Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	CANbus, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamv70n19b-aab

16.7.4 Embedded Trace Module (ETM) Pins

The Embedded Trace Module (ETM) uses the Trace Port Interface Unit (TPIU) to export data out of the system.

The TPUI features the pins:

- TRACECLK–always exported to enable synchronization back with the data. PCK3 is used internally.
- TRACED0–3–the instruction trace stream.

16.7.5 Flash Patch Breakpoint (FPB)

The FPB implements hardware breakpoints.

16.7.6 Data Watchpoint and Trace (DWT)

The DWT contains four comparators which can be configured to generate:

- PC sampling packets at set intervals
- PC or Data watchpoint packets
- Watchpoint event to halt core

The DWT contains counters for:

- Clock cycle (CYCCNT)
- Folded instructions
- Load Store Unit (LSU) operations
- Sleep cycles
- CPI (all instruction cycles except for the first cycle)
- Interrupt overhead

16.7.7 Instrumentation Trace Macrocell (ITM)

The ITM is an application driven trace source that supports `printf` style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets which can be generated by three different sources with several priority levels:

- Software trace: Software can write directly to ITM stimulus registers. This can be done using the `printf` function. For more information, refer to 16.7.5 Flash Patch Breakpoint (FPB).
- Hardware trace: The ITM emits packets generated by the DWT.
- Timestamping: Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp.

16.7.7.1 How to Configure the ITM

The following example describes how to output trace data in asynchronous trace mode.

Configure the TPIU for asynchronous trace mode. Refer to 16.7.7.3 How to Configure the TPIU.

1. Enable the write accesses into the ITM registers by writing “0xC5ACCE55” into the Lock Access Register (Address: 0xE0000FB0)
2. Write 0x00010015 into the Trace Control register:
   - Enable ITM.
   - Enable Synchronization packets.
   - Enable SWO behavior.
   - Fix the ATB ID to 1.

Value	Name	Description
0x10	SAM E70	SAM E70
0x11	SAM S70	SAM S70
0x12	SAM V71	SAM V71
0x13	SAM V70	SAM V70

Bits 19:16 – SRAMSIZ[3:0] Internal SRAM Size

Value	Name	Description
0	48K	48 Kbytes
1	192K	192 Kbytes
2	384K	384 Kbytes
3	6K	6 Kbytes
4	24K	24 Kbytes
5	4K	4 Kbytes
6	80K	80 Kbytes
7	160K	160 Kbytes
8	8K	8 Kbytes
9	16K	16 Kbytes
10	32K	32 Kbytes
11	64K	64 Kbytes
12	128K	128 Kbytes
13	256K	256 Kbytes
14	96K	96 Kbytes
15	512K	512 Kbytes

Bits 15:12 – NVPSIZ2[3:0] Second Nonvolatile Program Memory Size

Value	Name	Description
0	NONE	None
1	8K	8 Kbytes
2	16K	16 Kbytes
3	32K	32 Kbytes
4	–	Reserved
5	64K	64 Kbytes
6	–	Reserved
7	128K	128 Kbytes
8	–	Reserved
9	256K	256 Kbytes
10	512K	512 Kbytes
11	–	Reserved
12	1024K	1024 Kbytes
13	–	Reserved
14	2048K	2048 Kbytes
15	–	Reserved

Bits 11:8 – NVPSIZ[3:0] Nonvolatile Program Memory Size

27.6.9 RTC Interrupt Enable Register

Name: RTC_IER
Offset: 0x20
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERREN Time and/or Date Error Interrupt Enable

Value	Description
0	No effect.
1	The time and date error interrupt is enabled.

Bit 4 – CALEN Calendar Event Interrupt Enable

Value	Description
0	No effect.
1	The selected calendar event interrupt is enabled.

Bit 3 – TIMEN Time Event Interrupt Enable

Value	Description
0	No effect.
1	The selected time event interrupt is enabled.

Bit 2 – SECEN Second Event Interrupt Enable

Value	Description
0	No effect.
1	The second periodic interrupt is enabled.

31.20.21 PMC Write Protection Mode Register

Name: PMC_WPMR
Offset: 0x00E4
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPEN
Access								
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504D43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See "Register Write Protection" for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

The filters also introduce some latencies, illustrated in the following two figures.

The glitch filters are controlled by the Input Filter Enable Register (PIO_IFER), the Input Filter Disable Register (PIO_IFDR) and the Input Filter Status Register (PIO_IFSR). Writing PIO_IFER and PIO_IFDR respectively sets and clears bits in PIO_IFSR. This last register enables the glitch filter on the I/O lines.

When the glitch and/or debouncing filter is enabled, it does not modify the behavior of the inputs on the peripherals. It acts only on the value read in PIO_PDSR and on the input change interrupt detection. The glitch and debouncing filters require that the peripheral clock is enabled.

Figure 32-4. Input Glitch Filter Timing

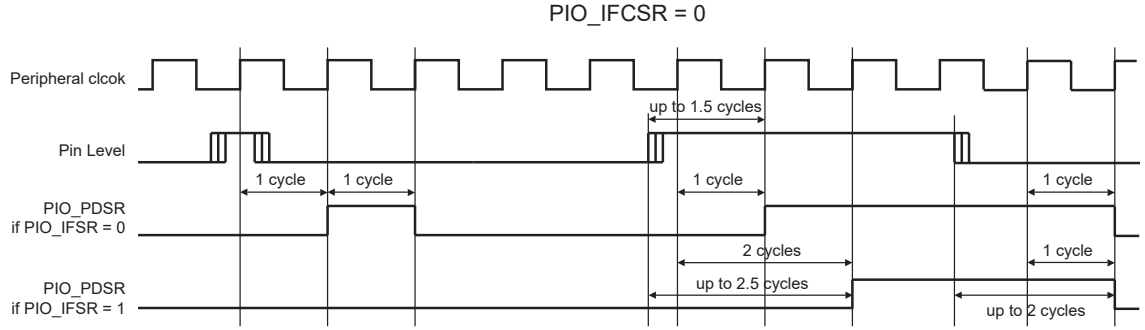
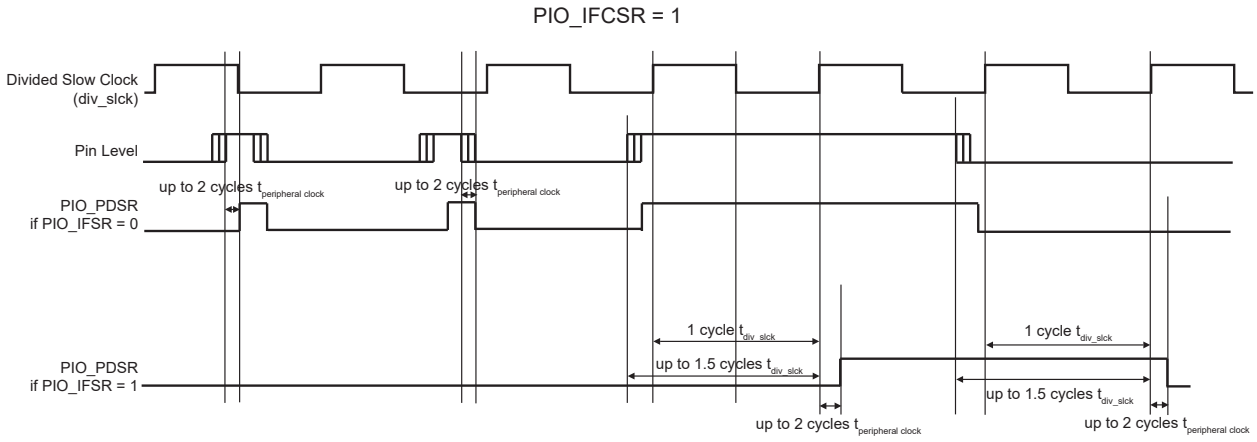


Figure 32-5. Input Debouncing Filter Timing



32.5.10 Input Edge/Level Interrupt

The PIO Controller can be programmed to generate an interrupt when it detects an edge or a level on an I/O line. The Input Edge/Level interrupt is controlled by writing the Interrupt Enable Register (PIO_IER) and the Interrupt Disable Register (PIO_IDR), which enable and disable the input change interrupt respectively by setting and clearing the corresponding bit in the Interrupt Mask Register (PIO_IMR). As input change detection is possible only by comparing two successive samplings of the input of the I/O line, the peripheral clock must be enabled. The Input Change interrupt is available regardless of the configuration of the I/O line, i.e., configured as an input only, controlled by the PIO Controller or assigned to a peripheral function.

By default, the interrupt can be generated at any time an edge is detected on the input.

Some additional interrupt modes can be enabled/disabled by writing in the Additional Interrupt Modes Enable Register (PIO_AIMER) and Additional Interrupt Modes Disable Register (PIO_AIMDR). The current state of this selection can be read through the Additional Interrupt Modes Mask Register (PIO_AIMMR).

Offset	Name	Bit Pos.								
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xC4	PIO_LSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xC8	PIO_ELSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xCC ... 0xCF	Reserved									
0xD0	PIO_FELLSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xD4	PIO_REHLSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xD8	PIO_FRLHSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xDC ... 0xDF	Reserved									
0xE0	PIO_LOCKSR	7:0	P7	P6	P5	P4	P3	P2	P1	P0
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		31:24	P31	P30	P29	P28	P27	P26	P25	P24
0xE4	PIO_WPMR	7:0								WPEN
		15:8	WPKEY[7:0]							
		23:16	WPKEY[15:8]							
		31:24	WPKEY[23:16]							
0xE8	PIO_WPSR	7:0								WPVS
		15:8	WPVSRC[7:0]							
		23:16	WPVSRC[15:8]							
		31:24								
0xEC ... 0xFF	Reserved									
0x0100	PIO_SCHMITT	7:0	SCHMITT7	SCHMITT6	SCHMITT5	SCHMITT4	SCHMITT3	SCHMITT2	SCHMITT1	SCHMITT0
		15:8	SCHMITT15	SCHMITT14	SCHMITT13	SCHMITT12	SCHMITT11	SCHMITT10	SCHMITT9	SCHMITT8
		23:16	SCHMITT23	SCHMITT22	SCHMITT21	SCHMITT20	SCHMITT19	SCHMITT18	SCHMITT17	SCHMITT16

38.8.99 GMAC Transmit LPI Transitions

Name: GMAC_TXLPI
Offset: 0x278
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – COUNT[23:0] Count of LIP Transitions
A count of the number of times the bit TXLPIEN (Enable LPI Transmission (bit 19)) goes from low to high in the GMAC_NCR.

Cleared on read.

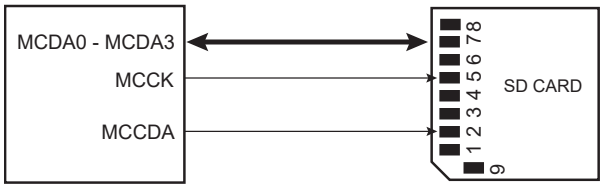
Table 40-3. SD Memory Card Bus Signals

Pin Number	Name	Type(1)	Description	HSMCI Pin Name(2) (Slot z)
1	CD/DAT[3]	I/O/PP	Card detect/ Data line Bit 3	MCDz3
2	CMD	PP	Command/response	MCCDz
3	VSS1	S	Supply voltage ground	VSS
4	VDD	S	Supply voltage	VDD
5	CLK	I/O	Clock	MCCK
6	VSS2	S	Supply voltage ground	VSS
7	DAT[0]	I/O/PP	Data line Bit 0	MCDz0
8	DAT[1]	I/O/PP	Data line Bit 1 or Interrupt	MCDz1
9	DAT[2]	I/O/PP	Data line Bit 2	MCDz2

Notes: 1. I: input, O: output, PP: Push Pull, OD: Open Drain.

2. When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA, MCDAy to HSMCIx_DAy.

Figure 40-6. SD Card Bus Connections with One Slot



Note: When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA MCDAy to HSMCIx_DAy.

When the HSMCI is configured to operate with SD memory cards, the width of the data bus can be selected in the HSMCI_SDCR. Clearing the SDCBUS bit in this register means that the width is one bit; setting it means that the width is four bits. In the case of High Speed MultiMedia cards, only the data line 0 is used. The other data lines can be used as independent PIOs.

40.8 High-Speed Multimedia Card Operations

After a power-on reset, the cards are initialized by a special message-based High-Speed Multimedia Card bus protocol. Each message is represented by one of the following tokens:

- Command—A command is a token that starts an operation. A command is sent from the host either to a single card (addressed command) or to all connected cards (broadcast command). A command is transferred serially on the CMD line.
- Response—A response is a token which is sent from an addressed card or (synchronously) from all connected cards to the host as an answer to a previously received command. A response is transferred serially on the CMD line.
- Data—Data can be transferred from the card to the host or vice versa. Data is transferred via the data line.

2. Configure the Slave mode.
3. Enable the DMA.
4. Wait for the DMA status flag indicating that the buffer transfer is complete.
5. Disable the DMA.
6. (Only if peripheral clock must be disabled) Wait for the TXCOMP flag to be raised in TWIHS_SR.

43.6.5.5.2 Data Receive with the DMA in Slave Mode

The following procedure shows an example to transmit data with DMA where the number of characters to receive is known.

1. Initialize the DMA (channels, memory pointers, size, etc.).
2. Configure the Slave mode.
3. Enable the DMA.
4. Wait for the DMA status flag indicating that the buffer transfer is complete.
5. Disable the DMA.
6. (Only if peripheral clock must be disabled) Wait for the TXCOMP flag to be raised in TWIHS_SR.

43.6.5.6 SMBus Mode

SMBus mode is enabled when a one is written to TWIHS_CR.SMBEN. SMBus mode operation is similar to I²C operation with the following exceptions:

- Only 7-bit addressing can be used.
- The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into the TWIHS_SMBTR.
- Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
- A set of addresses have been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring the TWIHS_CR.

43.6.5.6.1 Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing a one to TWIHS_CR.PECEN will send/check the PEC field in the current transfer. The PEC generator is always updated on every bit transmitted or received, so that PEC handling on the following linked transfers is correct.

In Slave Receiver mode, the master calculates a PEC value and transmits it to the slave after all data bytes have been transmitted. Upon reception of this PEC byte, the slave compares it to the PEC value it has computed itself. If the values match, the data was received correctly, and the slave returns an ACK to the master. If the PEC values differ, data was corrupted, and the slave returns a NACK value. TWIHS_SR.PECERR is set automatically if a PEC error occurred.

In Slave Transmitter mode, the slave calculates a PEC value and transmits it to the master after all data bytes have been transmitted. Upon reception of this PEC byte, the master compares it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the master must take appropriate action.

See Slave Read Write Flowcharts for detailed flowcharts.

43.6.5.6.2 Timeouts

The TWIHS SMBus Timing Register (TWIHS_SMBTR) configures the SMBus timeout values. If a timeout occurs, the slave leaves the bus. The TOUT bit is also set in TWIHS_SR.

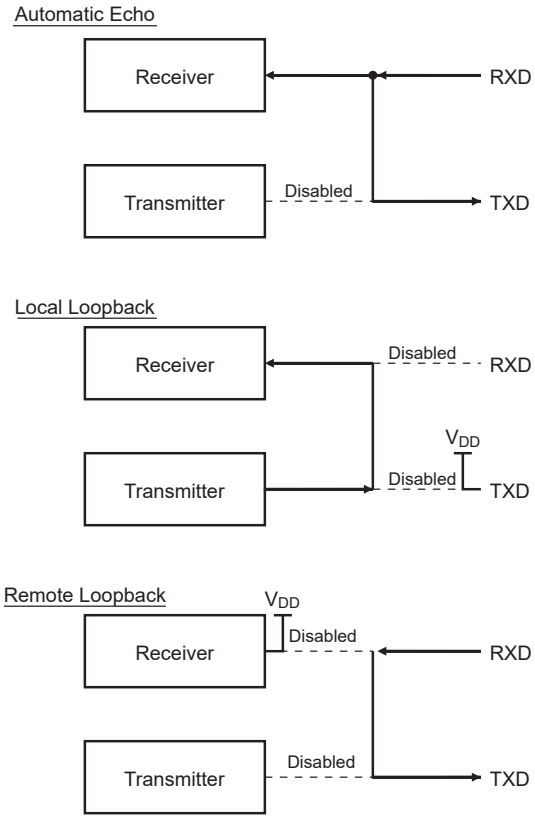
44.9 Register Summary

Note: Offsets 0x100–0x128 are reserved for PDC registers.

Offset	Name	Bit Pos.								
0x00	SSC_CR	7:0							RXDIS	RXEN
		15:8	SWRST						TXDIS	TXEN
		23:16								
		31:24								
0x04	SSC_CMR	7:0	DIV[7:0]							
		15:8					DIV[11:8]			
		23:16								
		31:24								
0x08 ... 0x0F	Reserved									
0x10	SSC_RCMR	7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
		15:8				STOP	START[3:0]			
		23:16	STTDLY[7:0]							
		31:24	PERIOD[7:0]							
0x14	SSC_RFMR	7:0	MSBF		LOOP	DATLEN[4:0]				
		15:8					DATNB[3:0]			
		23:16		FSOS[2:0]			FSLEN[3:0]			
		31:24	FSLEN_EXT[3:0]							FSEDGE
0x18	SSC_TCMR	7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
		15:8					START[3:0]			
		23:16	STTDLY[7:0]							
		31:24	PERIOD[7:0]							
0x1C	SSC_TFMR	7:0	MSBF		DATDEF	DATLEN[4:0]				
		15:8					DATNB[3:0]			
		23:16	FSDEN	FSOS[2:0]			FSLEN[3:0]			
		31:24	FSLEN_EXT[3:0]							FSEDGE
0x20	SSC_RHR	7:0	RDAT[7:0]							
		15:8	RDAT[15:8]							
		23:16	RDAT[23:16]							
		31:24	RDAT[31:24]							
0x24	SSC_THR	7:0	TDAT[7:0]							
		15:8	TDAT[15:8]							
		23:16	TDAT[23:16]							
		31:24	TDAT[31:24]							
0x28 ... 0x2F	Reserved									
0x30	SSC_RSHR	7:0	RSDAT[7:0]							
		15:8	RSDAT[15:8]							
		23:16								
		31:24								

Offset	Name	Bit Pos.								
		31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
0x0C	US_IDR (LON_MODE)	7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
		15:8						UNRE	TXEMPTY	
		23:16								
		31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
0x10	US_IMR	7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		23:16					CTSIC	DCDIC	DSRIC	RIIC
		31:24								MANE
0x10	US_IMR (SPI_MODE)	7:0			OVRE				TXRDY	RXRDY
		15:8						UNRE	TXEMPTY	
		23:16					NSSE			
		31:24								
0x10	US_IMR (LIN_MODE)	7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		23:16								
		31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
0x10	US_IMR (LON_MODE)	7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
		15:8						UNRE	TXEMPTY	
		23:16								
		31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
0x14	US_CSR	7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		23:16	CTS	DCD	DSR	RI	CTSIC	DCDIC	DSRIC	RIIC
		31:24								MANERR
0x14	US_CSR (SPI_MODE)	7:0			OVRE				TXRDY	RXRDY
		15:8						UNRE	TXEMPTY	
		23:16	NSS				NSSE			
		31:24								
0x14	US_CSR (LIN_MODE)	7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		23:16	LINBLS							
		31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
0x14	US_CSR (LON_MODE)	7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
		15:8						UNRE	TXEMPTY	
		23:16								
		31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
0x18	US_RHR	7:0	RXCHR[7:0]							
		15:8	RXSYNH							RXCHR[8:8]
		23:16								
		31:24								
0x1C	US_THR	7:0	TXCHR[7:0]							
		15:8	TXSYNH							TXCHR[8:8]
		23:16								
		31:24								
0x20	US_BRGR	7:0	CD[7:0]							

Figure 47-14. Test Modes



Buffer element is described in Tx Buffer Element. The table below describes the possible configurations for frame transmission.

Table 49-5. Possible Configurations for Frame Transmission

MCAN_CCCR		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when MCAN_TXBRP is updated, or when a transmission has been started.

49.5.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit MCAN_CCCR.TXP. If the bit is set, the MCAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (MCAN_CCCR.TXP = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

49.5.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the processor. Each dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via MCAN_TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

49.6.18 MCAN Interrupt Line Select Register

Name: MCAN_ILS
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines.

0: Interrupt assigned to interrupt line MCAN_INT0.

1: Interrupt assigned to interrupt line MCAN_INT1.

Bit	31	30	29	28	27	26	25	24
			ARAL	PEDL	PEAL	WDIL	BOL	EWL
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAL Access to Reserved Address Line

Bit 28 – PEDL Protocol Error in Data Phase Line

Bit 27 – PEAL Protocol Error in Arbitration Phase Line

Bit 26 – WDIL Watchdog Interrupt Line

Bit 25 – BOL Bus_Off Status Interrupt Line

Bit 24 – EWL Warning Status Interrupt Line

Bit 23 – EPL Error Passive Interrupt Line

Bit 22 – ELOL Error Logging Overflow Interrupt Line

Bit 19 – DRXL Message stored to Dedicated Receive Buffer Interrupt Line

49.6.31 MCAN Receive FIFO 1 Configuration

Name: MCAN_RXF1C
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
	F1OM	F1WM[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		F1S[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	F1SA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	F1SA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 31 – F1OM FIFO 1 Operation Mode
FIFO 1 can be operated in Blocking or in Overwrite mode (see Rx FIFOs).

Value	Description
0	FIFO 1 Blocking mode.
1	FIFO 1 Overwrite mode.

Bits 30:24 – F1WM[6:0] Receive FIFO 1 Watermark

Value	Description
0	Watermark interrupt disabled
1-64	Level for Receive FIFO 1 watermark interrupt (MCAN_IR.RF1W).
>64	Watermark interrupt disabled.

Bits 22:16 – F1S[6:0] Receive FIFO 1 Size
The elements in Receive FIFO 1 are indexed from 0 to F1S - 1.

Value	Description
0	No Receive FIFO 1
1-64	Number of elements in Receive FIFO 1.
>64	Values greater than 64 are interpreted as 64.

50.7.12 TC Interrupt Disable Register

Name: TC_IDRx
Offset: 0x28 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

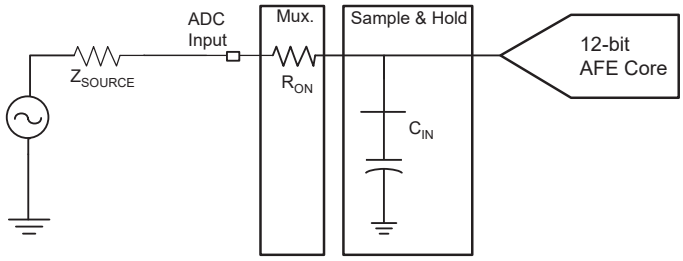
Table 58-38. Z_{IN} Input Impedance

f_S (MHz)	1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.007813
C_{IN} = 2 pF								
Z_{IN} (MΩ)	0.5	1	2	4	8	16	32	64
C_{IN} = 4 pF								
Z_{IN} (MΩ)	0.25	0.5	1	2	4	8	16	32
C_{IN} = 8 pF								
Z_{IN} (MΩ)	0.125	0.25	0.5	1	2	4	8	16

58.8.6.1 Track and Hold Time versus Source Output Impedance

The figure below shows a simplified acquisition path.

Figure 58-16. Simplified Acquisition Path



During the tracking phase, the AFE tracks the input signal during the tracking time shown below:

$t_{TRACK} = n \times C_{IN} \times (R_{ON}+Z_{SOURCE})/1000$

- Tracking time expressed in ns and Z_{SOURCE} expressed in Ω.
- n depends on the expected accuracy
- R_{ON}= 2 kOhm

Table 58-39. Number of Tau:n

Resolution (bits)	12	13	14	15	16
RES	0	2	3	4	5
n	8	9	10	11	12

The AFEC already includes a tracking time of 15 $t_{AFE\ Clock.}$

58.8.6.2 AFE DAC Offset Compensation

Table 58-40. DAC Static Performances (see Note 1)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
N	Resolution (see **Note 2**)	–	–	9	10	LSB
INL	Integral Non Linearity	–	-2.5	±0.7	2	LSB
DNL	Differential Non Linearity	–	-3	±0.5	1.8	LSB

Note:

1. DAC Offset is included in the AFE EO performances.

Symbol	Parameter	Conditions	Min	Max	Unit
		1.8V domain	0.8	–	ns
SPI_{12}	NPCS setup to SPCK rising (slave)	3.3V domain	3.9	–	ns
		1.8V domain	4.4	–	ns
SPI_{13}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
SPI_{14}	NPCS setup to SPCK falling (slave)	3.3V domain	4.0	–	ns
		1.8V domain	4.1	–	ns
SPI_{15}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns

Timings are given for the 3.3V domain, with V_{DDIO} from 2.85V to 3.6V, maximum external capacitor = 40 pF.

Table 58-57. SPI Timings

Symbol	Parameter	Conditions	Min	Max	Unit
SPI_0	MISO Setup time before SPCK rises (master)	3.3V domain	12.4	–	ns
SPI_1	MISO Hold time after SPCK rises (master)	3.3V domain	0	–	ns
SPI_2	SPCK rising to MOSI Delay (master)	3.3V domain	-3.7	2.2	ns
SPI_3	MISO Setup time before SPCK falls (master)	3.3V domain	12.6	–	ns
SPI_4	MISO Hold time after SPCK falls (master)	3.3V domain	0	–	ns
SPI_5	SPCK falling to MOSI Delay (master)	3.3V domain	-3.6	2.0	ns
SPI_6	SPCK falling to MISO Delay (slave)	3.3V domain	3.0	11.9	ns
SPI_7	MOSI Setup time before SPCK rises (slave)	3.3V domain	1.2	–	ns
SPI_8	MOSI Hold time after SPCK rises (slave)	3.3V domain	0.6	–	ns
SPI_9	SPCK rising to MISO Delay (slave)	3.3V domain	3.0	12.0	ns
SPI_{10}	MOSI Setup time before SPCK falls (slave)	3.3V domain	1.2	–	ns
SPI_{11}	MOSI Hold time after SPCK falls (slave)	3.3V domain	0.6	–	ns
SPI_{12}	NPCS setup to SPCK rising (slave)	3.3V domain	3.9	–	ns
SPI_{13}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns
SPI_{14}	NPCS setup to SPCK falling (slave)	3.3V domain	4.0	–	ns
SPI_{15}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns

Note that in SPI master mode, the device does not sample the data (MISO) on the opposite edge where the data clocks out (MOSI), but the same edge is used. See Figure 58-19 and Figure 58-20.

Signal Name	Recommended Pin Connection	Description
TD	Application dependent.	SSC Transmit Data Pulled-up input (100 kOhm) to VDDIO at reset.
RD	Application dependent.	SSC Receive Data Pulled-up input (100 kOhm) to VDDIO at reset.
TK	Application dependent.	SSC Transmit Clock Pulled-up input (100 kOhm) to VDDIO at reset.
RK	Application dependent.	SSC Receive Clock I Pulled-up input (100 kOhm) to VDDIO at reset.
TF	Application dependent.	SSC Transmit Frame Sync Pulled-up input (100 kOhm) to VDDIO at reset.
RF	Application dependent.	SSC Receive Frame Sync Pulled-up input (100 kOhm) to VDDIO at reset.
Image Sensor Interface		
ISI_D0–ISI_D11	Application dependent. (Signal can be level-shifted depending on the image sensor characteristics)	Image Sensor Data Pulled-up inputs (100 kOhm) to VDDIO at reset.
ISI_MCK	Application dependent. (Signal can be level-shifted depending on the image sensor characteristics)	Image sensor reference clock. No dedicated signal, PCK1 can be used. Pulled-up input (100 kOhm) to VDDIO at reset.
ISI_HSYNC	Application dependent. (Signal can be level-shifted depending on the image sensor characteristics)	Image sensor horizontal synchro Pulled-up input (100 kOhm) to VDDIO at reset.
ISI_VSYNC	Application dependent. (Signal can be level-shifted depending on the image sensor characteristics)	Image sensor vertical synchro Pulled-up input (100 kOhm) to VDDIO at reset.
ISI_PCK	Application dependent. (Signal can be level-shifted depending on the image sensor characteristics)	Image sensor data clock Pulled-up input (100 kOhm) to VDDIO at reset.
Timer/Counter		
TCLKx	Application dependent.	TC Channel x External Clock Input Pulled-up inputs (100 kOhm) to VDDIO at reset.
TIOAx	Application dependent.	TC Channel x I/O Line A

Date	Changes
	Removed redundant Section 15.7.2. NRST Pin and Section 15.7.3. ERASE Pin (already in Section 8. “Input/Output Lines”). Removed references to Embedded Trace Buffer (ETB). Section 16.7.8 ”IEEE1149.1 JTAG Boundary Scan”: updated condtions to enable boundary scan.
	Section 18. “Fast Flash Programming Interface (FFPI)” Table 18-1 “Signal Description List“: updated XIN information. Deleted comment for XIN. Section 18.3 “Parallel Fast Flash Programming”, Figure 18-1, “16-bit Parallel Programming Interface”: changed input source for XIN. Section 18.3.3 “Entering Parallel Programming Mode”: deleted note on device clocking. Reworded steps 2 and 3.
	Section 19. “Bus Matrix (MATRIX)” Table 19-4 “Register Mapping”: corrected reset values for MATRIX_PRASx and MATRIX_PRBSx registers. In Section 19.4.8 “SMC NAND Flash Chip Select Configuration Register”: - added warning to bit description SMC_NFCS1. - changed SDRAMEN bit description and added warning.
	Section 22. “Enhanced Embedded Flash Controller (EEFC)” Updated Section 22.2 “Embedded Characteristics”. Added Figure 22-1, “Flash Memory Areas”. Section 22.4.3.6 “Calibration Bit”: updated oscillators that are calibrated in production. Section 22.4.3.7 “Security Bit Protection”: added detail on ETM.
	Section 23. “Supply Controller (SUPC)” Figure 23-2, “Separate Backup Supply Powering Scheme”: updated figure and corrected min voltage in note on ADC/DAC/ACC.
	Section 24. “Watchdog Timer (WDT)” Section 24.1 “Description”: Replaced “Idle mode” with “Sleep mode (Idle mode)”. Section 24.4 “Functional Description”: replaced “Idle mode” with “Sleep mode” Section 24.4 “Functional Description”, Section 24.5.2 “Watchdog Timer Mode Register”: modified information on WDDIS bit setting to read “When setting the WDDIS bit, and while it is set, the fields WDV and WDD must not be modified.” Section 24.5.1 “Watchdog Timer Control Register”: added note on modification of WDT_CR values.. Section 24.5.2 “Watchdog Timer Mode Register”: added Note (2) on modification of WDT_MR values.
	Section 25. “Reinforced Safety Watchdog Timer (RSWDT)”