC + 1) CAN bit times for all NTCCRi.TRIGSRC settings.
STRESET	14	rw	**Stamping Reset** This bit gives the possibility to reset the time stamp for CAN FD messages.
STSTART	15	rw	**Stamping Start** This bit starts the external timer used for CAN FD messages. The source and the prescaler are identical to the timers A/B/C.

Field	Bits	Type	Description
TRIGSRC	20:18	rw	**Trigger Source** This bit selects the trigger source for the different modes in the node timer. 000_B Node i Timer is decremented per f_{SYNi} prescaled by (TPSC + 1) timing to 0. 001_B System Timer (STM) trigger event enabled Node i Timer is decremented per STM trigger event prescaled by (TPSC + 1). 010_B General Timer (GTM) trigger event enabled Node i Timer is decremented per GTM trigger event prescaled by (TPSC + 1). **others**, Reserved, do not use
0	7:0, 13:12, 17:16, 31:21	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.8 CAN Node timers for pretended networking

Node i Timer A Transmit Trigger Register

The Node i Timer A Transmit Trigger Register NTATTRi controls the node timing functions for Transmit Trigger Mode.

NTATTRi (i=0-3)
Node i Timer A Transmit Trigger Register (008124_H+i*400_H) **Application Reset Value: 0001 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							STRT	TXMO							
r							rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD															
rw															

Field	Bits	Type	Description
RELOAD	15:0	rw	**Reload Value** This bit field contains the reload value for the timer. The timer will restart when RELOAD is written.
TXMO	23:16	r	**Transmit Message Object** This transmit trigger is fixed to transmit buffer 1
STRT	24	rw	**Timer Start** This bit field controls the operation of the timer. 0_B Timer is stopped. 1_B Timer is started.

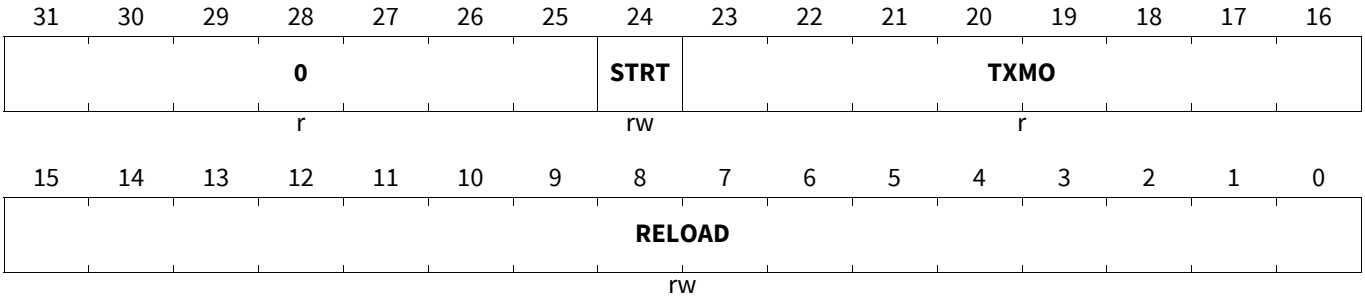
Field	Bits	Type	Description
0	31:25	r	**Reserved** Shall read 0; shall be written with 0.

Node i Timer B Transmit Trigger Register

The Node i Timer B Transmit Trigger Register NTBTTRi controls the node timing functions for Transmit Trigger Mode.

NTBTTRi (i=0-3)

Node i Timer B Transmit Trigger Register (008128$_H$+i*400$_H$) Application Reset Value: 0002 0000$_H$



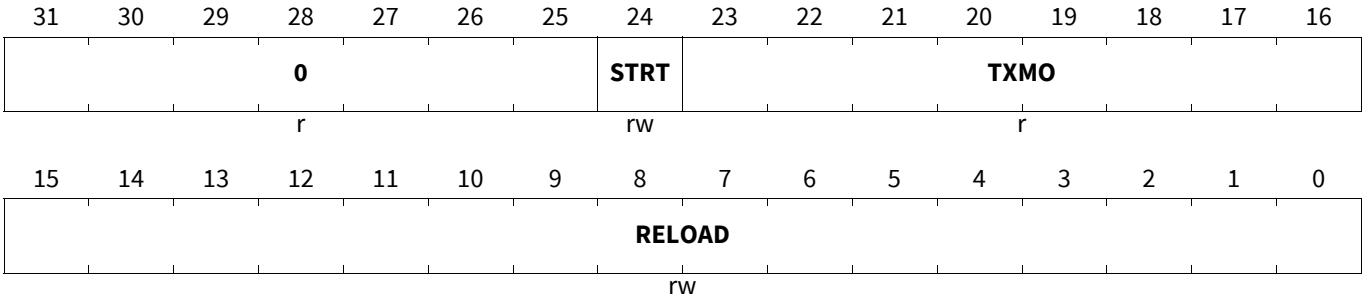
Field	Bits	Type	Description
RELOAD	15:0	rw	**Reload Value** This bit field contains the reload value for the timer. The timer will restart when RELOAD is written.
TXMO	23:16	r	**Transmit Message Object** This transmit object is fixed to transmit buffer 2
STRT	24	rw	**Timer Start** This bit field controls the operation of the timer. 0_B Timer is stopped. 1_B Timer is started.
0	31:25	r	**Reserved** Shall read 0; shall be written with 0.

Node i Timer C Transmit Trigger Register

The Node i Timer C Transmit Trigger Register NTCTTRi controls the node timing functions for Transmit Trigger Mode.

NTCTTRi (i=0-3)

Node i Timer C Transmit Trigger Register (00812C$_H$+i*400$_H$) Application Reset Value: 0003 0000$_H$



Field	Bits	Type	Description
RELOAD	15:0	rw	**Reload Value** This bit field contains the reload value for the timer. The timer will restart when RELOAD is written.
TXMO	23:16	r	**Transmit Message Object** This transmit trigger is fixed to transmit buffer 3
STRT	24	rw	**Timer Start** This bit field controls the operation of the timer. 0_B Timer is stopped. 1_B Timer is started.
0	31:25	r	**Reserved** Shall read 0; shall be written with 0.

40.4.4.9 Node Timer Receive Timerout Register

Node i Timer Receive Timeout Register

The Node i Timer Receive Timeout Register NTRTRi controls the node timing functions for Receive Timeout Mode. This feature is independent of Classical CAN and CAN FD.

This mode exists, to have for example network management supervision.

NTRTRi (i=0-3)
Node i Timer Receive Timeout Register **(008130_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								TE	TEIE	0					
r								rwh	rw	r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD															
rw															

Field	Bits	Type	Description
RELOAD	15:0	rw	**Reload Value** This bit field contains the reload value for the timer. The timer will start when RELOAD ≠ 0 is written. After half the time of the RELOAD value, the interrupt flags of the receive buffers will be cleared automatically, to ensure, that no message receive will be missed.
TEIE	22	rw	**Timer Event Interrupt Enable** This bit enables the node timer event interrupt of CAN node i. Bit field GRINT2.RETI selects the interrupt output line which becomes activated at this type of interrupt. 0_B Timer event interrupt is disabled 1_B Timer event interrupt is enabled

Field	Bits	Type	Description
TE	23	rwh	**Timer Event** This flag is set on a node timer transition from 1 to 0 in Receive Timeout Mode. This bit must be reset (i.e Write to '0') by software, writing a '1' has no effect. An interrupt request is generated if TEIE = 1. 0_B No timer event has occurred since last flag reset 1_B Timer event has occurred since last flag reset
0	21:16, 31:24	r	**Reserved** Shall read 0; shall be written with 0.

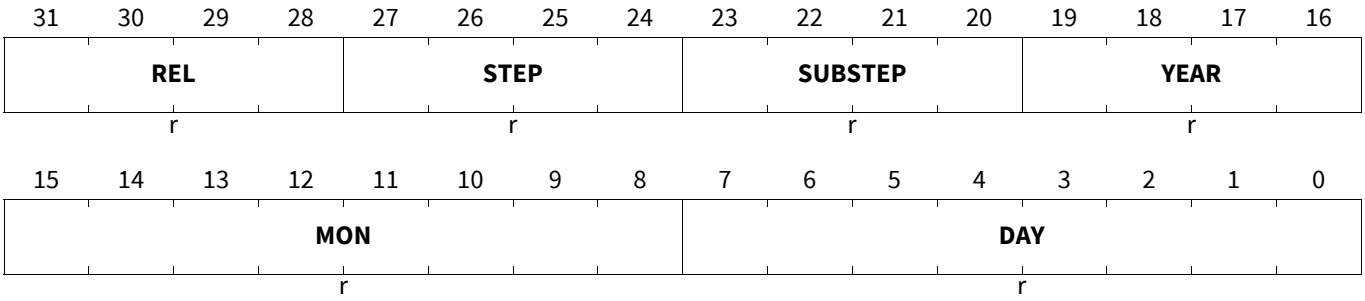
40.4.5 Registers within M_CAN

40.4.5.1 Standard Registers

Core Release Register i

CRELi (i=0-3)

Core Release Register i **(008200_H+i*400_H)** **Reset Value: Table 381**



Field	Bits	Type	Description
DAY	7:0	r	**Time Stamp Day**
MON	15:8	r	**Time Stamp Month**
YEAR	19:16	r	**Time Stamp Year**
SUBSTEP	23:20	r	**Sub-step of Core Release** One digit, BCD-coded.
STEP	27:24	r	**Step of Core Release** One digit, BCD-coded.
REL	31:28	r	**Core Release** One digit, BCD-coded.

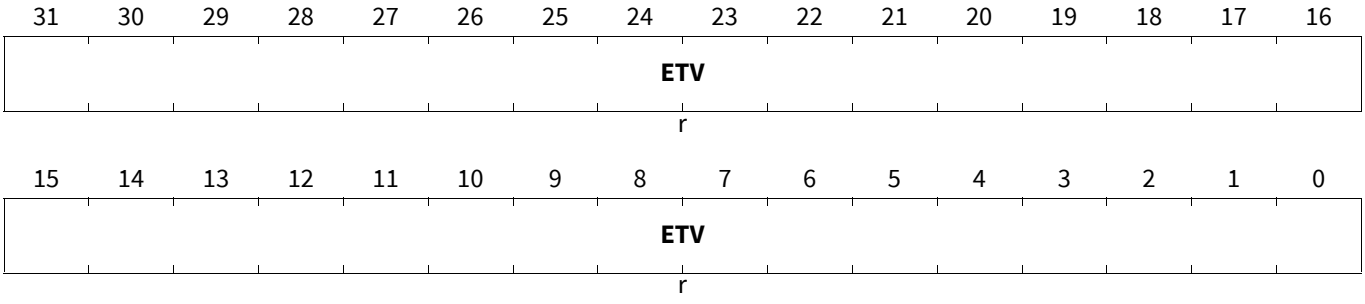
Table 381 Reset Values of CRELi (i=0-3)

Reset Type	Reset Value	Note
Application Reset	$3215\ 0323_H$	Node (i = 0) with TTCAN
Application Reset	$3215\ 0320_H$	Nodes (i > 0) without TTCAN

Endian Register i

ENDNi (i=0-3)

Endian Register i **(008204_H+i*400_H)** **Application Reset Value: $8765\ 4321_H$**



Field	Bits	Type	Description
ETV	31:0	r	**Endianness Test Value** The endianness test value is 0x87654321.

Data Bit Timing & Prescaler Register i

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programmed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 clock cycles. t_q = (DBRP + 1) clock cycles.

DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [DTSEG1 + DTSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

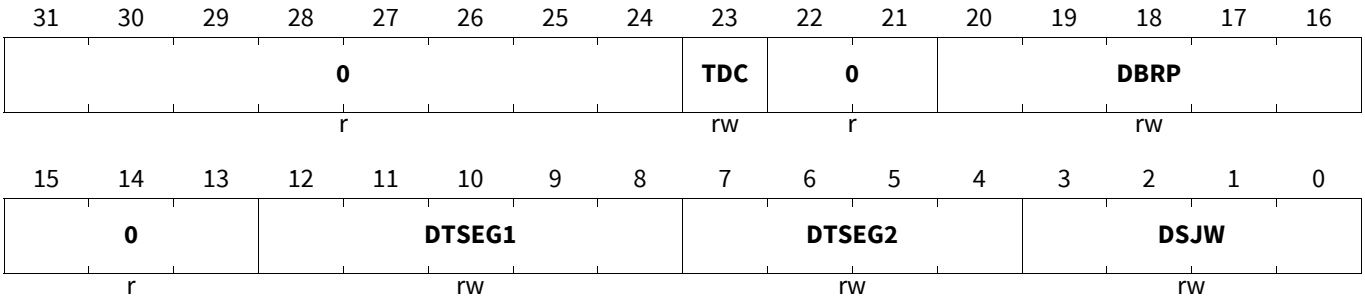
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

Notes

1. *With a CAN clock of 8 MHz, the reset value of 0x00000A33 configures the M_CAN for a fast bit rate of 500 kbit/s.*
2. *The bit rate configured for the CAN FD data phase via DBTP must be higher or equal to the bit rate configured for the arbitration phase via NBTP.*

DBTPi (i=0-3)

Data Bit Timing & Prescaler Register i **($00820C_H$+i*400_H)** **Application Reset Value: 0000 $0A33_H$**



Field	Bits	Type	Description
DSJW	3:0	rw	**Data (Re) Synchronization Jump Width** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
DTSEG2	7:4	rw	**Data time segment after sample point** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

Field	Bits	Type	Description
DTSEG1	12:8	rw	**Data time segment before sample point** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.
DBRP	20:16	rw	**Data Baud Rate Prescaler** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
TDC	23	rw	**Transmitter Delay Compensation** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Transmitter Delay Compensation disabled 1_B Transmitter Delay Compensation enabled
0	15:13, 22:21, 31:24	r	**Reserved** Shall read 0, shall be written with 0.

Test Register i

Write access to the Test Register has to be enabled by setting bit CCCR.TEST to ‘1’. All Test Register functions are set to their reset values when bit CCCR.TEST is reset.

Loop Back Mode and software control of transmit pin are hardware test modes. Programming of TEST.TX ≠ “00” may disturb the message transfer on the CAN bus.

TESTi (i=0-3)
Test Register i **(008210_H+i*400_H)** **Application Reset Value: 0000 00X0$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								RX	TX		LBCK	0	0	0	0
r								rh	rwh		rwh	r	r	r	r

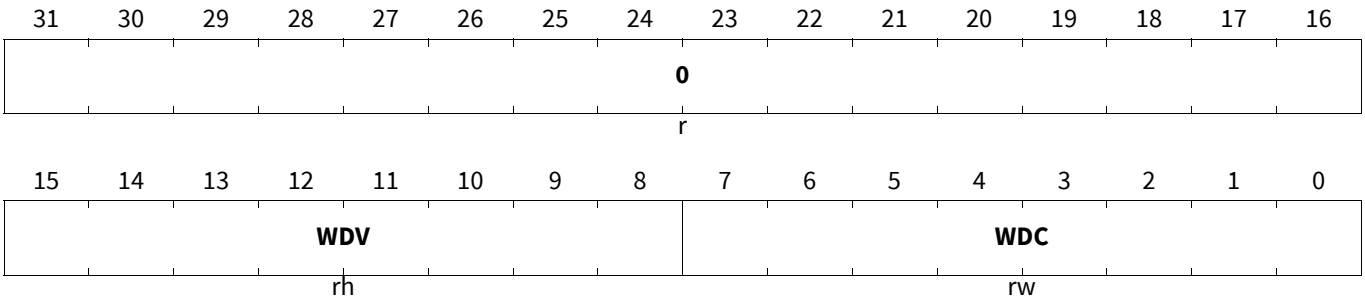
Field	Bits	Type	Description
LBCK	4	rwh	**Loop Back Mode** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. This is the external loop back mode, visible on the outside. 0_B Reset value, Loop Back Mode is disabled 1_B Loop Back Mode is enabled
TX	6:5	rwh	**Control of Transmit Pin** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_B Reset value, TX pin controlled by the CAN Core, updated at the end of the CAN bit time 01_B Sample Point can be monitored at the TX pin 10_B Dominant (‘0’) level at TX pin. 11_B Recessive (‘1’) at RX pin.
RX	7	rh	**Receive Pin** Monitors the actual value of RX pin. 0_B The CAN bus is dominant (RXD = ‘0’) 1_B The CAN bus is recessive (RXD = ‘1’)
0	0, 1, 2, 3, 31:8	r	**Reserved** Shall read 0, shall be written with 0.

RAM Watchdog i

The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the M_CAN’s Generic Master Interface starts the Message RAM Watchdog Counter with the value configured by RWD.WDC. The counter is reloaded with RWD.WDC when the Message RAM signals successful completion. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag IR.WDI is set. The RAM Watchdog Counter is clocked by the Host clock.

RWDi (i=0-3)

RAM Watchdog i **(008214_H+i*400_H)** **Application Reset Value: 0000 0000_H**



Field	Bits	Type	Description
WDC	7:0	rw	**Watchdog Configuration** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start value of the Message RAM Watchdog Counter. With the reset value of “00” the counter is disabled. 00_H Watchdog disabled **others**, Start value of the Message RAM Watchdog Counter
WDV	15:8	rh	**Watchdog Value** Actual Message RAM Watchdog Counter Value.
0	31:16	r	**Reserved** Shall read 0, shall be written with 0.

CC Control Register i

The CCCRi register enables and disables CAN bus participation and basic protocol functions. Due to synchronization mechanisms between the clock domains, after a write operation to CCCRi, the register shall be read back, until the set values are written to the register. Please keep in mind, that the register also includes hardware influenced bits.

Note: After enabling the CAN clocks in MCR register, the application software has to wait for 10 hostclock cycles before accessing the kernel registers.

Note: LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for rwh bits in this register.

CCCRi (i=0-3)
CC Control Register i **(008218_H+i*400_H)** **Application Reset Value: 0000 0001_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NISO	TXP	EFBI	PXHD	0		BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
rw	rw	rw	rw	r		rw	rw	rw	rw	rwh	rw	rh	rwh	rw	rwh

Field	Bits	Type	Description
INIT	0	rwh	**Initialization** *Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.* 0_B Normal Operation 1_B Initialization is started
CCE	1	rw	**Configuration Change Enable** 0_B The CPU has no write access to the protected configuration registers 1_B The CPU has write access to the protected configuration registers (while CCCR.INIT = '1')
ASM	2	rwh	**Restricted Operation Mode** Bit ASM can only be set by the Host when both CCE and INIT are set to '1'. In can also be set by the M_CAN. The bit can be reset by the Host at any time. For a description of the Restricted Operation Mode see paragraph Restricted Operation Mode. 0_B Normal CAN operation 1_B Restricted Operation Mode active
CSA	3	rh	**Clock Stop Acknowledge** 0_B No clock stop acknowledged 1_B M_CAN may be set in power down by stopping the synchronous and the asynchronous clock source
CSR	4	rw	**Clock Stop Request** 0_B No clock stop is requested 1_B Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.
MON	5	rwh	**Bus Monitoring Mode** Bit MON can only be set by the Host when both CCE and INIT are set to '1'. The bit can be reset by the Host at any time. 0_B Bus Monitoring Mode is disabled 1_B Bus Monitoring Mode is enabled
DAR	6	rw	**Disable Automatic Retransmission** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Automatic retransmission of messages not transmitted successfully enabled 1_B Automatic retransmission disabled

Field	Bits	Type	Description
TEST	7	rw	**Test Mode Enable** The TEST register can only be set, if CCE, INIT and TEST are set. Writes to test will only have effect, if all three bits are set. 0_B Normal operation, register TEST holds reset values 1_B Test Mode, write access to register TEST enabled
FDOE	8	rw	**FD Operation Enable** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B CAN FD frame format disabled. 1_B CAN FD frame format enabled.
BRSE	9	rw	**Bit Rate Switch Enable** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Bit rate switching for transmission disabled. 1_B Bit rate switching for transmission enabled.
PXHD	12	rw	**Protocol Exception Handling Disable** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Protocol exception handling enabled. 1_B Protocol exception handling disabled. (When protocol exception handling is disabled, the M_CAN will transmit an error frame when it detects a protocol exception condition.
EFBI	13	rw	**Edge Filtering during Bus Integration** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Edge filter disabled 1_B Two consecutive dominant tq required to detect an edge for hard synchronization.
TXP	14	rw	**Transmit Pause** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. If this bit is set, the M_CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see **Tx Handling**). 0_B Transmit pause disabled 1_B Transmit pause enabled
NISO	15	rw	**Non ISO Operation** If this bit is set, the M_CAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0. 0_B CAN FD frame format according to ISO11898-1 1_B CAN FD frame format according to Bosch CAN FD Specification V1.0
0	11:10, 31:16	r	**Reserved** Shall read 0, shall be written with 0.

Nominal Bit Timing & Prescaler Register i

This register is only writable if bits CCCR.CCE and CCCR.INIT are set.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 clock periods. t_q = (NBRP + 1) clock periods.

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

Note: With a CAN clock of 8 MHz, the reset value of 0600_H0A03 configures the M_CAN for a bit rate of 500 kbit/s.

NBTPi (i=0-3)

Nominal Bit Timing & Prescaler Register i (00821C$_H$+i*400$_H$) **Application Reset Value: 0600 0A03$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSJW							NBRP								
rw							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NTSEG1								0	NTSEG2						
rw								r	rw						

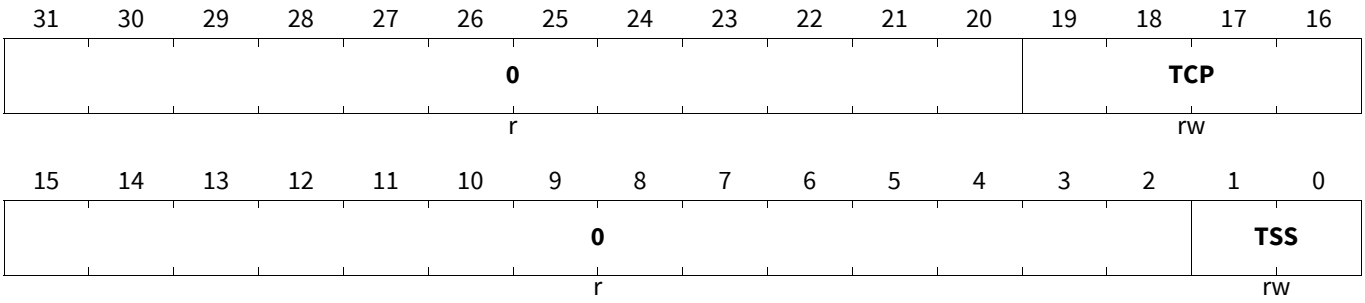
Field	Bits	Type	Description
NTSEG2	6:0	rw	**Nominal Time segment after sample point** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.
NTSEG1	15:8	rw	**Nominal Time segment before sample point** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.
NBRP	24:16	rw	**Baud Rate Prescaler** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
NSJW	31:25	rw	**(Re) Synchronization Jump Width** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Field	Bits	Type	Description
0	7	r	**Reserved** Shall read 0, shall be written with 0.

Timestamp Counter Configuration i

For a description of the Timestamp Counter see chapter Timestamp Generation

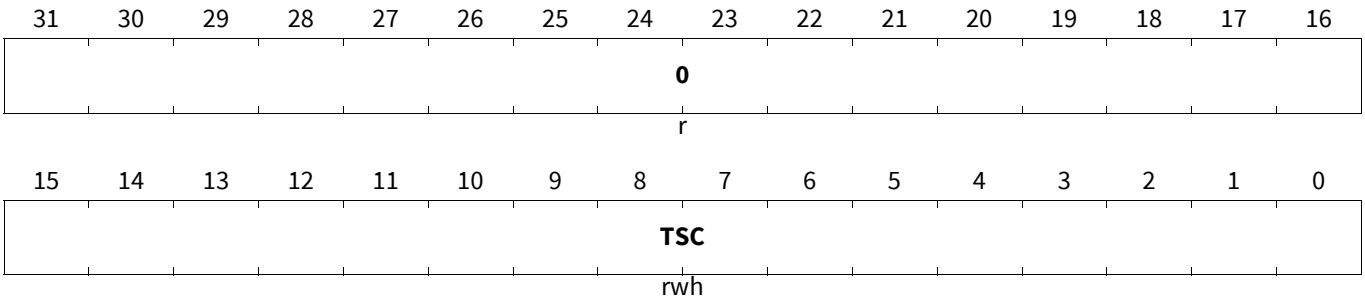
TSCCi (i=0-3)
Timestamp Counter Configuration i **(008220_H+i*400_H)** **Application Reset Value: 0000 0000_H**



Field	Bits	Type	Description
TSS	1:0	rw	**Time segment before sample point** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 01_B Timestamp counter value incremented according to TCP 10_B External timestamp counter value used, timer to be started in NTCCRy, the clock source as well as the chosen prescaler has to be configured before using this feature. **others**, Timestamp counter value always 0x0000
TCP	19:16	rw	**Timestamp Counter Prescaler** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1…16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. *Note: With CAN FD an external counter is required for timestamp generation (TSS = "10")*
0	15:2, 31:20	r	**Reserved** Shall read 0, shall be written with 0.

Timestamp Counter Value i

TSCVi (i=0-3)
Timestamp Counter Value i **(008224_H+i*400_H)** **Application Reset Value: 0000 0000_H**

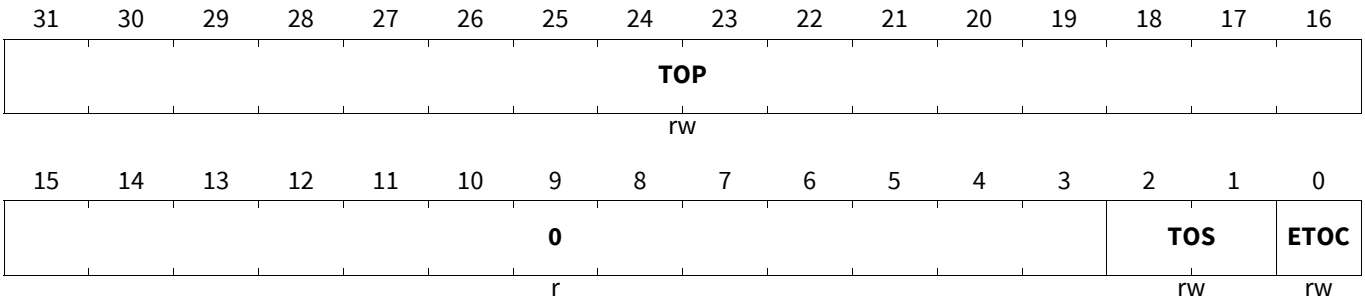


Field	Bits	Type	Description
TSC	15:0	rwh	**Timestamp Counter** The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC.TSS = “01”, the Timestamp Counter is incremented in multiples of CAN bit times [1…16] depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW. Write access resets the counter to zero. When TSCC.TSS = “10”, TSC reflects the external Timestamp Counter value. A write access has no impact. *Note: A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by write access to TSCV.*
0	31:16	r	**Reserved** Shall read 0, shall be written with 0.

Timeout Counter Configuration i

For a description of the Timeout Counter see **Timeout Counter**

TOCCi (i=0-3)
Timeout Counter Configuration i **(008228_H+i*400_H)** **Application Reset Value: FFFF 0000_H**

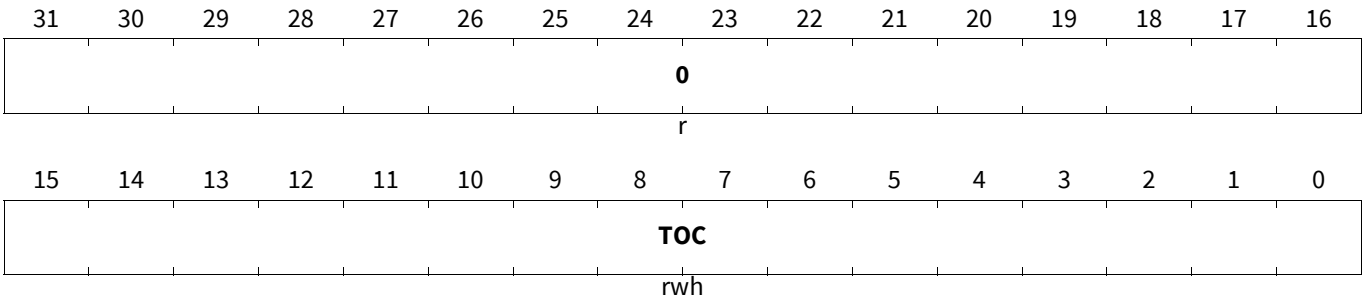


Field	Bits	Type	Description
ETOC	0	rw	**Enable Timeout Counter** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. *Note: For use of timeout function with CAN FD see chapter Timeout Counter.* 0_B Timeout Counter disabled 1_B Timeout Counter enabled
TOS	2:1	rw	**Timeout Select** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored. 00_B Continuous operation 01_B Timeout controlled by Tx Event FIFO 10_B Timeout controlled by Rx FIFO 0 11_B Timeout controlled by Rx FIFO 1
TOP	31:16	rw	**Timeout Period** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start value of the Timeout Counter (down-counter). Configures the Timeout Period.
0	15:3	r	**Reserved** Shall read 0, shall be written with 0.

Timeout Counter Value i

TOCVi (i=0-3)

Timeout Counter Value i **($00822C_H$+i*400_H)** **Application Reset Value: 0000 FFFF$_H$**



Field	Bits	Type	Description
TOC	15:0	rwh	**Timeout Counter** The Timeout Counter is decremented in multiples of CAN bit times [1…16] depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS. Any write access will lead to clearing of the counter.
0	31:16	r	**Reserved** Shall read 0, shall be written with 0.

Error Counter Register i

ECRi (i=0-3)
Error Counter Register i **(008240_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								CEL							
r								rh							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC							TEC							
rh	rh							rh							

Field	Bits	Type	Description
TEC	7:0	rh	**Transmit Error Counter** Actual state of the Transmit Error Counter, values between 0 and 255 *Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.*
REC	14:8	rh	**Receive Error Counter** Actual state of the Receive Error Counter, values between 0 and 127 *Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.*
RP	15	rh	**Receive Error Passive** 0_B The Receive Error Counter is below the error passive level of 128 1_B The Receive Error Counter has reached the error passive level of 128

Field	Bits	Type	Description
CEL	23:16	rh	**CAN Error Logging** The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO. The counter is reset on read, if the bit NPCRi.DELE is set for the node.
0	31:24	r	**Reserved** Shall read 0, shall be written with 0.

Protocol Status Register i

PSRi (i=0-3)

Protocol Status Register i **(008244_H+i*400_H)** **Application Reset Value: 0000 0707_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									TDCV						
r									r						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PXE	RFDF	RBRS	RESI	DLEC			BO	EW	EP	ACT		LEC		
r	rh	rh	rh	rh	rh			rh	rh	rh	rh		rh		

Field	Bits	Type	Description
LEC	2:0	rh	**Last Error Code** The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. This bit field is set to 0x7 on read, if NPCRi.DELE is set. *Note: The Bus_Off recovery sequence (see ISO11898-1) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences.* 000_B No Error: No error occurred since LEC has been reset by successful reception or transmission. 001_B Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 010_B Form Error: A fixed format part of a received frame has the wrong format. 011_B Ack Error: The message transmitted by the M_CAN was not acknowledged by another node. 100_B Bit1 Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 101_B Bit0 Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 110_B CRC Error: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. 111_B No Change: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

Field	Bits	Type	Description
ACT	4:3	rh	**Activity** Monitors the module's CAN communication state. *Note: ACT is set to "00" by a Protocol Exception Event.* 00_B Synchronizing - node is synchronizing on CAN communication 01_B Idle - node is neither receiver nor transmitter 10_B Receiver - node is operating as receiver 11_B Transmitter - node is operating as transmitter
EP	5	rh	**Error Passive** 0_B The M_CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected 1_B The M_CAN is in the Error_Passive state
EW	6	rh	**Warning Status** 0_B Both error counters are below the Error_Warning limit of 96 1_B At least one of error counter has reached the Error_Warning limit of 96
BO	7	rh	**Bus_Off Status** 0_B The M_CAN is not in Bus_Off[1)] 1_B The M_CAN is in Bus_Off state
DLEC	10:8	rh	**Data Phase Last Error Code** Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error. This bit field is set to 0x7 on read, if NPCRi.DELE is set. *Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.*
RESI	11	rh	**ESI flag of last received CAN FD Message** This bit is set together with REDF, independent of acceptance filtering. This bit is reset after read access, if NPCRi.DELE is set. 0_B Last received CAN FD message did not have its ESI flag set 1_B Last received CAN FD message had its ESI flag set
RBRS	12	rh	**BRS flag of last received CAN FD Message** This bit is set together with REDF, independent of acceptance filtering. This bit is reset after read access, if NPCRi.DELE is set. 0_B Last received CAN FD message did not have its BRS flag set 1_B Last received CAN FD message had its BRS flag set

Field	Bits	Type	Description
RFDF	13	rh	**Received a CAN FD Message** This bit is set independent of acceptance filtering. This bit is reset after read access, if NPCRi.DELE is set. 0_B Since this bit was reset by the CPU, no CAN FD message has been received 1_B Message in CAN FD format with FDF flag set, has been received
PXE	14	rh	**Protocol Exception Event** This bit is reset after read access, if NPCRi.DELE is set. 0_B No protocol exception event occurred since last read access 1_B Protocol exception event occurred
TDCV	22:16	r	**Transmitter Delay Compensation Value** Position of the secondary sample point, defined by the sum of the measured delay from TX to RX and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.
0	15, 31:23	r	**Reserved** Shall read 0, shall be written with 0.

1) The Bus_Off recovery sequence (see ISO11898-1) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences.

Transmitter Delay Compensation Register i

TDCRi (i=0-3)

Transmitter Delay Compensation Register i(008248_H+i*400_H) **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TDCO							0	TDCF						
r	rw							r	rw						

Field	Bits	Type	Description
TDCF	6:0	rw	**Transmitter Delay Compensation Filter Window Length** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Defines the minimum value for the Secondary Sample Point position, dominant edges on RX that would result in an earlier Secondary Sample Point position are ignored for transmitter delay measurement. This feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are from 0 to 127 mtq.

Field	Bits	Type	Description
TDCO	14:8	rw	**Transmitter Delay Compensation Offset** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Offset value defining the distance between the measured delay from TX to RX and the secondary sample point. Valid values are 0 to 127 mtq. The duration of one mtq is equal to the fASYNi clock period.
0	7, 31:15	r	**Reserved** Shall read 0, shall be written with 0.

Interrupt Register i

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host clears them. A flag is cleared by writing a “1” to the corresponding bit position. Writing a “0” has no effect. The configuration of IE controls whether an interrupt is generated.

Note: LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for all rwh bits in this register.

IRi (i=0-3)
Interrupt Register i **(008250_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		**0**	**PED**	**PEA**	**WDI**	**BO**	**EW**	**EP**	**ELO**	**0**	**0**	**DRX**	**TOO**	**MRAF**	**TSW**
r		r	rwh	rwh	rwh	rwh	rwh	rwh	rwh	r	r	rwh	rwh	rwh	rwh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFL	**TEFF**	**TEFW**	**TEFN**	**TFE**	**TCF**	**TC**	**HPM**	**RF1L**	**RF1F**	**RF1W**	**RF1N**	**RF0L**	**RF0F**	**RF0W**	**RF0N**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
RF0N	0	rwh	**Rx FIFO 0 New Message** 0_B No new message written to Rx FIFO 0 1_B New message written to Rx FIFO 0
RF0W	1	rwh	**Rx FIFO 0 Watermark Reached** 0_B Rx FIFO 0 fill level below watermark 1_B Rx FIFO 0 fill level reached watermark
RF0F	2	rwh	**Rx FIFO 0 Full** 0_B Rx FIFO 0 not full 1_B Rx FIFO 0 full
RF0L	3	rwh	**Rx FIFO 0 Message Lost** 0_B No Rx FIFO 0 message lost 1_B Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero
RF1N	4	rwh	**Rx FIFO 1 New Message** 0_B No new message written to Rx FIFO 1 1_B New message written to Rx FIFO 1

Field	Bits	Type	Description
RF1W	5	rwh	**Rx FIFO 1 Watermark Reached** 0_B Rx FIFO 1 fill level below watermark 1_B Rx FIFO 1 fill level reached watermark
RF1F	6	rwh	**Rx FIFO 1 Full** 0_B Rx FIFO 1 not full 1_B Rx FIFO 1 full
RF1L	7	rwh	**Rx FIFO 1 Message Lost** 0_B No Rx FIFO 1 message lost 1_B Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero
HPM	8	rwh	**High Priority Message** 0_B No high priority message received 1_B High priority message received
TC	9	rwh	**Transmission Completed** 0_B No transmission completed 1_B Transmission completed
TCF	10	rwh	**Transmission Cancellation Finished** 0_B No transmission cancellation finished 1_B Transmission cancellation finished
TFE	11	rwh	**Tx FIFO Empty** 0_B Tx FIFO non-empty 1_B Tx FIFO empty
TEFN	12	rwh	**Tx Event FIFO New Entry** 0_B Tx Event FIFO unchanged 1_B Tx Handler wrote Tx Event FIFO element
TEFW	13	rwh	**Tx Event FIFO Watermark Reached** 0_B Tx Event FIFO fill level below watermark 1_B Tx Event FIFO fill level reached watermark
TEFF	14	rwh	**Tx Event FIFO Full** 0_B Tx Event FIFO not full 1_B Tx Event FIFO full
TEFL	15	rwh	**Tx Event FIFO Element Lost** 0_B No Tx Event FIFO element lost 1_B Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero
TSW	16	rwh	**Timestamp Wraparound** 0_B No timestamp counter wrap-around 1_B Timestamp counter wrapped around

Field	Bits	Type	Description
MRAF	17	rwh	**Message RAM Access Failure** The flag is set, when the Rx Handler • has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. • was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the M_CAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM. 0_B No Message RAM access failure occurred 1_B Message RAM access failure occurred
TOO	18	rwh	**Timeout Occurred** 0_B No timeout 1_B Timeout reached
DRX	19	rwh	**Message stored to Dedicated Rx Buffer** The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0_B No Rx Buffer updated 1_B At least one received message stored into an Rx Buffer
ELO	22	rwh	**Error Logging Overflow** 0_B CAN Error Logging Counter did not overflow 1_B Overflow of CAN Error Logging Counter occurred
EP	23	rwh	**Error Passive** 0_B Error_Passive status unchanged 1_B Error_Passive status changed
EW	24	rwh	**Warning Status** 0_B Error_Warning status unchanged 1_B Error_Warning status changed
BO	25	rwh	**Bus_Off Status** 0_B Bus_Off status unchanged 1_B Bus_Off status changed
WDI	26	rwh	**Watchdog Interrupt** 0_B No Message RAM Watchdog event occurred 1_B Message RAM Watchdog event due to missing READY

Field	Bits	Type	Description
PEA	27	rwh	**Protocol Error in Arbitration Phase** (Nominal Bit Time is used) 0_B No protocol error in arbitration phase 1_B Protocol error in arbitration phase detected (PSR.LEC ≠ 0,7)
PED	28	rwh	**Protocol Error in Data Phase** (Data Bit Time is used) 0_B No protocol error in data phase detected 1_B Protocol error in data phase detected (PSR.DLEC ≠ 0,7)
0	20, 21, 29, 31:30	r	**Reserved** Shall read 0, shall be written with 0.

Interrupt Enable i

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register will be signalled on an interrupt line.

IEi (i=0-3)

Interrupt Enable i **(008254_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		**0**	**PEDE**	**PEAE**	**WDIE**	**BOE**	**EWE**	**EPE**	**ELOE**	**0**	**0**	**DRXE**	**TOOE**	**MRAFE**	**TSWE**
r		r	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLE	**TEFFE**	**TEFWE**	**TEFNE**	**TFEE**	**TCFE**	**TCE**	**HPME**	**RF1LE**	**RF1FE**	**RF1WE**	**RF1NE**	**RF0LE**	**RF0FE**	**RF0WE**	**RF0NE**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RF0NE	0	rw	**Rx FIFO 0 New Message Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
RF0WE	1	rw	**Rx FIFO 0 Watermark Reached Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
RF0FE	2	rw	**Rx FIFO 0 Full Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
RF0LE	3	rw	**Rx FIFO 0 Message Lost Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled

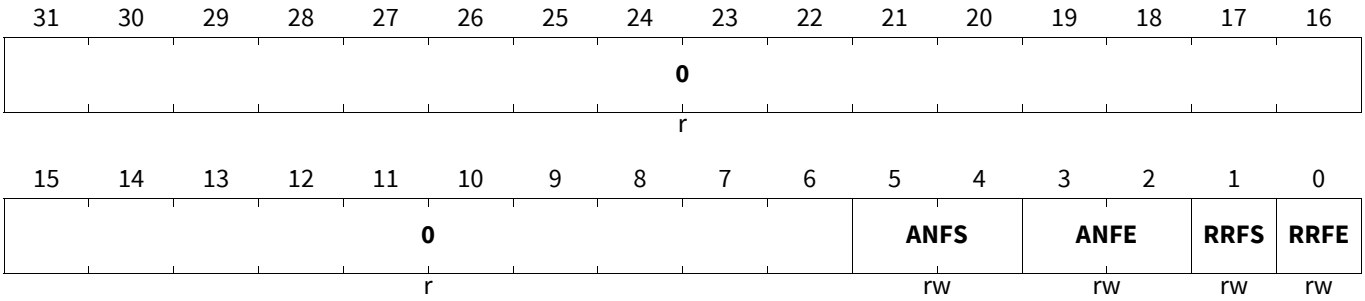
Field	Bits	Type	Description
RF1NE	4	rw	**Rx FIFO 1 New Message Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
RF1WE	5	rw	**Rx FIFO 1 Watermark Reached Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
RF1FE	6	rw	**Rx FIFO 1 Full Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
RF1LE	7	rw	**Rx FIFO 1 Message Lost Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
HPME	8	rw	**High Priority Message Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TCE	9	rw	**Transmission Completed Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TCFE	10	rw	**Transmission Cancellation Finished Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TFEE	11	rw	**Tx FIFO Empty Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TEFNE	12	rw	**Tx Event FIFO New Entry Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TEFWE	13	rw	**Tx Event FIFO Watermark Reached Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TEFFE	14	rw	**Tx Event FIFO Full Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TEFLE	15	rw	**Tx Event FIFO Element Lost Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TSWE	16	rw	**Timestamp Wraparound Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
MRAFE	17	rw	**Message RAM Access Failure Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
TOOE	18	rw	**Timeout Occurred Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled

Field	Bits	Type	Description
DRXE	19	rw	**Message stored to Dedicated Rx Buffer Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
ELOE	22	rw	**Error Logging Overflow Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
EPE	23	rw	**Error Passive Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
EWE	24	rw	**Warning Status Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
BOE	25	rw	**Bus_Off Status Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
WDIE	26	rw	**Watchdog Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled
PEAE	27	rw	**Protocol Error in Arbitration Phase Enable** 0_B Interrupt disabled 1_B Interrupt enabled
PEDE	28	rw	**Protocol Error in Data Phase Enable** 0_B Interrupt disabled 1_B Interrupt enabled
0	20, 21, 29, 31:30	r	**Reserved** Shall read 0, shall be written with 0.

Global Filter Configuration i

Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages as described in **Figure 594** and **Figure 595**.

GFCi (i=0-3)
Global Filter Configuration i **(008280_H+i*400_H)** **Application Reset Value: 0000 0000_H**



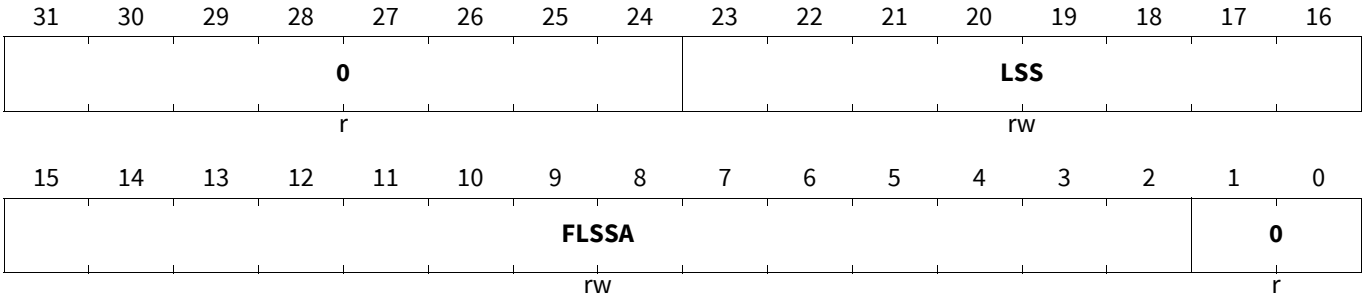
Field	Bits	Type	Description
RRFE	0	rw	**Reject Remote Frames Extended** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Filter remote frames with 29-bit extended IDs 1_B Reject all remote frames with 29-bit extended IDs
RRFS	1	rw	**Reject Remote Frames Standard** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Filter remote frames with 11-bit standard IDs 1_B Reject all remote frames with 11-bit standard IDs
ANFE	3:2	rw	**Accept Non-matching Frames Extended** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 00_B Accept in Rx FIFO 0 01_B Accept in Rx FIFO 1 10_B Reject 11_B Reject
ANFS	5:4	rw	**Accept Non-matching Frames Standard** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 00_B Accept in Rx FIFO 0 01_B Accept in Rx FIFO 1 10_B Reject 11_B Reject
0	31:6	r	**Reserved** Shall read 0, shall be written with 0.

Standard ID Filter Configuration i

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages.

SIDFCi (i=0-3)

Standard ID Filter Configuration i **(008284_H+i*400_H)** **Application Reset Value: 0000 0000_H**

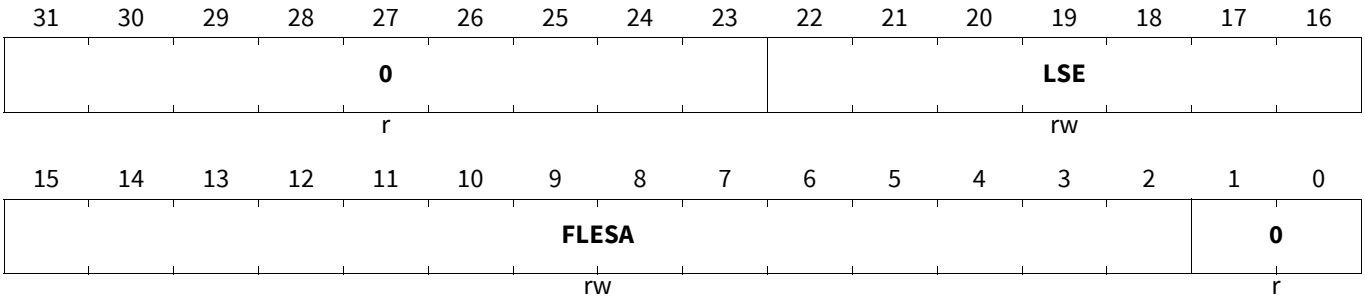


Field	Bits	Type	Description
FLSSA	15:2	rw	**Filter List Standard Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of standard Message ID filter list (32-bit word address).
LSS	23:16	rw	**List Size Standard** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_H No standard Message ID filter 01_H 1 Message ID filter elements … 80_H 128 Message ID filter elements **others**, 128 Message ID filter elements
0	1:0, 31:24	r	**Reserved** Shall read 0, shall be written with 0.

Extended ID Filter Configuration i

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for standard messages as described in **Figure 595**.

XIDFCi (i=0-3)
Extended ID Filter Configuration i **(008288_H+i*400_H)** **Application Reset Value: 0000 0000_H**



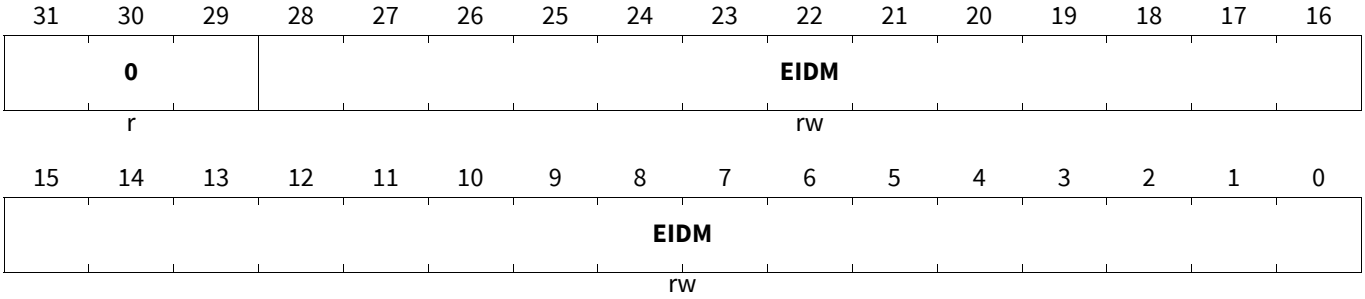
Field	Bits	Type	Description
FLESA	15:2	rw	**Filter List Extended Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of extended Message ID filter list (32-bit word addess).
LSE	22:16	rw	**List Size Extended** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_H No standard Message ID filter 01_H 1 extended Message ID filter element … 40_H 64 extended Message ID filter element **others**, 64 extended Message ID filter elements

Field	Bits	Type	Description
0	1:0, 31:23	r	**Reserved** Shall read 0, shall be written with 0.

Extended ID AND Mask i

XIDAMi (i=0-3)

Extended ID AND Mask i **(008290_H+i*400_H)** **Application Reset Value: 1FFF FFFF$_H$**



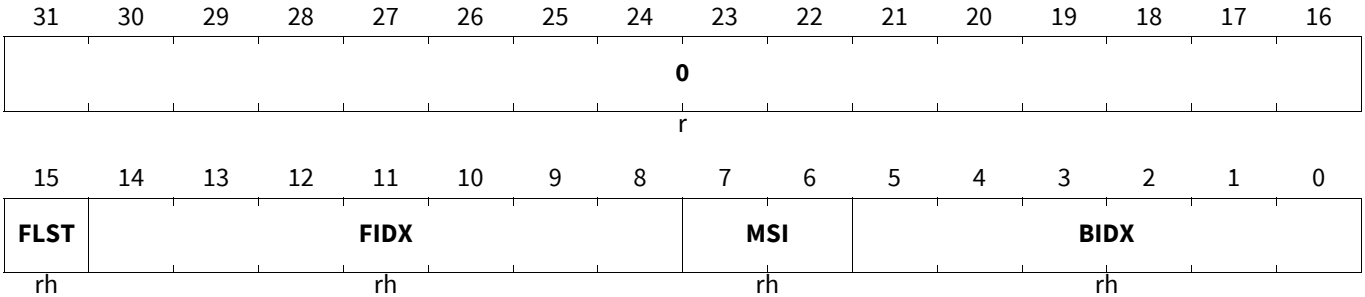
Field	Bits	Type	Description
EIDM	28:0	rw	**Extended ID Mask** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.
0	31:29	r	**Reserved** Shall read 0, shall be written with 0.

High Priority Message Status i

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

HPMSi (i=0-3)

High Priority Message Status i **(008294_H+i*400_H)** **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
BIDX	5:0	rh	**Buffer Index** Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = '1'.
MSI	7:6	rh	**Message Storage Indicator** 00_B No FIFO selected 01_B FIFO message lost 10_B Message stored in FIFO 0 11_B Message stored in FIFO 1
FIDX	14:8	rh	**Filter Index** Index of matching filter element. Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1.
FLST	15	rh	**Filter List** Indicates the filter list of the matching filter element. 0_B Standard Filter List 1_B Extended Filter List
0	31:16	r	**Reserved** Shall read 0, shall be written with 0.

New Data 1 i

The register holds the New Data flags of Rx Buffers 0 to 31.

Note: LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for all rwh bits in this register.

NDAT1i (i=0-3)
New Data 1 i **(008298_H+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND31	**ND30**	**ND29**	**ND28**	**ND27**	**ND26**	**ND25**	**ND24**	**ND23**	**ND22**	**ND21**	**ND20**	**ND19**	**ND18**	**ND17**	**ND16**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND15	**ND14**	**ND13**	**ND12**	**ND11**	**ND10**	**ND9**	**ND8**	**ND7**	**ND6**	**ND5**	**ND4**	**ND3**	**ND2**	**ND1**	**ND0**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
NDy (y=0-31)	y	rwh	**New Data in Rx Buffer y - ND** The flag is set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a "1" to the corresponding bit position. Writing a "0" has no effect. 0_B Rx Buffer not updated 1_B Rx Buffer updated from new message

New Data 2 i

The register holds the New Data flags of Rx Buffers 32 to 63.

Note: LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for all rwh bits in this register.

NDAT2i (i=0-3)
New Data 2 i **($00829C_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
NDy (y=32-63)	y-32	rwh	**New Data in Rx Buffer y - ND** The flag is set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a “1” to the corresponding bit position. Writing a “0” has no effect. 0_B Rx Buffer not updated 1_B Rx Buffer updated from new message

Rx FIFO 0 Configuration i

RXF0Ci (i=0-3)
Rx FIFO 0 Configuration i **($0082A0_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F0OM	F0WM							0	F0S						
rw	rw							r	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0SA														0	
rw														r	

Field	Bits	Type	Description
F0SA	15:2	rw	**Rx FIFO 0 Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of Rx FIFO 0 in Message RAM (32-bit word address, see **Figure 605**).

Field	Bits	Type	Description
F0S	22:16	rw	**Rx FIFO 0 Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_H No Rx FIFO 0 01_H 1 Rx FIFO 0 elements … 40_H 64 Rx FIFO 0 elements **others**, 64 Rx FIFO 0 elements
F0WM	30:24	rw	**Rx FIFO 0 Watermark** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 01_H Level for Rx FIFO 0 watermark interrupt (IR.RF0W) … 40_H Level for Rx FIFO 0 watermark interrupt (IR.RF0W) **others**, Watermark interrupt disabled
F0OM	31	rw	**FIFO 0 Operation Mode** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. FIFO 0 can be operated in blocking or in overwrite mode (see **Rx FIFOs**). 0_B FIFO 0 blocking mode 1_B FIFO 0 overwrite mode
0	1:0, 23	r	**Reserved** Shall read 0, shall be written with 0.

Rx FIFO 0 Status i

RXF0Si (i=0-3)

Rx FIFO 0 Status i **($0082A4_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						RF0L	F0F	0		F0PI					
r						rh	rh	r		rh					

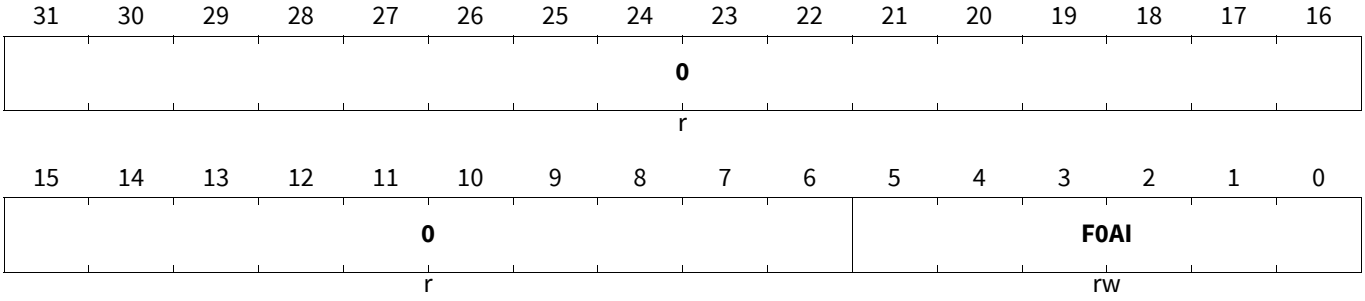
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		F0GI						0		F0FL					
r		rh						r		rh					

Field	Bits	Type	Description
F0FL	6:0	rh	**Rx FIFO 0 Fill Level** Number of elements stored in Rx FIFO 0, range 0 to 64.
F0GI	13:8	rh	**Rx FIFO 0 Get Index** Rx FIFO 0 read index pointer, range 0 to 63.
F0PI	21:16	rh	**Rx FIFO 0 Put Index** Rx FIFO 0 write index pointer, range 0 to 63.

Field	Bits	Type	Description
F0F	24	rh	**Rx FIFO 0 Full** 0_B Rx FIFO 0 not full 1_B Rx FIFO 0 full
RF0L	25	rh	**Rx FIFO 0 Message Lost** This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset. *Note: Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag.* 0_B No Rx FIFO 0 message lost 1_B Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero
0	7, 15:14, 23:22, 31:26	r	**Reserved** Shall read 0, shall be written with 0.

Rx FIFO 0 Acknowledge i

RXF0Ai (i=0-3)
Rx FIFO 0 Acknowledge i **($0082A8_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

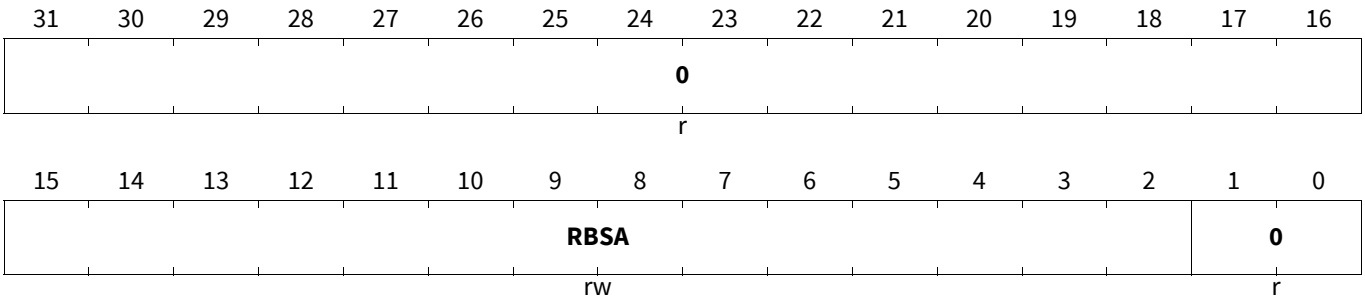


Field	Bits	Type	Description
F0AI	5:0	rw	**Rx FIFO 0 Acknowledge Index** After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL.
0	31:6	r	**Reserved** Shall read 0, shall be written with 0.

Rx Buffer Configuration i

RXBCi (i=0-3)

Rx Buffer Configuration i **(0082AC$_H$+i*400$_H$)** **Application Reset Value: 0000 0000$_H$**

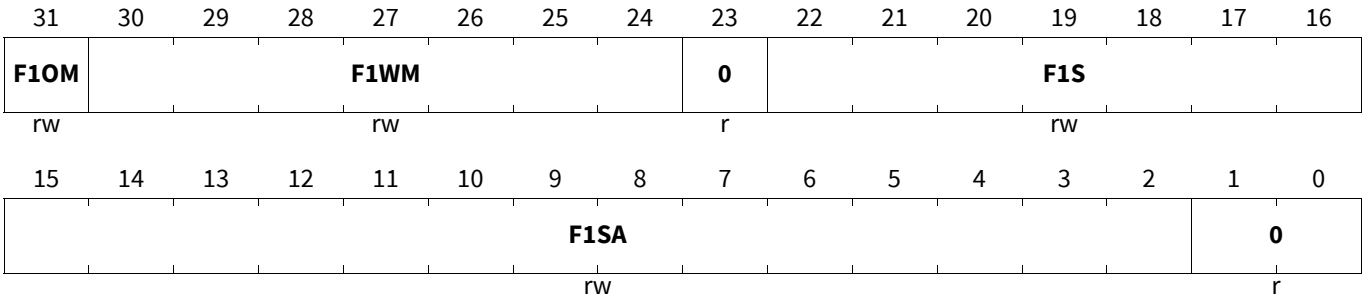


Field	Bits	Type	Description
RBSA	15:2	rw	**Rx Buffer Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).
0	1:0, 31:16	r	**Reserved** Shall read 0, shall be written with 0.

Rx FIFO 1 Configuration i

RXF1Ci (i=0-3)

Rx FIFO 1 Configuration i **(0082B0$_H$+i*400$_H$)** **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
F1SA	15:2	rw	**Rx FIFO 1 Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of Rx FIFO 1 in Message RAM (32-bit word address).

Field	Bits	Type	Description
F1S	22:16	rw	**Rx FIFO 1 Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_H No Rx FIFO 1 01_H 1 Rx FIFO 1 elements … 40_H 64 Rx FIFO 1 elements **others**, 64 Rx FIFO 1 elements
F1WM	30:24	rw	**Rx FIFO 1 Watermark** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 01_H Level for Rx FIFO 1 watermark interrupt (IR.RF1W) … 40_H Level for Rx FIFO 1 watermark interrupt (IR.RF1W) **others**, Watermark interrupt disabled
F1OM	31	rw	**FIFO 1 Operation Mode** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. FIFO 1 can be operated in blocking or in overwrite mode. 0_B FIFO 1 blocking mode 1_B FIFO 1 overwrite mode
0	1:0, 23	r	**Reserved** Shall read 0, shall be written with 0.

Rx FIFO 1 Status i

RXF1Si (i=0-3)

Rx FIFO 1 Status i **(0082B4$_H$+i*400$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		**0**				**RF1L**	**F1F**	**0**		**F1PI**					
r		r				rh	rh	r		rh					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		**F1GI**						**0**		**F1FL**					
r		rh						r		rh					

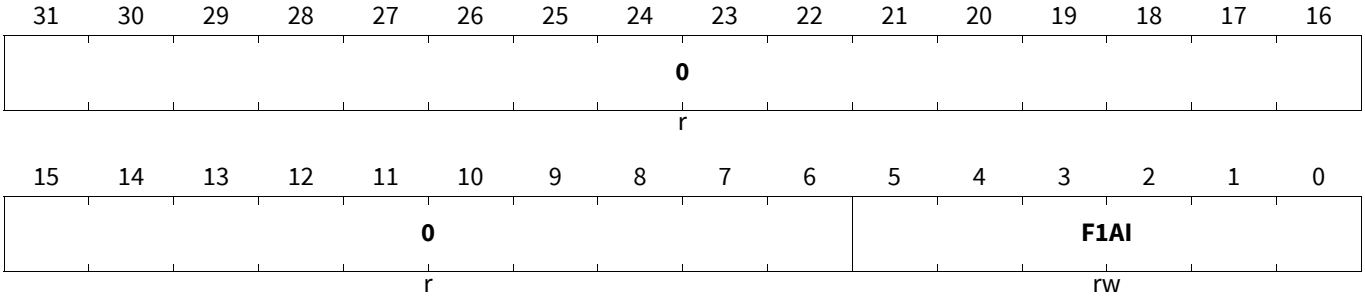
Field	Bits	Type	Description
F1FL	6:0	rh	**Rx FIFO 1 Fill Level** Number of elements stored in Rx FIFO 1, range 0 to 64.
F1GI	13:8	rh	**Rx FIFO 1 Get Index** Rx FIFO 1 read index pointer, range 0 to 63.
F1PI	21:16	rh	**Rx FIFO 1 Put Index** Rx FIFO 1 write index pointer, range 0 to 63.

Field	Bits	Type	Description
F1F	24	rh	**Rx FIFO 1 Full** 0_B Rx FIFO 1 not full 1_B Rx FIFO 1 full
RF1L	25	rh	**Rx FIFO 1 Message Lost** This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset. *Note: Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag.* 0_B No Rx FIFO 1 message lost 1_B Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero
0	7, 15:14, 23:22, 29:26, 31:30	r	**Reserved** Shall read 0, shall be written with 0.

Rx FIFO 1 Acknowledge i

RXF1Ai (i=0-3)

Rx FIFO 1 Acknowledge i **($0082B8_H$+i*400_H)** **Application Reset Value: 0000 0000_H**



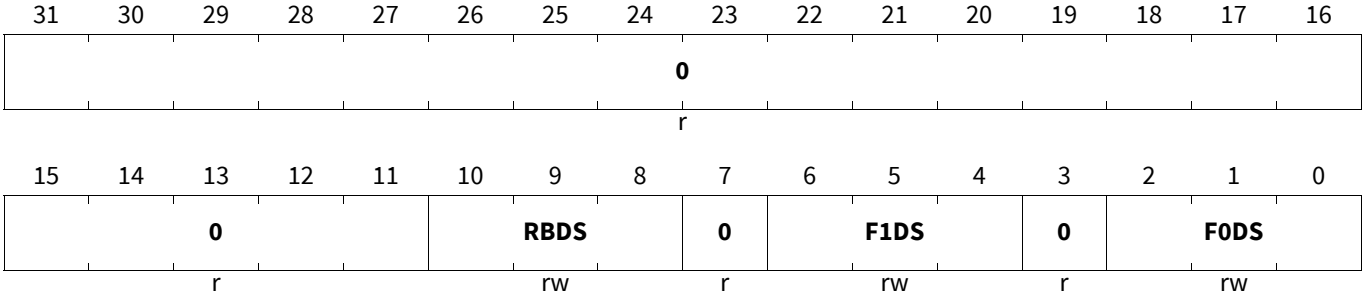
Field	Bits	Type	Description
F1AI	5:0	rw	**Rx FIFO 1 Acknowledge Index** After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL
0	31:6	r	**Reserved** Shall read 0, shall be written with 0.

Rx Buffer/FIFO Element Size Configuration i

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes > 8 bytes are intended for CAN FD operation only.

RXESCi (i=0-3)

Rx Buffer/FIFO Element Size Configuration i(0082BC$_H$+i*400$_H$) **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
F0DS	2:0	rw	**Rx FIFO 0 Data Field Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. *Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.* 000_B 8-byte data field 001_B 12-byte data field 010_B 16-byte data field 011_B 20-byte data field 100_B 24-byte data field 101_B 32-byte data field 110_B 48-byte data field 111_B 64-byte data field
F1DS	6:4	rw	**Rx FIFO 1 Data Field Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 000_B 8-byte data field 001_B 12-byte data field 010_B 16-byte data field 011_B 20-byte data field 100_B 24-byte data field 101_B 32-byte data field 110_B 48-byte data field 111_B 64-byte data field

Field	Bits	Type	Description
RBDS	10:8	rw	**Rx Buffer Data Field Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 000_B 8-byte data field 001_B 12-byte data field 010_B 16-byte data field 011_B 20-byte data field 100_B 24-byte data field 101_B 32-byte data field 110_B 48-byte data field 111_B 64-byte data field
0	3, 7, 31:11	r	**Reserved** Shall read 0, shall be written with 0.

Tx Buffer Configuration i

TXBCi (i=0-3)
Tx Buffer Configuration i **($0082C0_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		**TFQM**		**TFQS**				**0**		**NDTB**					
r		rw		rw				r		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBSA														**0**	
rw														r	

Field	Bits	Type	Description
TBSA	15:2	rw	**Tx Buffers Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of Tx Buffers section in Message RAM (32-bit word address).
NDTB	21:16	rw	**Number of Dedicated Transmit Buffers** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. *Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.* 00_H No Dedicated Tx Buffers 01_H 1 Dedicated Tx Buffers … 20_H 32 Dedicated Tx Buffers **others**, 32 Dedicated Tx Buffers

Field	Bits	Type	Description
TFQS	29:24	rw	**Transmit FIFO/Queue Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_H No Tx FIFO/Queue 01_H 1 Tx Buffers used for Tx FIFO/Queue … 20_H 32 Tx Buffers used for Tx FIFO/Queue **others**, 32 Tx Buffers used for Tx FIFO/Queue
TFQM	30	rw	**Tx FIFO/Queue Mode** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Tx FIFO operation 1_B Tx Queue operation
0	1:0, 23:22, 31	r	**Reserved** Shall read 0, shall be written with 0.

Tx FIFO/Queue Status i

The Tx FIFO/Queue status is related to the pending Tx requests listed in register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

TXFQSi (i=0-3)
Tx FIFO/Queue Status i **($0082C4_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0										TFQF	TFQPI				
r										rh	rh				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			TFGI					0		TFFL					
r			rh					r		rh					

Field	Bits	Type	Description
TFFL	5:0	rh	**Tx FIFO Free Level** Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1') *Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.*

Field	Bits	Type	Description
TFGI	12:8	rh	**Tx FIFO Get Index** Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1').
TFQPI	20:16	rh	**Tx FIFO/Queue Put Index** Tx FIFO/Queue write index pointer, range 0 to 31.
TFQF	21	rh	**Tx FIFO/Queue Full** 0_B Tx FIFO/Queue not full 1_B Tx FIFO/Queue full
0	7:6, 15:13, 31:22	r	**Reserved** Shall read 0, shall be written with 0.

Tx Buffer Element Size Configuration i

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

TXESCi (i=0-3)

Tx Buffer Element Size Configuration i **($0082C8_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0													TBDS		
r													rw		

Field	Bits	Type	Description
TBDS	2:0	rw	**Tx Buffer Data Field Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. *Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as "0xCC" (padding bytes).* 000_B 8-byte data field 001_B 12-byte data field 010_B 16-byte data field 011_B 20-byte data field 100_B 24-byte data field 101_B 32-byte data field 110_B 48-byte data field 111_B 64-byte data field

Field	Bits	Type	Description
0	31:3	r	**Reserved** Shall read 0, shall be written with 0.

Tx Buffer Request Pending i

Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR.

TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan (Tx Handling) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.

After a cancellation has been requested, a finished cancellation is signalled via TXBCF

- after successful transmission together with the corresponding TXBTO bit
- when the transmission has not yet been started at the point of cancellation
- when the transmission has been aborted due to lost arbitration
- when an error occurred during frame transmission

In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions.

Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this "Add Request" is cancelled immediately, the corresponding TXBRP bit is reset.

TXBRPi (i=0-3)
Tx Buffer Request Pending i **($0082CC_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRP31	**TRP30**	**TRP29**	**TRP28**	**TRP27**	**TRP26**	**TRP25**	**TRP24**	**TRP23**	**TRP22**	**TRP21**	**TRP20**	**TRP19**	**TRP18**	**TRP17**	**TRP16**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP15	**TRP14**	**TRP13**	**TRP12**	**TRP11**	**TRP10**	**TRP9**	**TRP8**	**TRP7**	**TRP6**	**TRP5**	**TRP4**	**TRP3**	**TRP2**	**TRP1**	**TRP0**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TRPz (z=0-31)	z	rh	**Transmission Request Pending Tx Buffer z - TRP** 0_B No transmission request pending 1_B Transmission request pending

Tx Buffer Add Request i

Each Tx Buffer has its own "Add Request" bit. Writing a ‘1’ will set the corresponding "Add Request" bit; writing a ‘0’ has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to

TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.

Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored. LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for all rwh bits in this register.

TXBARi (i=0-3)

Tx Buffer Add Request i **($0082D0_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AR31	**AR30**	**AR29**	**AR28**	**AR27**	**AR26**	**AR25**	**AR24**	**AR23**	**AR22**	**AR21**	**AR20**	**AR19**	**AR18**	**AR17**	**AR16**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR15	**AR14**	**AR13**	**AR12**	**AR11**	**AR10**	**AR9**	**AR8**	**AR7**	**AR6**	**AR5**	**AR4**	**AR3**	**AR2**	**AR1**	**AR0**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
ARz (z=0-31)	z	rwh	**Add Request Tx Buffer z - AR** 0_B No transmission request added 1_B Transmission requested added

Tx Buffer Cancellation Request i

Each Tx Buffer has its own Cancellation Request bit. Writing a ‘1’ will set the corresponding Cancellation Request bit; writing a ‘0’ has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.

Note: LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for all rwh bits in this register.

TXBCRi (i=0-3)

Tx Buffer Cancellation Request i **($0082D4_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR31	**CR30**	**CR29**	**CR28**	**CR27**	**CR26**	**CR25**	**CR24**	**CR23**	**CR22**	**CR21**	**CR20**	**CR19**	**CR18**	**CR17**	**CR16**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	**CR14**	**CR13**	**CR12**	**CR11**	**CR10**	**CR9**	**CR8**	**CR7**	**CR6**	**CR5**	**CR4**	**CR3**	**CR2**	**CR1**	**CR0**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
CRz (z=0-31)	z	rwh	**Cancellation Request Tx Buffer z - CR** 0_B No cancellation pending 1_B Cancellation pending

Tx Buffer Transmission Occurred i

Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR.

TXBTOi (i=0-3)
Tx Buffer Transmission Occurred i **($0082D8_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TO31	**TO30**	**TO29**	**TO28**	**TO27**	**TO26**	**TO25**	**TO24**	**TO23**	**TO22**	**TO21**	**TO20**	**TO19**	**TO18**	**TO17**	**TO16**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO15	**TO14**	**TO13**	**TO12**	**TO11**	**TO10**	**TO9**	**TO8**	**TO7**	**TO6**	**TO5**	**TO4**	**TO3**	**TO2**	**TO1**	**TO0**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
TOz (z=0-31)	z	rh	**Transmission Occurred Tx Buffer z - TO** 0_B No transmission occurred 1_B Transmission occurred

Tx Buffer Cancellation Finished i

Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR.

TXBCFi (i=0-3)
Tx Buffer Cancellation Finished i **($0082DC_H$+i*400_H)** **Application Reset Value: 0000 0000_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CF31	**CF30**	**CF29**	**CF28**	**CF27**	**CF26**	**CF25**	**CF24**	**CF23**	**CF22**	**CF21**	**CF20**	**CF19**	**CF18**	**CF17**	**CF16**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF15	**CF14**	**CF13**	**CF12**	**CF11**	**CF10**	**CF9**	**CF8**	**CF7**	**CF6**	**CF5**	**CF4**	**CF3**	**CF2**	**CF1**	**CF0**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
CFz (z=0-31)	z	rh	**Cancellation Finished Tx Buffer z - CF** 0_B No transmit buffer cancellation 1_B Transmit buffer cancellation finished

Tx Buffer Transmission Interrupt Enable i

Each Tx Buffer has its own Transmission Interrupt Enable bit.

TXBTIEi (i=0-3)

Tx Buffer Transmission Interrupt Enable i (0082E0$_H$+i*400$_H$) **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIE31	**TIE30**	**TIE29**	**TIE28**	**TIE27**	**TIE26**	**TIE25**	**TIE24**	**TIE23**	**TIE22**	**TIE21**	**TIE20**	**TIE19**	**TIE18**	**TIE17**	**TIE16**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE15	**TIE14**	**TIE13**	**TIE12**	**TIE11**	**TIE10**	**TIE9**	**TIE8**	**TIE7**	**TIE6**	**TIE5**	**TIE4**	**TIE3**	**TIE2**	**TIE1**	**TIE0**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
TIEz (z=0-31)	z	rw	**Transmission Interrupt Enable Tx Buffer z - TIE** 0_B Transmission interrupt disabled 1_B Transmission interrupt enable

Tx Buffer Cancellation Finished Interrupt Enable i

Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

TXBCIEi (i=0-3)

Tx Buffer Cancellation Finished Interrupt Enable i(0082E4$_H$+i*400$_H$) **Application Reset Value: 0000 0000$_H$**

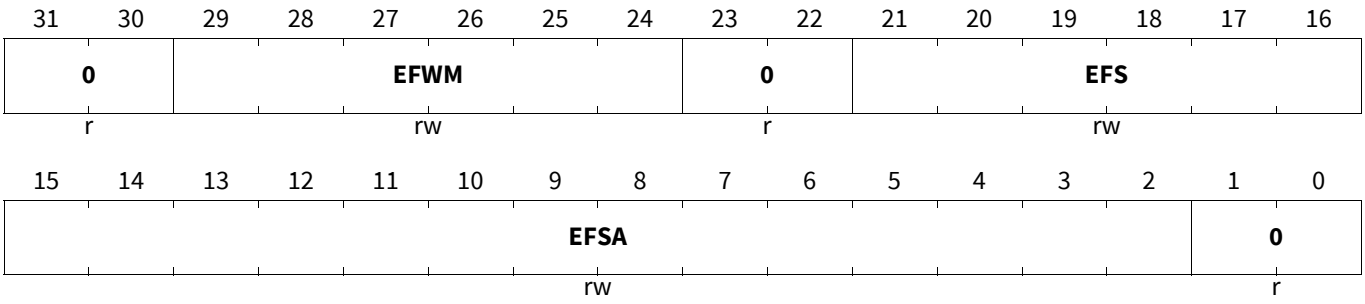
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFIE31	**CFIE30**	**CFIE29**	**CFIE28**	**CFIE27**	**CFIE26**	**CFIE25**	**CFIE24**	**CFIE23**	**CFIE22**	**CFIE21**	**CFIE20**	**CFIE19**	**CFIE18**	**CFIE17**	**CFIE16**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE15	**CFIE14**	**CFIE13**	**CFIE12**	**CFIE11**	**CFIE10**	**CFIE9**	**CFIE8**	**CFIE7**	**CFIE6**	**CFIE5**	**CFIE4**	**CFIE3**	**CFIE2**	**CFIE1**	**CFIE0**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
CFIEz (z=0-31)	z	rw	**Cancellation Finished Interrupt Enable Tx Buffer z - CFIE** 0_B Cancellation finished interrupt disabled 1_B Cancellation finished interrupt enabled

Tx Event FIFO Configuration i

TXEFCi (i=0-3)

Tx Event FIFO Configuration i **(0082F0$_H$+i*400$_H$)** **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
EFSA	15:2	rw	**Event FIFO Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of Tx Event FIFO in Message RAM (32-bit word address).
EFS	21:16	rw	**Event FIFO Size** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. The Tx Event FIFO elements are indexed from 0 to EFS - 1 00_H Tx Event FIFO disabled 01_H 1 Tx Event FIFO elements … 20_H 32 Tx Event FIFO elements **others**, 32 Tx Event FIFO elements
EFWM	29:24	rw	**Event FIFO Watermark** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 01_H Level for Tx Event FIFO watermark interrupt (IR.TEFW) … 20_H Level for Tx Event FIFO watermark interrupt (IR.TEFW) **others**, Watermark interrupt disabled
0	1:0, 23:22, 31:30	r	**Reserved** Shall read 0, shall be written with 0.

Tx Event FIFO Status i

TXEFSi (i=0-3)

Tx Event FIFO Status i **($0082F4_H$+i*400_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0						TEFL	EFF	0			EFPI				
r						rh	rh	r			rh				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			EFGI					0		EFFL					
r			rh					r		rh					

Field	Bits	Type	Description
EFFL	5:0	rh	**Event FIFO Fill Level** Number of elements stored in Tx Event FIFO, range 0 to 32.
EFGI	12:8	rh	**Event FIFO Get Index** Tx Event FIFO read index pointer, range 0 to 31.
EFPI	20:16	rh	**Event FIFO Put Index** Tx Event FIFO write index pointer, range 0 to 31.
EFF	24	rh	**Event FIFO Full** 0_B Tx Event FIFO not full 1_B Tx Event FIFO full
TEFL	25	rh	**Tx Event FIFO Element Lost** This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset. 0_B No Tx Event FIFO element lost 1_B Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
0	7:6, 15:13, 23:21, 31:26	r	**Reserved** Shall read 0, shall be written with 0.

Tx Event FIFO Acknowledge i

TXEFAi (i=0-3)

Tx Event FIFO Acknowledge i **($0082F8_H$+i*400_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0													EFAI		
r													rw		

Field	Bits	Type	Description
EFAI	4:0	rw	**Event FIFO Acknowledge Index** After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the FIFO 0 Fill Level TXEFS.EFFL.
0	31:5	r	**Reserved** Shall read 0, shall be written with 0.

TT Trigger Memory Configuration 0

The TTCAN function exists only on CAN0 node 0.

TTTMC0
TT Trigger Memory Configuration 0 (008300_H) Application Reset Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									**TME**						
r									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMSA														**0**	
rw														r	

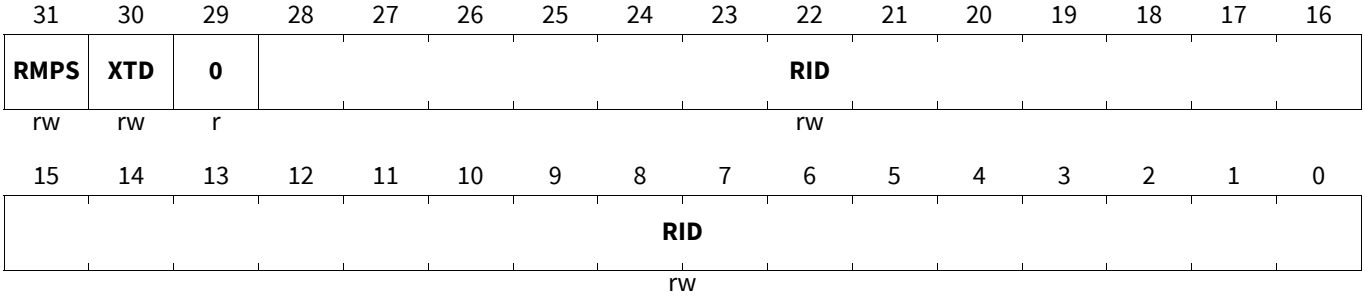
Field	Bits	Type	Description
TMSA	15:2	rw	**Trigger Memory Start Address** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Start address of Trigger Memory in Message RAM.
TME	22:16	rw	**Trigger Memory Elements** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_H No Trigger Memory 01_H 1 Trigger Memory element … 40_H 64 Trigger Memory element **others**, 64 Trigger Memory elements
0	1:0, 31:23	r	**Reserved** Shall read 0, shall be written with 0.

TT Reference Message Configuration 0

The TTCAN function exists only on CAN0 node 0. For details about handling of reference messages.

TTRMC0

TT Reference Message Configuration 0 (008304$_H$) Application Reset Value: 0000 0000$_H$



Field	Bits	Type	Description
RID	28:0	rw	**Reference Identifier** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Identifier transmitted with reference message and used for reference message filtering. Standard or extended reference identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].
XTD	30	rw	**Extended Identifier** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B 11-bit standard identifier 1_B 29-bit extended identifier
RMPS	31	rw	**Reference Message Payload Select** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Ignored in case of time slaves. 0_B Reference message has no additional payload 1_B The following elements are taken from Tx Buffer 0: Message Marker MM, Event FIFO Control EFC, Data Length Code DLC, Data Bytes DB (Level 1: bytes 2-8, Level 0,2: bytes 5-8)
0	29	r	**Reserved** Shall read 0, shall be written with 0.

TT Operation Configuration 0

The TTCAN function exists only on CAN0 node 0.

TTOCF0

TT Operation Configuration 0 **(008308$_H$)** **Application Reset Value: 0001 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0					EVTP	ECC	EGTF	AWL							
r					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EECS	IRTO							LDSDL			TM	GEN	0	OM	
rw	rw							rw			rw	rw	r	rw	

Field	Bits	Type	Description
OM	1:0	rw	**Operation Mode** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 00_B Event-driven CAN communication, default 01_B TTCAN level 1 10_B TTCAN level 2 11_B TTCAN level 0
GEN	3	rw	**Gap Enable** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Strictly time-triggered operation 1_B External event-synchronized time-triggered operation
TM	4	rw	**Time Master** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Time Master function disabled 1_B Potential Time Master
LDSDL	7:5	rw	**LD of Synchronization Deviation Limit** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. The Synchronization Deviation Limit SDL is configured by its dual logarithm LDSDL with SDL = $2^{(LDSDL+5)}$. It should not exceed the clock tolerance given by the CAN bit timing configuration. LD of Synchronization Deviation Limit (SDL ≤ 32…4096)
IRTO	14:8	rw	**Initial Reference Trigger Offset** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Positive offset, range from 0 to 127

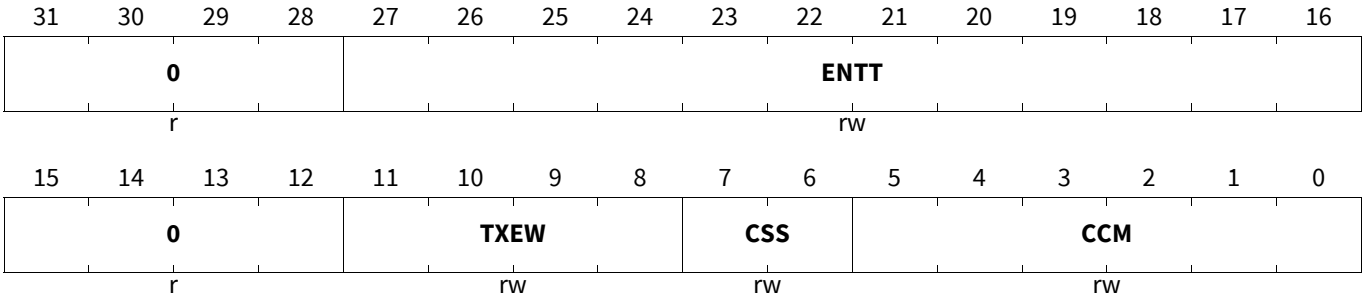
Field	Bits	Type	Description
EECS	15	rw	**Enable External Clock Synchronization** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. If enabled, TUR configuration (TURCF.NCL only) may be updated during TTCAN operation. 0_B External clock synchronization in TTCAN Level 0,2 disabled 1_B External clock synchronization in TTCAN Level 0,2 enabled
AWL	23:16	rw	**Application Watchdog Limit** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. The application watchdog can be disabled by programming AWL to 0x00. Maximum time after which the application has to serve the application watchdog. The application watchdog is incremented once each 256 NTUs.
EGTF	24	rw	**Enable Global Time Filtering** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Global time filtering in TTCAN Level 0,2 is disabled 1_B Global time filtering in TTCAN Level 0,2 is enabled
ECC	25	rw	**Enable Clock Calibration** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Automatic clock calibration in TTCAN Level 0,2 is disabled 1_B Automatic clock calibration in TTCAN Level 0,2 is enabled
EVTP	26	rw	**Event Trigger Polarity** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0_B Rising edge trigger 1_B Falling edge trigger
0	2, 31:27	r	**Reserved** Shall read 0, shall be written with 0.

TT Matrix Limits 0

The TTCAN function exists only on CAN0 node 0.

TTMLM0

TT Matrix Limits 0 **($00830C_H$)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
CCM	5:0	rw	**Cycle Count Max** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. *Note: ISO 11898-4, 5.2.1 requires, that only the listed cycle count values are configured. Other values are possible but may lead to inconsistent matrix cycles.* 00_H 1 Basic Cycle per Matrix Cycle 01_H 2 Basic Cycles per Matrix Cycle 03_H 4 Basic Cycles per Matrix Cycle 07_H 8 Basic Cycles per Matrix Cycle $0F_H$ 16 Basic Cycles per Matrix Cycle $1F_H$ 32 Basic Cycles per Matrix Cycle $3F_H$ 64 Basic Cycles per Matrix Cycle **others**, Reserved
CSS	7:6	rw	**Cycle Start Synchronization** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Enables sync pulse output at start of cycle. 00_B No sync pulse 01_B Sync pulse at start of basic cycle 10_B Sync pulse at start of matrix cycle 11_B Reserved
TXEW	11:8	rw	**Tx Enable Window** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Length of Tx enable window, 1-16 NTU cycles
ENTT	27:16	rw	**Expected Number of Tx Triggers** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Expected number of Tx Triggers in one Matrix Cycle
0	15:12, 31:28	r	**Reserved** Shall read 0, shall be written with 0.

TUR Configuration 0

The TTCAN function exists only on CAN0 node 0.

The length of the NTU is given by: NTU = CAN Clock Period x NC / DC

NC is an 18-bit value. Its high part, NCH[17:16] is hard wired to 0b01. Therefore the range of NC is 10000_H…$1FFFF_H$. The value configured by NCL is the initial value for TURNA.NAV[15:0]. DC is set to 1000_H by hardware reset and it may not be written to 0000_H.

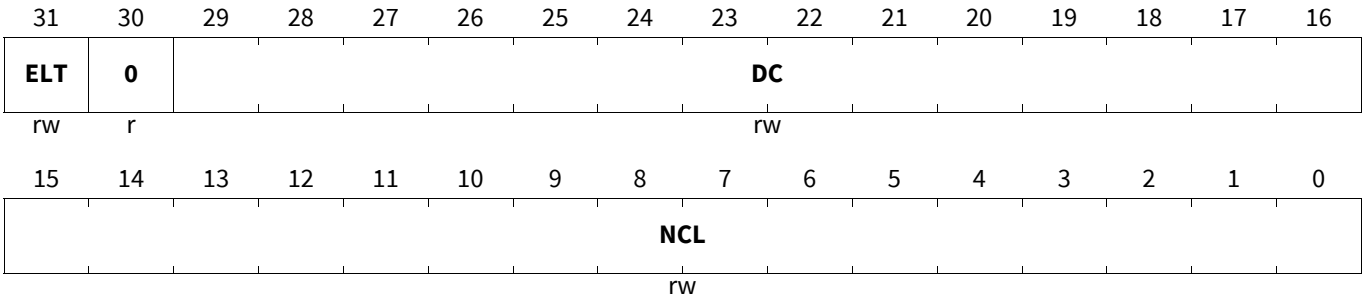
Level 1: NC ≥ 4 x DC and NTU = CAN bit time

Level 0,2: NC ≥ 8 x DC

The actual value of TUR may be changed by the clock drift compensation function of TTCAN Level 0 and Level 2 in order to adjust the node's local view of the NTU to the time master's view of the NTU. DC will not be changed

by the automatic drift compensation, TURNA.NAV may be adjusted around NC in the range of the Synchronisation Deviation Limit given by TTOCF.LDSDL. NC and DC should be programmed to the largest suitable values in order to allow the best computational accuracy for the drift compensation process.

TURCF0
TUR Configuration 0 **(008310_H)** **Application Reset Value: $1000\ 0000_H$**



Field	Bits	Type	Description
NCL	15:0	rw	**Numerator Configuration Low** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. Write access to the TUR Numerator Configuration Low is only possible during configuration with TURCF.ELT = ‘0’ or if TTOCF.EECS (external clock synchronization enabled) is set. When a new value for NCL is written outside TT Configuration Mode, the new value takes effect when TTOST.WECS is cleared to ‘0’. NCL is locked TTOST.WECS is ‘1’. *Note: If NC < 7 x DC in TTCAN Level 1, then it is required that subsequent time marks in the Trigger Memory must differ by at least 2 NTU.* 0000_HNumerator Configuration Low … $FFFF_H$Numerator Configuration Low
DC	29:16	rw	**Denominator Configuration** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. 0000_HIllegal value 0001_HDenominator Configuration … $3FFF_H$Denominator Configuration

Field	Bits	Type	Description
ELT	31	rw	**Enable Local Time** This bitfield is CCE and INIT protected. Writes will only have effect, if both bits are set. *Note: Local time is started by setting ELT. It remains active until ELT is reset or until the next hardware reset. TURCF.DC is locked when TURCF.ELT = '1'. If ELT is written to '0', the readable value will stay at '1' until the new value has been synchronized into the CAN clock domain. During this time write access to the other bits of the register remains locked.* 0_B Local time is stopped, default 1_B Local time is enabled
0	30	r	**Reserved** Shall read 0, shall be written with 0.

TT Operation Control 0

The TTCAN function exists only on CAN0 node 0.

TTOCN0

TT Operation Control 0 **(008314_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKC	**0**	**ESCN**	**NIG**	**TMG**	**FGP**	**GCS**	**TTIE**	**TMC**		**RTIE**	**SWS**		**SWP**	**ECS**	**SGT**
rh	r	rw	rwh	rw	rwh	rw	rw	rw		rw	rw		rw	rwh	rwh

Field	Bits	Type	Description
SGT	0	rwh	**Set Global time** Writing a '1' to SGT sets TTOST.WGDT if the node is the actual Time Master. SGT is reset after one Host clock period. The global time preset takes effect when the node transmits the next reference message with the Master_Ref_Mark modified by the preset value written to TTGTP.
ECS	1	rwh	**External Clock Synchronization** Writing a '1' to ECS sets TTOST.WECS if the node is the actual Time Master. ECS is reset after one Host clock period. The external clock synchronization takes effect at the start of the next basic cycle.
SWP	2	rw	**Stop Watch Polarity** 0_B Rising edge trigger 1_B Falling edge trigger

Field	Bits	Type	Description
SWS	4:3	rw	**Stop Watch Source** 00_B Stop Watch disabled 01_B Actual value of cycle time is copied to TTCPT.SWV 10_B Actual value of local time is copied to TTCPT.SWV 11_B Actual value of global time is copied to TTCPT.SWV
RTIE	5	rw	**Register Time Mark Interrupt Pulse Enable** Register time mark interrupts are configured by register TTTMK. A register time mark interrupt pulse with the length of one TTCAN clock period is generated when the time referenced by TTOCN.TMC (cycle, local, or global) equals TTTMK.TM, independent of the synchronization state. 0_B Register Time Mark Interrupt output disabled 1_B Register Time Mark Interrupt output enabled
TMC	7:6	rw	**Register Time Mark Compare** *Note: When changing the time mark reference (cycle, local, global time), it is recommended to first write TMC = "00", then reconfigure TTTMK, and finally set TMC to the intended time reference.* 00_B No Register Time Mark Interrupt generated 01_B Register Time Mark Interrupt if Time Mark = cycle time 10_B Register Time Mark Interrupt if Time Mark = local time 11_B Register Time Mark Interrupt if Time Mark = global time
TTIE	8	rw	**Trigger Time Mark Interrupt Pulse Enable** External time mark events are configured by trigger memory element TMEX. A trigger time mark interrupt pulse is generated when the trigger memory element becomes active, and the M_CAN is in synchronization state In_Schedule or In_Gap. 0_B Trigger Time Mark Interrupt output disabled 1_B Trigger Time Mark Interrupt output enabled
GCS	9	rw	**Gap Control Select** 0_B Gap control independent from event trigger 1_B Gap control by the event trigger
FGP	10	rwh	**Finish Gap** Set by the CPU, reset by each reference message 0_B No reference message requested 1_B Application requested start of reference message
TMG	11	rw	**Time Mark Gap** 0_B Reset by each reference message 1_B Next reference message started when Register Time Mark interrupt TTIR.RTMI is activated

Field	Bits	Type	Description
NIG	12	rwh	**Next is Gap** This bit can only be set when the M_CAN is the actual Time Master and when it is configured for external event-synchronized time-triggered operation (TTOCF.GEN = '1') 0_B No action, reset by reception of any reference message 1_B Transmit next reference message with Next_is_Gap = '1'
ESCN	13	rw	**External Synchronization Control** If enabled the M_CAN synchronizes its cycle time phase to an external event signalled by a rising edge at the event trigger. 0_B External synchronization disabled 1_B External synchronization enabled
LCKC	15	rh	**TT Operation Control Register Locked** Set by a write access to register TTOCN. Reset when the updated configuration has been synchronized into the CAN clock domain. 0_B Write access to TTOCN enabled 1_B Write access to TTOCN locked
0	14, 31:16	r	**Reserved** Shall read 0, shall be written with 0.

TT Global Time Preset 0

If TTOST.WGDT is set, the next reference message will be transmitted with the Master_Ref_Mark modified by the preset value and with Disc_Bit = '1', presetting the global time in all nodes simultaneously.

TP is reset to 0x0000 each time a reference message with Disc_Bit = '1' becomes valid or if the node is not the current Time Master. TP is locked while TTOST.WGTD = '1' after setting TTOCN.SGT until the reference message with Disc_Bit = '1' becomes valid or until the node is no longer the current Time Master.

The TTCAN function exists only on CAN0 node 0.

TTGTP0

TT Global Time Preset 0 **(008318_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTP															
rwh															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP															
rwh															

Field	Bits	Type	Description
TP	15:0	rwh	**Time Preset** CTP is write-protected while TTOCN.ESCN or TTOST.SPL are set. 0000_H Next Master Reference Mark = Master Reference Mark + TP … $7FFF_H$ Next Master Reference Mark = Master Reference Mark + TP 8000_H Reserved 8001_H Next Master Reference Mark = Master Reference Mark - (0x10000 - TP) … $FFFF_H$ Next Master Reference Mark = Master Reference Mark - (0x10000 - TP)
CTP	31:16	rwh	**Cycle Time Target Phase** CTP is write-protected while TTOCN.ESCN or TTOST.SPL are set. 0000_H Defines target value of cycle time when a rising edge of the event trigger is expected … $FFFF_H$ Defines target value of cycle time when a rising edge of the event trigger is expected

TT Time Mark 0

A time mark interrupt (TTIR.RTMI = ‘1’) is generated when the time base indicated by TTOCN.TMC (cycle time, local time, or global time) has the same value as TM.

The TTCAN function exists only on CAN0 node 0.

TTTMK0

TT Time Mark 0 **($00831C_H$)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCKM				**0**					**TICC**						
rh				r					rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM															
rw															

Field	Bits	Type	Description
TM	15:0	rw	**Time Mark** *Note: When using byte access to register TTTMK it is recommended to first disable the time mark compare function (TTOCN.TMC = “00”) to avoid compares on inconsistent register values.* 0000_H Time Mark … $FFFF_H$ Time Mark

Field	Bits	Type	Description
TICC	22:16	rw	**Time Mark Cycle Code** Cycle count for which the time mark is valid. 00_H valid for all cycles 01_H valid for all cycles 02_H valid every second cycle at cycle count mod2 = 0 03_H valid every second cycle at cycle count mod2 = 1 04_H valid every fourth cycle at cycle count mod4 = 0 … 07_H valid every fourth cycle at cycle count mod4 = 3 08_H valid every eighth cycle at cycle count mod8 = 0 … $0F_H$ valid every eighth cycle at cycle count mod8 = 7 10_H valid every sixteenth cycle at cycle count mod16 = 0 … $1F_H$ valid every sixteenth cycle at cycle count mod16 = 15 20_H valid every thirty-second cycle at cycle count mod32 = 0 … $3F_H$ valid every thirty-second cycle at cycle count mod32 = 31 40_H valid every sixty-fourth cycle at cycle count mod64 = 0 … $7F_H$ valid every sixty-fourth cycle at cycle count mod64 = 63
LCKM	31	rh	**TT Time Mark Register Locked** Always set by a write access to registers TTOCN. Set by write access to register TTTMK when TTOCN.TMC ≠ "00”. Reset when the registers have been synchronized into the CAN clock domain. 0_B Write access to TTTMK enabled 1_B Write access to TTTMK locked
0	30:23	r	**Reserved** Shall read 0, shall be written with 0.

TT Interrupt Register 0

The interrupt register for TTCAN related events.

Note: LDMST or SWAPMSK.W instructions should be used only with bit mask enabled for all rwh bits in this register.

The TTCAN function exists only on CAN0 node 0.

TTIR0

TT Interrupt Register 0 **(008320$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0													CER	AW	WT
r													rwh	rwh	rwh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWT	ELC	SE2	SE1	TXO	TXU	GTE	GTD	GTW	SWE	TTMI	RTMI	SOG	CSM	SMC	SBC
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
SBC	0	rwh	**Start of Basic Cycle** 0_B No Basic Cycle started since bit has been reset 1_B Basic Cycle started
SMC	1	rwh	**Start of Matrix Cycle** 0_B No Matrix Cycle started since bit has been reset 1_B Matrix Cycle started
CSM	2	rwh	**Change of Synchronization Mode** 0_B No change in master to slave relation or schedule synchronization 1_B Master to slave relation or schedule synchronization changed, also set when TTOST.SPL is reset
SOG	3	rwh	**Start of Gap** 0_B No reference message seen with Next_is_Gap bit set 1_B Reference message with Next_is_Gap bit set becomes valid
RTMI	4	rwh	**Register Time Mark Interrupt** Set when time referenced by TTOCN.TMC (cycle, local, or global) equals TTTMK.TM, independent of the synchronization state. 0_B Time mark not reached 1_B Time mark reached
TTMI	5	rwh	**Trigger Time Mark Event Internal** Internal time mark events are configured by trigger memory element TMIN. Set when the trigger memory element becomes active, and the M_CAN is in synchronization state In_Gap or In_Schedule. 0_B Time mark not reached 1_B Time mark reached (Level 0: cycle time TTOCF.IRTO · 0x200)
SWE	6	rwh	**Stop Watch Polarity** 0_B No rising/falling edge at stop watch trigger detected 1_B Rising/falling edge at stop watch trigger detected

Field	Bits	Type	Description
GTW	7	rwh	**Global Time Wrap** 0_B No global time wrap occurred 1_B Global time wrap from 0xFFFF to 0x0000 occurred
GTD	8	rwh	**Global Time Discontinuity** 0_B No discontinuity of global time 1_B Discontinuity of global time
GTE	9	rwh	**Global Time Error** Synchronization deviation SD exceeds limit specified by TTOCF.LDSDL, TTCAN Level 0,2 only. 0_B Synchronization deviation within limit 1_B Synchronization deviation exceeded limit
TXU	10	rwh	**Tx Count Underflow** 0_B Number of Tx Trigger as expected 1_B Less Tx trigger than expected in one matrix cycle
TXO	11	rwh	**Tx Count Overflow** 0_B Number of Tx Trigger as expected 1_B More Tx trigger than expected in one matrix cycle
SE1	12	rwh	**Scheduling Error 1** 0_B No scheduling error 1 1_B Scheduling error 1 occurred
SE2	13	rwh	**Scheduling Error 2** 0_B No scheduling error 2 1_B Scheduling error 2 occurred
ELC	14	rwh	**Error Level Changed** Not set when error level changed during initialization. 0_B No change in error level 1_B Error level changed
IWT	15	rwh	**Initialization Watch Trigger** The initialization is restarted by resetting IWT. 0_B No missing reference message during system startup 1_B No system startup due to missing reference message
WT	16	rwh	**Watch Trigger** 0_B No missing reference message 1_B Missing reference message (Level 0: cycle time 0xFF00)
AW	17	rwh	**Application Watchdog** 0_B Application watchdog served in time 1_B Application watchdog not served in time

Field	Bits	Type	Description
CER	18	rwh	**Configuration Error** Trigger out of order. 0_B No error found in trigger list 1_B Error found in trigger list
0	31:19	r	**Reserved** Shall read 0, shall be written with 0.

TT Interrupt Enable 0

The settings in the TT Interrupt Enable register determine which status changes in the TT Interrupt Register will result in an interrupt.

R = Read, W = Write; -n = value after reset

TTIE0
TT Interrupt Enable 0 **(008324_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0													CERE	AWE	WTE
r													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWTE	ELCE	SE2E	SE1E	TXOE	TXUE	GTEE	GTDE	GTWE	SWEE	TTMIE	RTMIE	SOGE	CSME	SMCE	SBCE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
SBCE	0	rw	**Start of Basic Cycle Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
SMCE	1	rw	**Start of Matrix Cycle Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
CSME	2	rw	**Change of Synchronization Mode Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
SOGE	3	rw	**Start of Gap Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
RTMIE	4	rw	**Register Time Mark Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled

Field	Bits	Type	Description
TTMIE	5	rw	**Trigger Time Mark Event Internal Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
SWEE	6	rw	**Stop Watch Polarity Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
GTWE	7	rw	**Global Time Wrap Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
GTDE	8	rw	**Global Time Discontinuity Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
GTEE	9	rw	**Global Time Error Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
TXUE	10	rw	**Tx Count Underflow Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
TXOE	11	rw	**Tx Count Overflow Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
SE1E	12	rw	**Scheduling Error 1 Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
SE2E	13	rw	**Scheduling Error 2 Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
ELCE	14	rw	**Change Error Level Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
IWTE	15	rw	**Initialization Watch Trigger Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled

Field	Bits	Type	Description
WTE	16	rw	**Watch Trigger Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
AWE	17	rw	**Application Watchdog Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
CERE	18	rw	**Configuration Error Interrupt Enable** 0_B TT interrupt disabled 1_B TT interrupt enabled
0	31:19	r	**Reserved** Shall read 0, shall be written with 0.

TT Operation Status 0

R = Read, P = Protected write; -n = value after reset

The TTCAN function exists only on CAN0 node 0.

TTOST0

TT Operation Status 0 **($00832C_H$)** **Application Reset Value: $2000\ 0080_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPL	**WECS**	**AWE**	**WFE**	**GSI**	**TMP**			**GFI**	**WGTD**	**0**					
rh	rh	rh	rh	rh	rh			rh	rh	r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO								**QCS**	**QGTP**	**SYS**		**MS**		**EL**	
rh								rh	rh	rh		rh		rh	

Field	Bits	Type	Description
EL	1:0	rh	**Error Level** 00_B Severity 0 - No Error 01_B Severity 1 - Warning 10_B Severity 2 - Error 11_B Severity 3 - Severe Error
MS	3:2	rh	**Master State** 00_B Master_Off, no master properties relevant 01_B Operating as Time Slave 10_B Operating as Backup Time Master 11_B Operating as current Time Master

Field	Bits	Type	Description
SYS	5:4	rh	**Synchronization State** 00_B Out of Synchronization 01_B Synchronizing to TTCAN communication 10_B Schedule suspended by Gap (In_Gap) 11_B Synchronized to schedule (In_Schedule)
QGTP	6	rh	**Quality of Global Time Phase** Only relevant in TTCAN Level 0 and Level 2, otherwise fixed to '0'. 0_B Global time not valid 1_B Global time in phase with Time Master
QCS	7	rh	**Quality of Clock Speed** Only relevant in TTCAN Level 0 and Level 2, otherwise fixed to '1'. 0_B Local clock speed not synchronized to Time Master clock speed 1_B Synchronization Deviation ≤ SDL
RTO	15:8	rh	**Reference Trigger Offset** The Reference Trigger Offset value is a signed integer with a range from -127 (0x81) to 127 (0x7F). There is no notification when the lower limit of -127 is reached. In case the M_CAN becomes Time Master (MS[1:0] = "11"), the reset of RTO is delayed due to synchronization between Host and CAN clock domain. For time slaves the value configured by TTOCF.IRTO is read. Actual Reference Trigger offset value
WGTD	22	rh	**Wait for Global Time Discontinuity** 0_B No global time preset pending 1_B Node waits for the global time preset to take effect. The bit is reset when the node has transmitted a reference message with Disc_Bit = '1' or after it received a reference message.
GFI	23	rh	**Gap Finished Indicator** Set when the CPU writes TTOCN.FGP, or by a time mark interrupt if TMG = '1', or via event trigger input if TTOCN.GCS = '1'. Not set by Ref_Trigger_Gap or when Gap is finished by another node sending a reference message. 0_B Reset at the end of each reference message 1_B Gap finished by M_CAN
TMP	26:24	rh	**Time Master Priority** 000_B Priority of actual Time Master … 111_B Priority of actual Time Master
GSI	27	rh	**Gap Started Indicator** 0_B No Gap in schedule, reset by each reference message and for all time slaves 1_B Gap time after Basic Cycle has started
WFE	28	rh	**Wait for Event** 0_B No Gap announced, reset by a reference message with Next_is_Gap = '0' 1_B Reference message with Next_is_Gap = '1' received

Field	Bits	Type	Description
AWE	29	rh	**Application Watchdog Event** The application watchdog is served by reading TTOST. When the watchdog is not served in time, bit AWE is set, all TTCAN communication is stopped, and the M_CAN is set into Bus Monitoring Mode. 0_B Application Watchdog served in time 1_B Failed to serve Application Watchdog in time
WECS	30	rh	**Wait for External Clock Synchronization** 0_B No external clock synchronization pending 1_B Node waits for external clock synchronization to take effect. The bit is reset at the start of the next basic cycle.
SPL	31	rh	**Schedule Phase Lock** The bit is valid only when external synchronization is enabled (TTOCN.ESCN = '1'). In this case it signals that the difference between cycle time configured by TTGTP.CTP and the cycle time at the rising edge at the event trigger is less or equal 9 NTU. 0_B Phase outside range 1_B Phase inside range
0	21:16	r	**Reserved** **Shall read 0, shall be written with 0.**

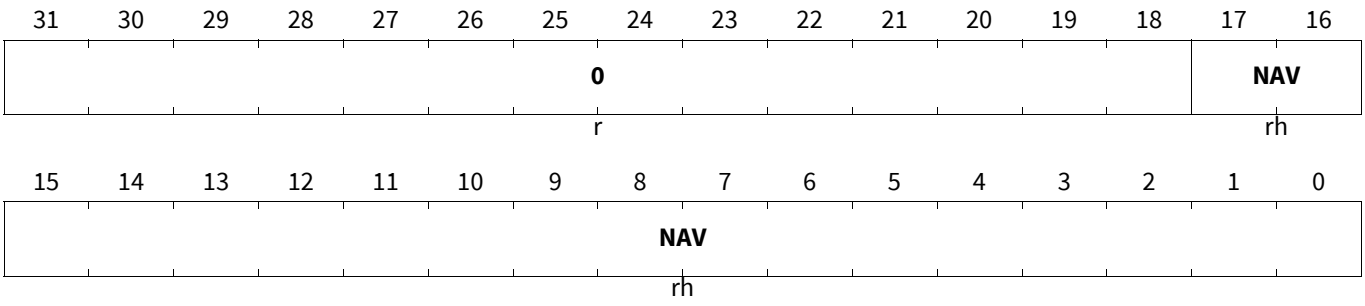
TUR Numerator Actual 0

The TTCAN function exists only on CAN0 node 0.

There is no drift compensation in TTCAN Level 1 (NAV = NC). In TTCAN Level 0 and Level 2, the drift between the node's local clock and the time master's local clock is calculated. The drift is compensated when the Synchronisation Deviation (difference between NC and the calculated NAV) is not more than $2^{(TTOCF.LDSDL + 5)}$. With TTOCF.LDSDL ≤ 7, this results in a maximum range for NAV of (NC - 0x1000) ≤ NAV ≤ (NC + 0x1000).

TURNA0

TUR Numerator Actual 0 **(008330_H)** **Application Reset Value: $0001\ 0000_H$**



Field	Bits	Type	Description
NAV	17:0	rh	**Numerator Actual Value** $0F000_H$ Actual numerator value … $20FFF_H$ Actual numerator value **others**, Illegal value

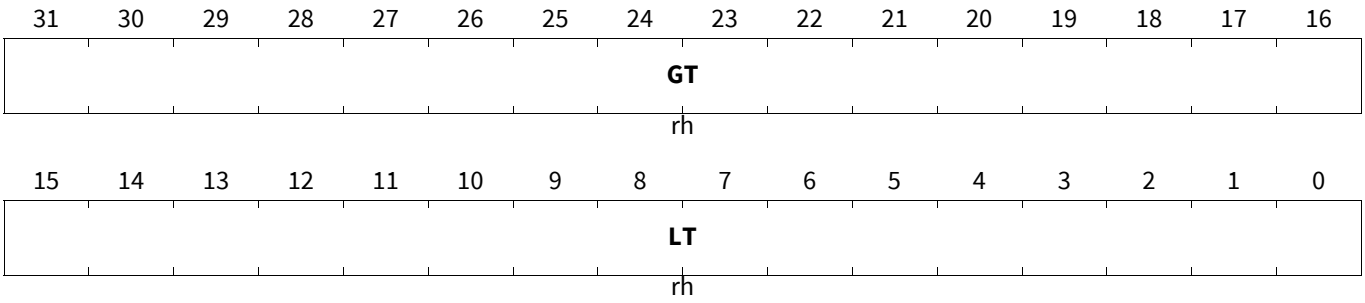
Field	Bits	Type	Description
0	31:18	r	**Reserved** Shall read 0, shall be written with 0.

TT Local & Global Time 0

The TTCAN function exists only on CAN0 node 0.

TTLGT0

TT Local & Global Time 0 **(008334$_H$)** **Application Reset Value: 0000 0000$_H$**



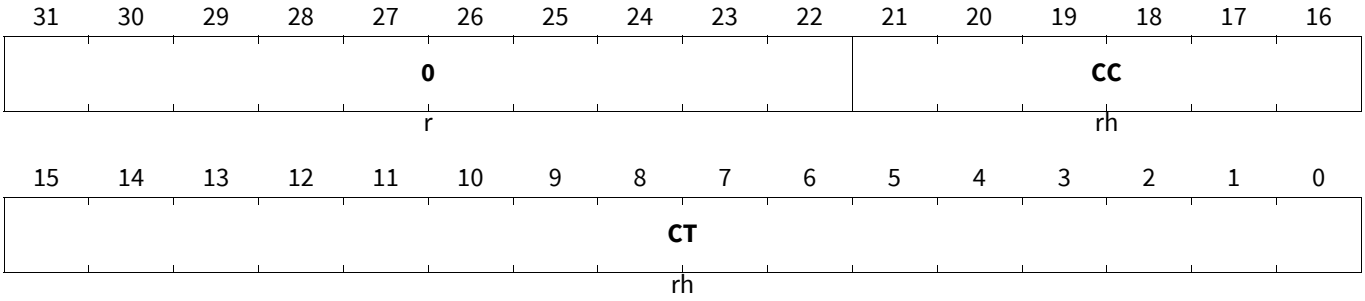
Field	Bits	Type	Description
LT	15:0	rh	**Local Time** Non-fractional part of local time, incremented once each local NTU. Local time value of TTCAN node
GT	31:16	rh	**Global Time** Non-fractional part of the sum of the node's local time and its local offset. Global time value of TTCAN network

TT Cycle Time & Count 0

The TTCAN function exists only on CAN0 node 0.

TTCTC0

TT Cycle Time & Count 0 **(008338$_H$)** **Application Reset Value: 003F 0000$_H$**



Field	Bits	Type	Description
CT	15:0	rh	**Cycle Time** Non-fractional part of the difference of the node's local time and Ref_Mark. Cycle time value of TTCAN Basic Cycle

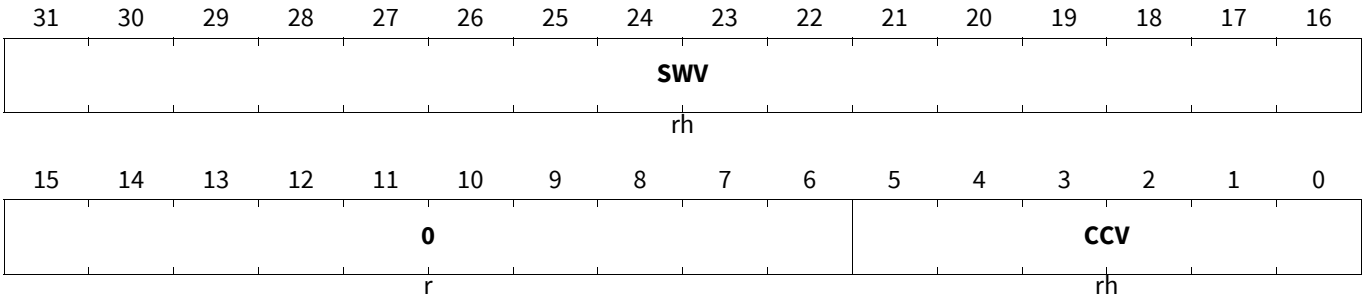
Field	Bits	Type	Description
CC	21:16	rh	**Cycle Count** Number of actual Basic Cycle in the System Matrix
0	31:22	r	**Reserved** Shall read 0, shall be written with 0.

TT Capture Time 0

The TTCAN function exists only on CAN0 node 0.

TTCPT0

TT Capture Time 0 **(00833C$_H$)** **Application Reset Value: 0000 0000$_H$**



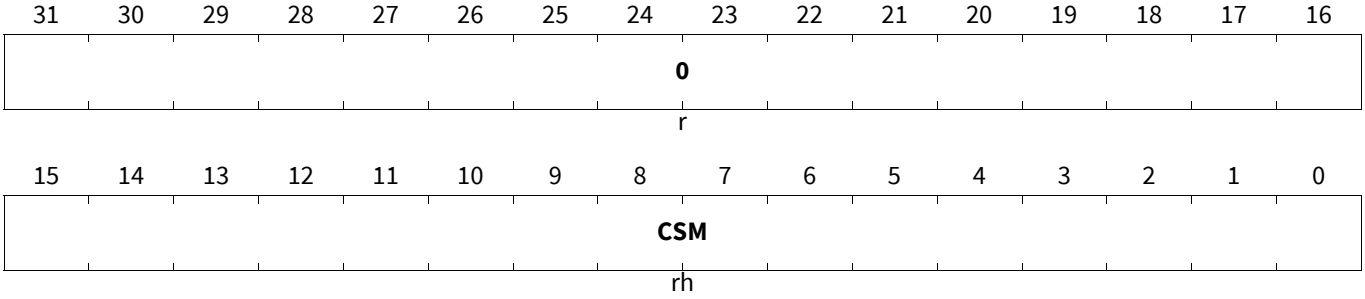
Field	Bits	Type	Description
CCV	5:0	rh	**Cycle Count Value** Cycle count value captured together with SWV. Captured cycle count value
SWV	31:16	rh	**Stop Watch Value** On a rising/falling edge (as configured via TTOCN.SWP) at the Stop Watch Trigger, when TTOCN.SWS is ≠ “00” and TTIR.SWE is ‘0’, the actual time value as selected by TTOCN.SWS (cycle, local, global) is copied to SWV and TTIR.SWE will be set to ‘1’. Capturing of the next stop watch value is enabled by resetting TTIR.SWE. Captured Stop Watch value
0	15:6	r	**Reserved** Shall read 0, shall be written with 0.

TT Cycle Sync Mark 0

The TTCAN function exists only on CAN0 node 0.

TTCSM0

TT Cycle Sync Mark 0 **(008340_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
CSM	15:0	rh	**Cycle Sync Mark** The Cycle Sync Mark is measured in cycle time. It is updated when the reference message becomes valid and retains its value until the next reference message becomes valid. Captured cycle time
0	31:16	r	**Reserved** Shall read 0, shall be written with 0.

40.4.6 Message RAM

M_CAN

40.4.6.1 Message RAM Configuration

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via RXESCi.F0DS, RXESCi.F1DS, RXESCi.RBDS, and TXESCi.TBDS.

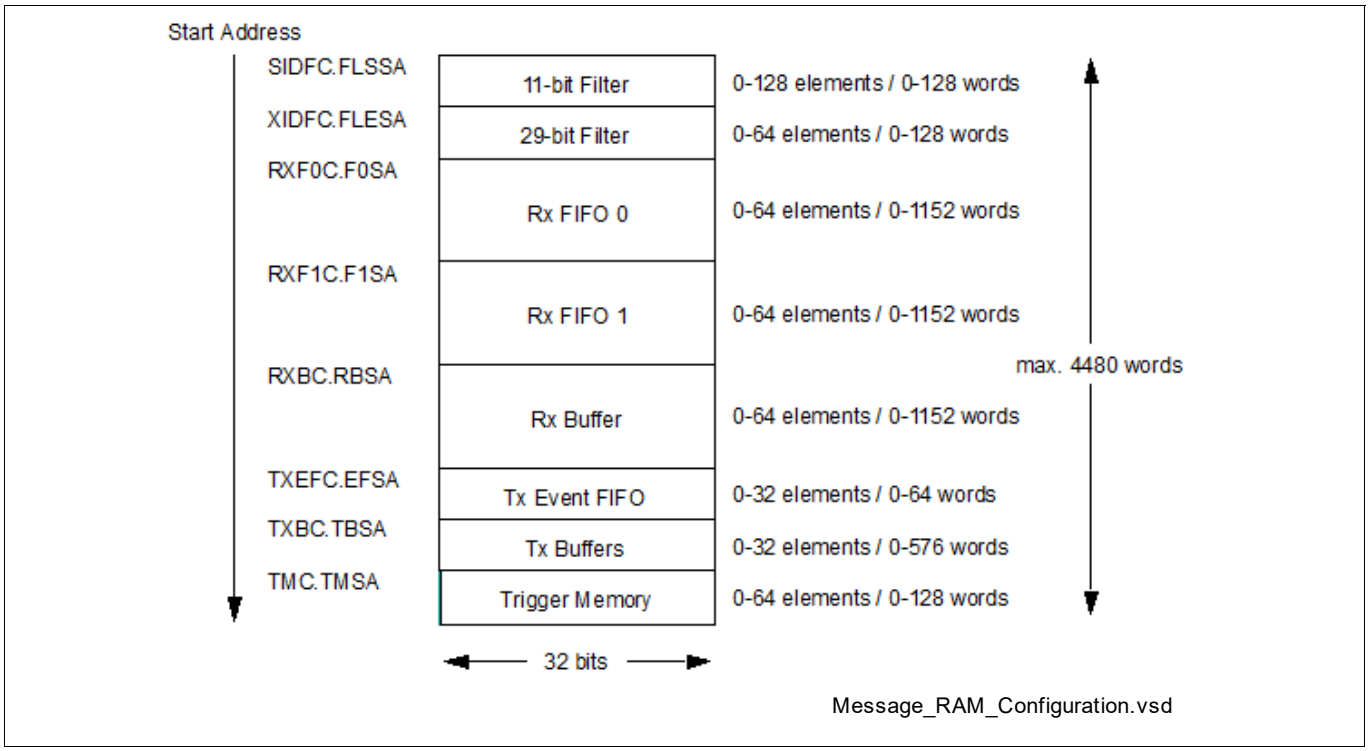


Figure 605 Message RAM Configuration

When the M_CAN addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

40.4.6.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in **Table 382** below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESCi.

Table 382 Message Layout - Rx Buffer and FIFO Element

	3 1	2 4	2 3	1 6	1 5	8	7	0
R0	ESI	XTD	RTR	ID[28:0]				
R1	ANMF	FIDX[6:0]		0	FDF	BRS	DLC[3:0]	RXTS[15:0]
R2	DB3[7:0]		DB2[7:0]		DB1[7:0]		DB0[7:0]	
R3	DB7[7:0]		DB6[7:0]		DB5[7:0]		DB4[7:0]	
…	…		…		…		…	
Rn	DBm[7:0]		DBm-1[7:0]		DBm-2[7:0]		DBm-3[7:0]	

Register 0

See Message layout, **Table 382**.

See Message layout.

RxMsgk_R0 (k=0-63)

Register 0 **($000000_H + k*48_H$)** **Application Reset Value: XXXX XXXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESI	**XTD**	**RTR**	**ID**												
rwh	rwh	rwh	rwh												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID															
rwh															

Field	Bits	Type	Description
ID	28:0	rwh	**Identifier** Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

Field	Bits	Type	Description
RTR	29	rwh	**Remote Transmission Request** Signals to the Host whether the received frame is a data frame or a remote frame. *Note: There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1'), bit RTR reflects the state of the reserved bit r1.* 0_B Received frame is a data frame 1_B Received frame is a remote frame
XTD	30	rwh	**Extended Identifier** Signals to the Host whether the received frame has a standard or extended identifier. 0_B 11-bit standard identifier 1_B 29-bit extended identifier
ESI	31	rwh	**Error State Indicator** 0_B Transmitting node is error active 1_B Transmitting node is error passive

Register 1

See Message layout, **Table 382**.

See Message layout.

RxMsgk_R1 (k=0-63)

Register 1 **(000004_H+k*48_H)** **Application Reset Value: XXXX $XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ANMF	**FIDX**							**0**		**FDF**	**BRS**	**DLC**			
rwh	rwh							rwh		rwh	rwh	rwh			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXTS															
rwh															

Field	Bits	Type	Description
RXTS	15:0	rwh	**Rx Timestamp** Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.

Field	Bits	Type	Description
DLC	19:16	rwh	**Data Length Code** 0_H CAN + CAN FD: received frame has 0 data bytes … 8_H CAN + CAN FD: received frame has 8 data bytes 9_H CAN FD: received frame has 12 (9*4-24) data bytes CAN: received frame has 8 data bytes … C_H CAN FD: received frame has 24 (12*4-24) data bytes CAN: received frame has 8 data bytes D_H CAN FD: received frame has 32 (13*16-176) data bytes CAN: received frame has 8 data bytes … F_H CAN FD: received frame has 64 (15*16-176) data bytes CAN: received frame has 8 data bytes
BRS	20	rwh	**Bit Rate Switch** 0_B Frame received without bit rate switching 1_B Frame received with bit rate switching
FDF	21	rwh	**Frame Data Format** 0_B Standard frame format 1_B CAN FD frame format (new DLC-coding and CRC)
FIDX	30:24	rwh	**Filter Index** 00_H Index of matching Rx acceptance filter element (invalid if ANMF = '1'). Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1. … $7F_H$ Index of matching Rx acceptance filter element (invalid if ANMF = '1'). Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1.
ANMF	31	rwh	**Accepted Non-matching Frame** Acceptance of non-matching frames may be enabled via GFC.ANFS and GFC.ANFE. 0_B Received frame matching filter index FIDX 1_B Received frame did not match any Rx filter element
0	23:22	rwh	**Reserved** Shall read 0, shall be written with 0.

Data Byte m

RxMsgk_DBm (k=0-63;m=0-63)

Data Byte m **(000008_H+k*48_H+m)** **Application Reset Value: XX_H**

7	6	5	4	3	2	1	0
DB							
rwh							

Field	Bits	Type	Description
DB	7:0	rwh	**Data Byte m**

Note: Depending on the configuration of the element size (RXESC), between two and sixteen 32-bit words (Rn = 3…17) are used for storage of a CAN message's data field.

40.4.6.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBCi.TFQS and TXBCi.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESCi.

Table 383 Message Layout - Tx Buffer Element

	31…24	23…16	15…8	7…0
T0	0 | XTD | RTR | ID[28:0]			
T1	MM[7:0]	EFC | 0 | FDF | BRS | DLC[3:0]	0	
T2	DB3[7:0]	DB2[7:0]	DB1[7:0]	DB0[7:0]
T3	DB7[7:0]	DB6[7:0]	DB5[7:0]	DB4[7:0]
…	…	…	…	…
Tn	DBm[7:0]	DBm-1[7:0]	DBm-2[7:0]	DBm-3[7:0]

Transmit Buffer 0

See Message layout.

TxMsgk_T0 (k=0-31)
Transmit Buffer 0 **($000000_H + k*48_H$)** **Application Reset Value: XXXX XXXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESI	**XTD**	**RTR**	**ID**												
rw	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID															
rw															

Field	Bits	Type	Description
ID	28:0	rw	**Identifier** Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

Field	Bits	Type	Description
RTR	29	rw	**Remote Transmission Request** *Note: When RTR = 1, the M_CAN transmits a remote frame according to ISO11898-1, even if CCCR.FDOE enables the transmission in CAN FD format.* 0_B Transmit data frame 1_B Transmit remote frame
XTD	30	rw	**Extended Identifier** 0_B 11-bit standard identifier 1_B 29-bit extended identifier
ESI	31	rw	**Error State Indicator** *Note: The ESI bit of the transmit buffer is OR'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.* 0_B ESI bit in CAN FD format depends only on error passive flag 1_B ESI bit in CAN FD format transmitted recessive

Transmit Buffer 1

See Message layout.

TxMsgk_T1 (k=0-31)

Transmit Buffer 1 **(000004_H+k*48_H)** **Application Reset Value: XXXX $XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MM								EFC	0	FDF	BRS	DLC			
rw								rw	rw	rw	rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															
rw															

Field	Bits	Type	Description
DLC	19:16	rw	**Data Length Code** 0_H CAN + CAN FD: received frame has 0 data bytes … 8_H CAN + CAN FD: received frame has 8 data bytes 9_H CAN FD: received frame has 12 (9*4-24) data bytes CAN: received frame has 8 data bytes … C_H CAN FD: received frame has 24 (12*4-24) data bytes CAN: received frame has 8 data bytes D_H CAN FD: received frame has 32 (13*16-176) data bytes CAN: received frame has 8 data bytes … F_H CAN FD: received frame has 64 (15*16-176) data bytes CAN: received frame has 8 data bytes
BRS	20	rw	**Bit Rate Switching** *Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR.FDOE = 1'. Bit BRS is only evaluated when in addition CCCR.BRSE = '1'.* 0_B CAN FD frames transmitted without bit rate switching 1_B CAN FD frames transmitted with bit rate switching
FDF	21	rw	**FD Format** 0_B Frame transmitted in Classical CAN format 1_B Frame transmitted in CAN FD format
EFC	23	rw	**Event FIFO Control** 0_B Don't store Tx events 1_B Store Tx events
MM	31:24	rw	**Message Marker** Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.
0	15:0, 22	rw	**Reserved** Shall read 0, shall be written with 0.

Data Byte m

TxMsgk_DBm (k=0-31;m=0-63)

Data Byte m **(000008_H+k*48_H+m)** **Application Reset Value: XX_H**

7	6	5	4	3	2	1	0
DB							
rw							

Field	Bits	Type	Description
DB	7:0	rw	**Data Byte m**

Note: Depending on the configuration of the element size (TXESC), between two and sixteen 32-bit words (Tn = 3…17) are used for storage of a CAN message's data field.

40.4.6.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 384 Message Layout - Tx Event FIFO Element

	31…24	23…16	15…8	7…0
E0	ESI, XTD, RTR, ID[28:0]			
E1	MM[7:0]	ET[1:0], FDF, BRS, DLC[3:0]	TXTS[15:0]	

Event 0

See Message layout.

TxEventk_E0 (k=0-31)

Event 0 **(000000_H + k*8)** **Application Reset Value: XXXX XXXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESI	XTD	RTR	ID												
rwh	rwh	rwh	rwh												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID															
rwh															

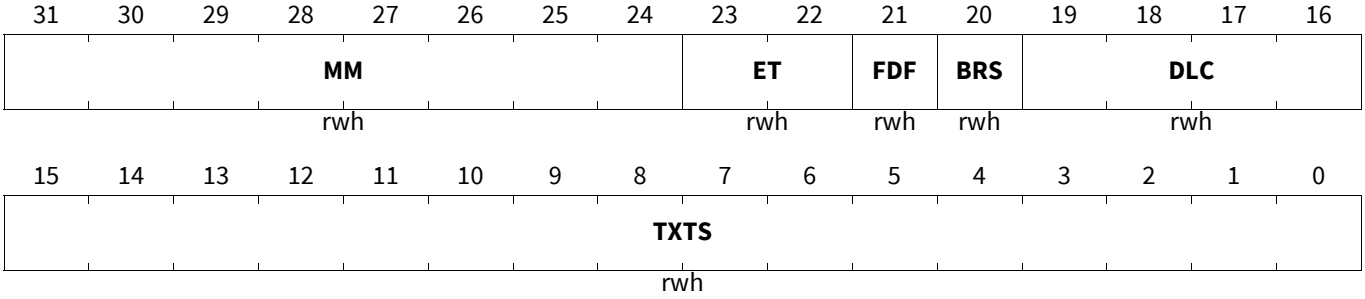
Field	Bits	Type	Description
ID	28:0	rwh	**Identifier** Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
RTR	29	rwh	**Remote Transmission Request** 0_B Data frame transmitted 1_B Remote frame transmitted
XTD	30	rwh	**Extended Identifier** 0_B 11-bit standard identifier 1_B 29-bit extended identifier
ESI	31	rwh	**Error State Indicator** 0_B Transmitting node is error active 1_B Transmitting node is error passive

Event 1

See Message layout.

TxEventk_E1 (k=0-31)

Event 1 **(000004_H+k*8)** **Application Reset Value: XXXX $XXXX_H$**



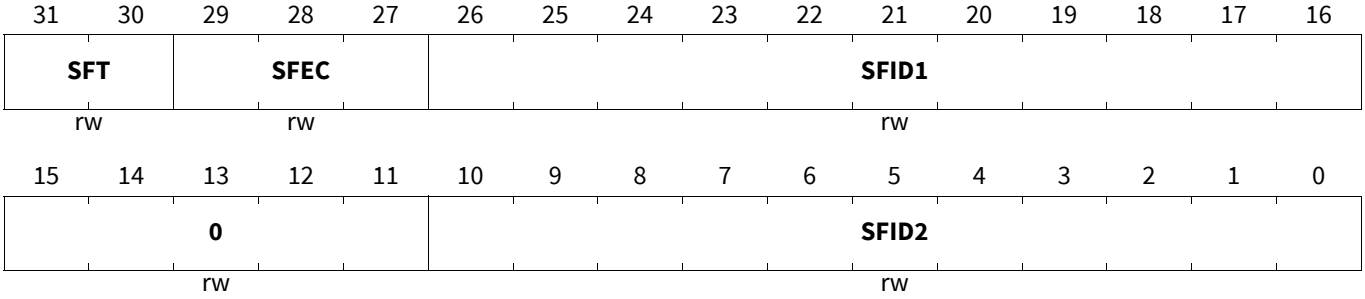
Field	Bits	Type	Description
TXTS	15:0	rwh	**Tx Timestamp** Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.
DLC	19:16	rwh	**Data Length Code** 0_H CAN + CAN FD: received frame has 0 data bytes … 8_H CAN + CAN FD: received frame has 8 data bytes 9_H CAN FD: received frame has 12 (9*4-24) data bytes CAN: received frame has 8 data bytes … C_H CAN FD: received frame has 24 (12*4-24) data bytes CAN: received frame has 8 data bytes D_H CAN FD: received frame has 32 (13*16-176) data bytes CAN: received frame has 8 data bytes … F_H CAN FD: received frame has 64 (15*16-176) data bytes CAN: received frame has 8 data bytes
BRS	20	rwh	**Bit Rate Switch** 0_B Frame transmitted without bit rate switching 1_B Frame transmitted with bit rate switching
FDF	21	rwh	**FD Format** 0_B Standard frame format 1_B CAN FD frame format (new DLC-coding and CRC)
ET	23:22	rwh	**Event Type** 00_B Reserved 01_B Tx event 10_B Transmission in spite of cancellation (always set for transmissions in DAR mode) 11_B Reserved
MM	31:24	rwh	**Message Marker** Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

40.4.6.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address SIDFCi.FLSSA plus the index of the filter element (0…127).

Standard Message 0

StdMsgk_S0 (k=0-127)
Standard Message 0 **(000000_H + k*4)** **Application Reset Value: XXXX $XXXX_H$**



Field	Bits	Type	Description
SFID2	10:0	rw	**Standard Filter ID 2** This bit field has a different meaning depending on the configuration of SFEC: 1) SFEC = “001”…”110” Second ID of standard ID filter element 2) SFEC = “111” Filter for Rx Buffers or for debug messages **SFID2[5:0]** defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message. **SFID2[8:6]** is used to control the filter event pins. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one host clock period in case the filter matches. **SFID2[10:9]** decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. 000_H Store message into an Rx Buffer … $1FF_H$ Store message into an Rx Buffer 200_H Debug Message A … $3FF_H$ Debug Message A 400_H Debug Message B … $5FF_H$ Debug Message B 600_H Debug Message C … $7FF_H$ Debug Message C

Field	Bits	Type	Description
SFID1	26:16	rw	**Standard Filter ID 1** First ID of standard ID filter element. When filtering for Rx Buffers or for debug messages this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.
SFEC	29:27	rw	**Standard Filter Element Configuration** All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101” or “110”, a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match. 000_B Disable filter element 001_B Store in Rx FIFO 0 if filter matches 010_B Store in Rx FIFO 1 if filter matches 011_B Reject ID if filter matches 100_B Set priority if filter matches 101_B Set priority and store in FIFO 0 if filter matches 110_B Set priority and store in FIFO 1 if filter matches 111_B Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored
SFT	31:30	rw	**Standard Filter Type** *Note: With SFT = “11” the filter element is disabled and the acceptance filtering continues (same behaviour as with SFEC = “000”)* 00_B Range filter from SF1ID to SF2ID (SF2ID ≥ SF1ID) 01_B Dual ID filter for SF1ID or SF2ID 10_B Classic filter: SF1ID = filter, SF2ID = mask 11_B Filter element disabled
0	15:11	rw	**Reserved** Shall read 0, shall be written with 0.

40.4.6.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFCi.FLESA plus two times the index of the filter element (0…63).

Table 385 Message Layout - Extended Message ID Filter Element

3 1	2 4	2 3	1 6	1 5	8	7	0

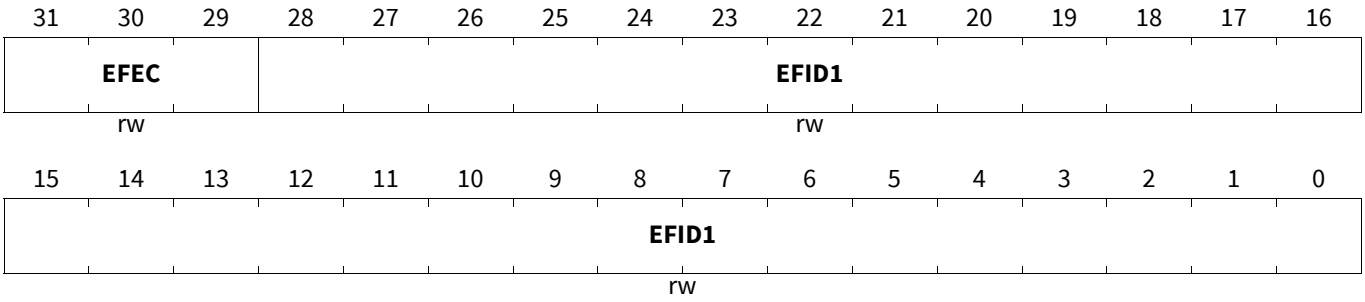
Table 385 Message Layout - Extended Message ID Filter Element (cont'd)

F0	EFEC[2:0]		EFID1[28:0]
F1	EFT[1:0]	0	EFID2[28:0]

Filter Element 0

See Message layout.

ExtMsgk_F0 (k=0-63)
Filter Element 0 **(000000_H + k*8)** **Application Reset Value: XXXX $XXXX_H$**

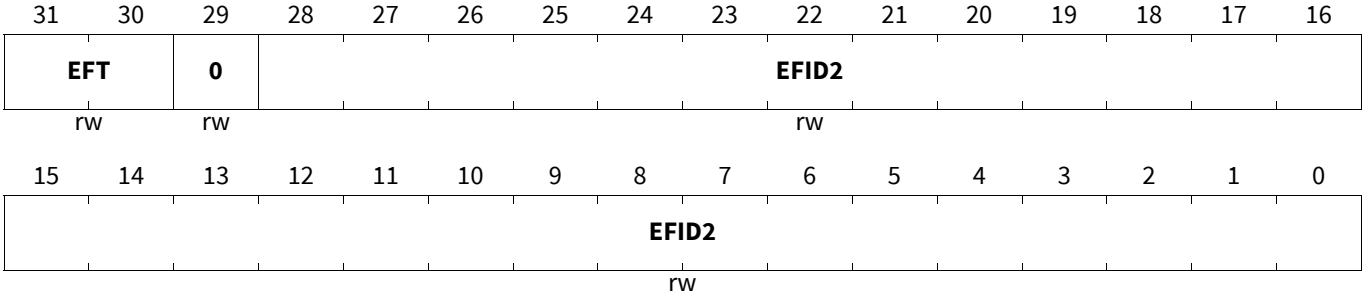


Field	Bits	Type	Description
EFID1	28:0	rw	**Extended Filter ID 1** First ID of extended ID filter element. When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAMi masking mechanism is used.
EFEC	31:29	rw	**Extended Filter Element Configuration** All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = "100", "101" or "110", a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match. 000_B Disable filter element 001_B Store in Rx FIFO 0 if filter matches 010_B Store in Rx FIFO 1 if filter matches 011_B Reject ID if filter matches 100_B Set priority if filter matches 101_B Set priority and store in FIFO 0 if filter matches 110_B Set priority and store in FIFO 1 if filter matches 111_B Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

Filter Element 1

See Message layout.

ExtMsgk_F1 (k=0-63)
Filter Element 1 **(000004_H+k*8)** **Application Reset Value: XXXX $XXXX_H$**



Field	Bits	Type	Description
EFID2	28:0	rw	**Extended Filter ID 2** This bit field has a different meaning depending on the configuration of EFEC: 1) EFEC = “001”…”110” Second ID of extended ID filter element 2) EFEC = “111” Filter for Rx Buffers or for debug messages **EFID2[5:0]** defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message. **EFID2[8:6]** is used to control the filter event. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one host clock period in case the filter matches. **EFID2[10:9]** decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. 00000000_HStore message into an Rx Buffer … $000001FF_H$Store message into an Rx Buffer 00000200_HDebug Message A … $000003FF_H$Debug Message A 00000400_HDebug Message B … $000005FF_H$Debug Message B 00000600_HDebug Message C … $000007FF_H$Debug Message C
EFT	31:30	rw	**Extended Filter Type** 00_B Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID) 01_B Dual ID filter for EF1ID or EF2ID 10_B Classic filter: EF1ID = filter, EF2ID = mask 11_B Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), XIDAM mask not applied
0	29	rw	**Reserved** Shall read 0, shall be written with 0

40.4.6.7 Trigger Memory Element

Up to 64 trigger memory elements can be configured. When accessing a Trigger Memory element, its address is the Trigger Memory Start Address TTTMC.TMSA plus the index of the trigger memory element (0…63)

Table 386 Message Layout - Trigger Memory Element

	31	16	15	8	7	0
TM0	TM[15:0]		0	CC[6:0]	0	TMIN, TMEX, TYPE[3:0]
TM1	0	FTYPE, MNR[6:0]	0			MSC[2:0]

Trigger Memory Element 0

See Message layout, see Trigger Memory Element Overview.

TrigMsgk_TM0 (k=0-63)
Trigger Memory Element 0 **(000000_H + k*8)** **Application Reset Value: XXXX $XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TM															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CC							0		TMIN	TMEX	TYPE			
rw	rw							rw		rw	rw	rw			

Field	Bits	Type	Description
TYPE	3:0	rw	**Trigger Type** *Note: No ASC implemented. If and only if implemented: For ASC operation (ASC = "10", "11") only trigger types Rx_Trigger and Time_Base_Trigger should be used.* 0_H Tx_Ref_Trigger - valid when not in Gap 1_H Tx_Ref_Trigger_Gap - valid when in Gap 2_H Tx_Trigger_Single - starts a single transmission in an exclusive time window 3_H Tx_Trigger_Continous - starts continuous transmission in an exclusive time window 4_H Tx_Trigger_Arbitration - starts a transmission in an arbitrating time window 5_H Tx_Trigger_Merged - starts a merged arbitration window 6_H Watch_Trigger - valid when not in Gap 7_H Watch_Trigger_Gap - valid when in Gap 8_H Rx_Trigger - check for reception 9_H Time_Base_Trigger - only control TMIN, TMEX, and ASC **Others**, End_of_List - illegal type, causes config error
TMEX	4	rw	**Time Mark Event External** 0_B No action 1_B When the time mark of the trigger memory element becomes active and TTOCN.TTMIE = '1'
TMIN	5	rw	**Time Mark Event Internal** 0_B No action 1_B TTIR.TTMI is set when trigger memory element becomes active

Field	Bits	Type	Description
CC	14:8	rw	**Cycle Code** Cycle count for which the trigger is valid. Ignored for trigger types Tx_Ref_Trigger, Tx_Ref_Trigger_Gap, Watch_Trigger, Watch_Trigger_Gap, End_of_List. 00_H valid for all cycles 01_H valid for all cycles 02_H valid every second cycle at cycle count mod2 = c 03_H valid every second cycle at cycle count mod2 = c 04_H valid every fourth cycle at cycle count mod4 = cc … 07_H valid every fourth cycle at cycle count mod4 = cc 08_H valid every eighth cycle at cycle count mod8 = ccc … $0F_H$ valid every eighth cycle at cycle count mod8 = ccc 10_H valid every sixteenth cycle at cycle count mod16 = cccc … $1F_H$ valid every sixteenth cycle at cycle count mod16 = cccc 20_H valid every thirty-second cycle at cycle count mod32 = ccccc … $3F_H$ valid every thirty-second cycle at cycle count mod32 = ccccc 40_H valid every sixty-fourth cycle at cycle count mod64 = cccccc … $7F_H$ valid every sixty-fourth cycle at cycle count mod64 = cccccc
TM	31:16	rw	**Time Mark** Cycle time for which the trigger becomes active.
0	7:6, 15	rw	**Reserved** Shall read 0, shall be written with 0.

See Message layout, see Trigger Memory Element Overview.

TrigMsgk_TM1 (k=0-63)

(000004_H+k*8) **PowerOn Reset Value: XXXX XXXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								FTYPE	MMR						
rw								rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														MSC	
rw														rwh	

Field	Bits	Type	Description
MSC	2:0	rwh	**Message Status Count** Counts scheduling errors for periodic messages in exclusive time windows. It has no function for arbitrating messages and in event-driven CAN communication (ISO11898-1). *Note: The trigger memory elements have to be written when the M_CAN is in INIT state. Write access to the trigger memory elements outside INIT state is not allowed. There is an exception for TMIN and TMEX when they are defined as part of a trigger memory element of TYPE Tx_Ref_Trigger. In this case they become active at the time mark modified by the actual Reference Trigger Offset (TTOST.RTO).* 000_B Actual status … 111_B Actual status
MMR	22:16	rw	**Message Number** Transmission: Trigger is valid for configured Tx Buffer number. Valid values are 0 to 31. Reception: Trigger is valid for standard / extended message ID filter element number. Valid values are 0 to 63 resp. 0 to 127.
FTYPE	23	rw	**Filter Type** 0_B 11-bit standard message ID 1_B 29-bit extended message ID
0	15:3, 31:24	rw	**Reserved** Shall read 0, shall be written with 0.