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, Ethernet, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 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/atsame70n21b-an

12. Event System

The events generated by peripherals (source) are designed to be directly routed to peripherals (destination) using these events without processor intervention. The trigger source can be programmed in the destination peripheral.

12.1 Embedded Characteristics

- Timers, PWM, I/Os and peripherals generate event triggers which are directly routed to destination peripherals, such as AFEC or DACC to start measurement/conversion without processor intervention.
- UART, USART, QSPI, SPI, TWI, PWM, HSMCI, AES, AFEC, DACC, PIO, TC (Capture mode) also generate event triggers directly connected to the DMA Controller for data transfer without processor intervention.
- Parallel capture logic is directly embedded in the PIO and generates trigger events to the DMA Controller to capture data without processor intervention.
- PWM safety events (faults) are in combinational form and directly routed from event generators (AFEC, ACC, PMC, TC) to the PWM module.
- PWM output comparators (OCx) generate events directly connected to the TC.
- PMC safety event (clock failure detection) can be programmed to switch the MCK on reliable main RC internal clock without processor intervention.

12.2 Real-time Event Mapping

Table 12-1. Real-time Event Mapping List

Function	Application	Description	Event Source	Event Destination
Safety	General-purpose	Automatic switch to reliable main RC oscillator in case of main crystal clock failure (see **Note 1**)	Power Management Controller (PMC)	PMC
	General-purpose, motor control, power factor correction (PFC)	Puts the PWM outputs in Safe mode in case of main crystal clock failure (see **Notes 1, 2**)	PMC	Pulse Width Modulation 0 and 1 (PWM0 and PWM1)
	Motor control, PFC	Puts the PWM outputs in Safe mode (overcurrent detection, etc.) (see **Notes 2, 3**)	Analog Comparator Controller (ACC)	PWM0 and PWM1
	Motor control, PFC	Puts the PWM outputs in Safe mode (overspeed, overcurrent detection, etc.) (see **Notes 2, 4**)	Analog Front-End Controller (AFEC0)	PWM0 and PWM1
			AFEC1	PWM0 and PWM1

29. General Purpose Backup Registers (GPBR)

29.1 Description

The System Controller embeds 128 bits of General Purpose Backup registers organized as 8 32-bit registers.

It is possible to generate an immediate clear of the content of General Purpose Backup registers 0 to 3 (first half) if a Low-power Debounce event is detected on one of the wakeup pins, WKUP0 or WKUP1. The content of the other General Purpose Backup registers (second half) remains unchanged.

The Supply Controller module must be programmed accordingly. In the register SUPC_WUMR in the Supply Controller module, LPDBCCLR, LPDBCEN0 and/or LPDBCEN1 bit must be configured to 1 and LPDBC must be other than 0.

If a Tamper event has been detected, it is not possible to write to the General Purpose Backup registers while the LPDBCS0 or LPDBCS1 flags are not cleared in the Supply Controller Status Register (SUPC_SR).

29.2 Embedded Characteristics

- 128 bits of General Purpose Backup Registers
- Immediate Clear on Tamper Event

38.8.24 GMAC Type ID Match n Register

Name: GMAC_TIDMx
Offset: 0xA8 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENIDn							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TID[15:8]							
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
	TID[7:0]							
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 31 – ENIDn Enable Copying of TID Matched Frames

Value	Description
0	TID n is not part of the comparison match.
1	TID n is processed for the comparison match.

Bits 15:0 – TID[15:0] Type ID Match n
For use in comparisons with received frames type ID/length frames.

Offset	Name	Bit Pos.								
0x0364	USBHS_DEVDMAADDRESS7	7:0	BUFF_ADD[7:0]							
		15:8	BUFF_ADD[15:8]							
		23:16	BUFF_ADD[23:16]							
		31:24	BUFF_ADD[31:24]							
0x0368	USBHS_DEVDMACONTROL7	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x036C	USBHS_DEVDMASTATUS7	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0370 ... 0x03FF	Reserved									
0x0400	USBHS_HSTCTRL	7:0								
		15:8			SPDCONF[1:0]			RESUME	RESET	SOFE
		23:16								
		31:24								
0x0404	USBHS_HSTISR	7:0		HWUPI	HSOFI	RXRSMI	RSMEDI	RSTI	DDISCI	DCONNI
		15:8	PEP_7	PEP_6	PEP_5	PEP_4	PEP_3	PEP_2	PEP_1	PEP_0
		23:16							PEP_9	PEP_8
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x0408	USBHS_HSTICR	7:0		HWUPIC	HSOFIC	RXRSMIC	RSMEDIC	RSTIC	DDISCIC	DCONNIC
		15:8								
		23:16								
		31:24								
0x040C	USBHS_HSTIFR	7:0		HWUPIS	HSOFIS	RXRSMIS	RSMEDIS	RSTIS	DDISCIS	DCONNIS
		15:8								
		23:16								
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x0410	USBHS_HSTIMR	7:0		HWUPIE	HSOFIE	RXRSMIE	RSMEDIE	RSTIE	DDISCIE	DCONNIE
		15:8	PEP_7	PEP_6	PEP_5	PEP_4	PEP_3	PEP_2	PEP_1	PEP_0
		23:16							PEP_9	PEP_8
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x0414	USBHS_HSTIDR	7:0		HWUPIEC	HSOFIEC	RXRSMIEC	RSMEDIEC	RSTIEC	DDISCIEC	DCONNIEC
		15:8	PEP_7	PEP_6	PEP_5	PEP_4	PEP_3	PEP_2	PEP_1	PEP_0
		23:16							PEP_9	PEP_8
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x0418	USBHS_HSTIER	7:0		HWUPIES	HSOFIES	RXRSMIES	RSMEDIES	RSTIES	DDISCIES	DCONNIES
		15:8	PEP_7	PEP_6	PEP_5	PEP_4	PEP_3	PEP_2	PEP_1	PEP_0
		23:16							PEP_9	PEP_8
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x041C	USBHS_HSTPIP	7:0	PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0
		15:8								PEN8
		23:16	PRST7	PRST6	PRST5	PRST4	PRST3	PRST2	PRST1	PRST0

Offset	Name	Bit Pos.								
		31:24								
0x0590	USBHS_HSTPIPIFR0 (INTPIPES)	7:0	SHORTPACKETIS	RXSTALLDIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0590	USBHS_HSTPIPIFR0 (ISOPIPES)	7:0	SHORTPACKETIS	CRCERRIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0594	USBHS_HSTPIPIFR1 (INTPIPES)	7:0	SHORTPACKETIS	RXSTALLDIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0594	USBHS_HSTPIPIFR1 (ISOPIPES)	7:0	SHORTPACKETIS	CRCERRIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0598	USBHS_HSTPIPIFR2 (INTPIPES)	7:0	SHORTPACKETIS	RXSTALLDIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0598	USBHS_HSTPIPIFR2 (ISOPIPES)	7:0	SHORTPACKETIS	CRCERRIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x059C	USBHS_HSTPIPIFR3 (INTPIPES)	7:0	SHORTPACKETIS	RXSTALLDIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x059C	USBHS_HSTPIPIFR3 (ISOPIPES)	7:0	SHORTPACKETIS	CRCERRIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x05A0	USBHS_HSTPIPIFR4 (INTPIPES)	7:0	SHORTPACKETIS	RXSTALLDIS	OVERFIS	NAKEDIS	PERRIS	UNDERFIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								

39.6.68 Host DMA Channel x Status Register

Name: USBHS_HSTDMASTATUSx
Offset: 0x070C + x*0x10 [x=0..6]
Reset: 0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BUFF_COUNT[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUFF_COUNT[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
Access								
Reset		0	0	0			0	0

Bits 31:16 – BUFF_COUNT[15:0] Buffer Byte Count
This field determines the current number of bytes still to be transferred for this buffer.

This field is decremented from the AHB source bus access byte width at the end of this bus address phase.

The access byte width is 4 by default, or less, at DMA start or end, if the start or end address is not aligned on a word boundary.

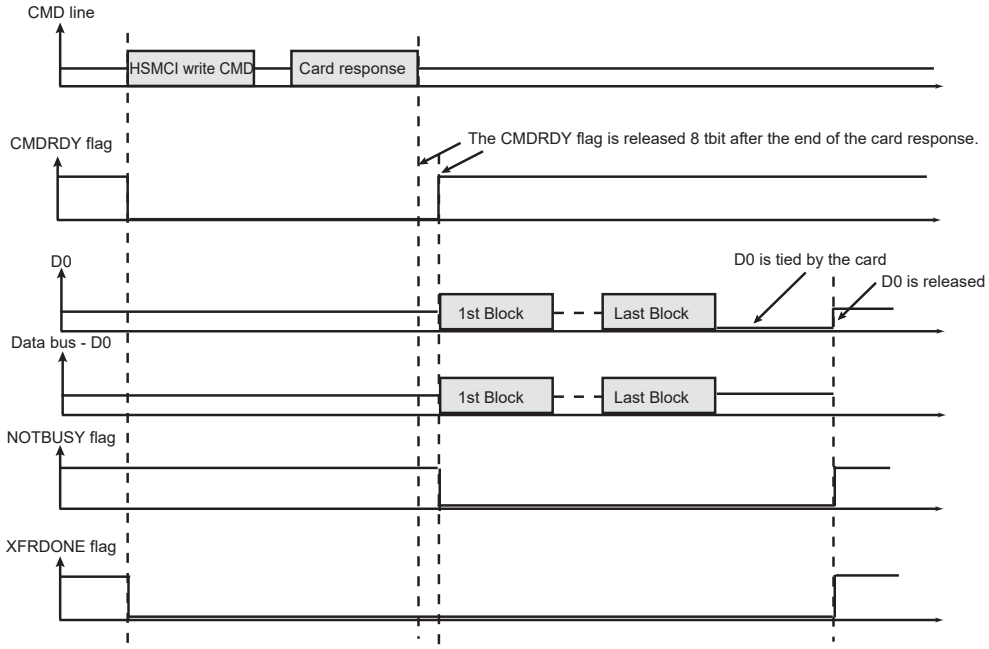
At the end of buffer, the DMA accesses the USBHS device only for the number of bytes needed to complete it.

Note: For IN pipes, if the receive buffer byte length (USBHS_HSTDMACONTROL.BUFF_LENGTH) has been defaulted to zero because the USB transfer length is unknown, the actual buffer byte length received is 0x10000-BUFF_COUNT.

Bit 6 – DESC_LDST Descriptor Loaded Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when a descriptor has been loaded from the system bus.

Figure 40-12. XFRDONE During a Write Access



40.13 Register Write Protection

To prevent any single software error from corrupting HSMCI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the HSMCI Write Protection Mode Register (HSMCI_WPMR).

If a write access to a write-protected register is detected, the WPVS bit in the HSMCI Write Protection Status Register (HSMCI_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the HSMCI_WPSR.

The following registers can be protected:

- HSMCI Mode Register
- HSMCI Data Timeout Register
- HSMCI SDCard/SDIO Register
- HSMCI Completion Signal Timeout Register
- HSMCI DMA Configuration Register
- HSMCI Configuration Register

41.8.6 SPI Interrupt Enable Register

Name: SPI_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

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

0: No effect.

1: Enables 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
						UNDES	TXEMPTY	NSSR
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 10 – UNDES Underrun Error Interrupt Enable

Bit 9 – TXEMPTY Transmission Registers Empty Enable

Bit 8 – NSSR NSS Rising Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – MODF Mode Fault Error Interrupt Enable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Enable

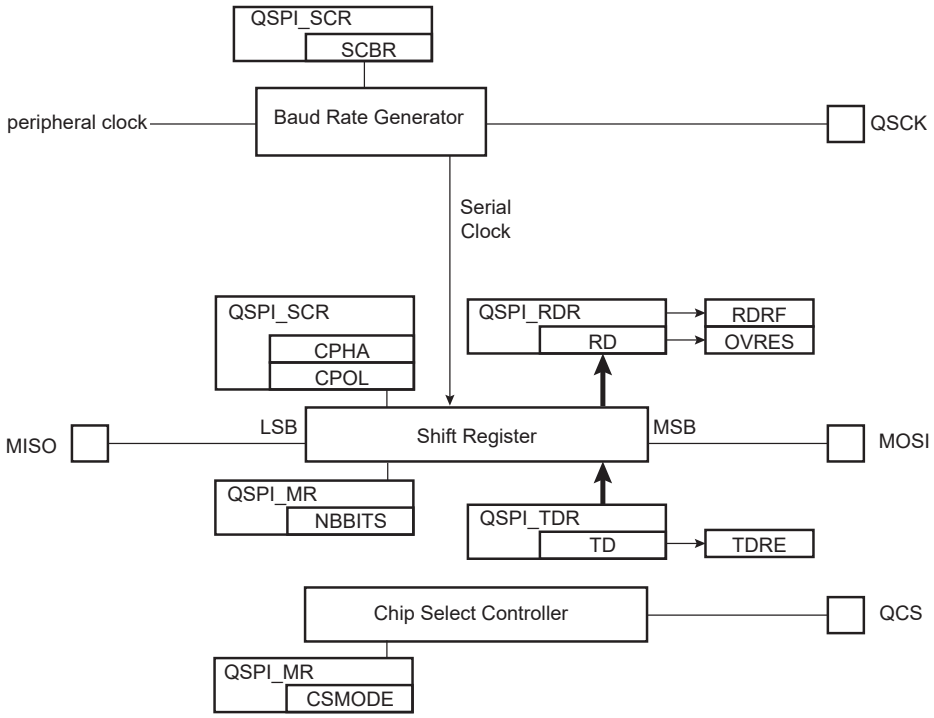
Bit 0 – RDRF Receive Data Register Full Interrupt Enable

If the QSPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in QSPI_SR is set. As long as this flag is set, data is loaded in QSPI_RDR. The user must read the QSPI_SR to clear the OVRES bit.

The following figures show, respectively, a block diagram of the SPI when operating in Master mode, and a flow chart describing how transfers are handled.

42.6.4.2 SPI Mode Block Diagram

Figure 42-5. SPI Mode Block Diagram



43.7.5 TWIHS Clock Waveform Generator Register

Name: TWIHS_CWGR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWIHS Write Protection Mode Register.

TWIHS_CWGR is used in Master mode only.

Bit	31	30	29	28	27	26	25	24
			HOLD[5:0]					
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
						CKDIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	CHDIV[7:0]							
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	7	6	5	4	3	2	1	0
	CLDIV[7:0]							
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

Bits 29:24 – HOLD[5:0] TWD Hold Time Versus TWCK Falling
If High-speed mode is selected TWD is internally modified on the TWCK falling edge to meet the I2C specified maximum hold time, else if High-speed mode is not configured TWD is kept unchanged after TWCK falling edge for a period of (HOLD + 3) × $t_{peripheral\ clock}$.

Bits 18:16 – CKDIV[2:0] Clock Divider
The CKDIV is used to increase both SCL high and low periods.

Bits 15:8 – CHDIV[7:0] Clock High Divider
The SCL high period is defined as follows:

$t_{high} = ((CHDIV \times 2^{CKDIV}) + 3) \times t_{peripheral\ clock}$

Bits 7:0 – CLDIV[7:0] Clock Low Divider
The SCL low period is defined as follows:

$t_{low} = ((CLDIV \times 2^{CKDIV}) + 3) \times t_{peripheral\ clock}$

Value	Description
0	The USART does not drive the SCK pin.
1	The USART drives the SCK pin if USCLKS does not select the external clock SCK.

Bit 17 – MODE9 9-bit Character Length

Value	Description
0	CHRL defines character length.
1	9-bit character length.

Bit 16 – MSBF Bit Order

Value	Description
0	Least significant bit is sent/received first.
1	Most significant bit is sent/received first.

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic Echo. Receiver input is connected to the TXD pin.
2	LOCAL_LOOPBACK	Local Loopback. Transmitter output is connected to the Receiver Input.
3	REMOTE_LOOPBACK	Remote Loopback. RXD pin is internally connected to the TXD pin.

Bits 13:12 – NBSTOP[1:0] Number of Stop Bits

Value	Name	Description
0	1_BIT	1 stop bit
1	1_5_BIT	1.5 stop bit (SYNC = 0) or reserved (SYNC = 1)
2	2_BIT	2 stop bits

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even parity
1	ODD	Odd parity
2	SPACE	Parity forced to 0 (Space)
3	MARK	Parity forced to 1 (Mark)
4	NO	No parity
6	MULTIDROP	Multidrop mode

Bit 8 – SYNC Synchronous Mode Select

Value	Description
0	USART operates in Asynchronous mode.
1	USART operates in Synchronous mode.

Bits 7:6 – CHRL[1:0] Character Length

47.4 Product Dependencies

47.4.1 I/O Lines

The UART pins are multiplexed with PIO lines. The user must first configure the corresponding PIO Controller to enable I/O line operations of the UART.

47.4.2 Power Management

The UART clock can be controlled through the Power Management Controller (PMC). In this case, the user must first configure the PMC to enable the UART clock. Usually, the peripheral identifier used for this purpose is 1.

In SleepWalking mode (asynchronous partial wake-up), the PMC must be configured to enable SleepWalking for the UART in the Sleepwalking Enable Register (PMC_SLPWK_ER). Depending on the instructions (requests) provided by the UART to the PMC, the system clock may or may not be automatically provided to the UART.

47.4.3 Interrupt Sources

The UART interrupt line is connected to one of the interrupt sources of the Interrupt Controller. Interrupt handling requires programming of the Interrupt Controller before configuring the UART.

47.5 Functional Description

The UART operates in Asynchronous mode only and supports only 8-bit character handling (with parity). It has no clock pin.

The UART is made up of a receiver and a transmitter that operate independently, and a common baud rate generator. Receiver timeout and transmitter time guard are not implemented. However, all the implemented features are compatible with those of a standard USART.

47.5.1 Baud Rate Generator

The baud rate generator provides the bit period clock named baud rate clock to both the receiver and the transmitter.

The baud rate clock is the peripheral clock divided by 16 times the clock divisor (CD) value written in the Baud Rate Generator register (UART_BRGR). If UART_BRGR is set to 0, the baud rate clock is disabled and the UART remains inactive. The maximum allowable baud rate is peripheral clock orPMC PCK (PCK) divided by 16. The minimum allowable baud rate is peripheral clock or PCK divided by (16 x 65536).The clock source driving the baud rate generator (peripheral clock or PCK) can be selected by writing the bit BRSRCCK in UART_MR.

If PCK is selected, the baud rate is independent of the processor/bus clock. Thus the processor clock can be changed while UART is enabled. The processor clock frequency changes must be performed only by programming the field PRES in PMC_MCKR (see "Power Management Controller (PMC)"). Other methods to modify the processor/bus clock frequency (PLL multiplier, etc.) are forbidden when UART is enabled.

The peripheral clock frequency must be at least three times higher than PCK.

48.7.6 MediaLB Interrupt Enable Register

Name: MLB_MIEN
Offset: 0x02C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			CTX_BREAK	CTX_PE	CTX_DONE	CRX_BREAK	CRX_PE	CRX_DONE
Access								
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		ATX_BREAK	ATX_PE	ATX_DONE	ARX_BREAK	ARX_PE	ARX_DONE	SYNC_PE
Access								
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ISOC_BUFO	ISOC_PE
Access								
Reset							0	0

Bit 29 – CTX_BREAK Control Tx Break Enable

Value	Description
1	A ReceiverBreak response received from the receiver on a control Tx channel causes the appropriate channel bit in the MLB_MS0 or MLB_MS1 registers to be set.

Bit 28 – CTX_PE Control Tx Protocol Error Enable

Value	Description
1	A ProtocolError generated by the receiver on a control Tx channel causes the appropriate channel bit in the MLB_MS0 or MLB_MS1 registers to be set.

Bit 27 – CTX_DONE Control Tx Packet Done Enable

Value	Description
1	A packet transmitted with no errors on a control Tx channel causes the appropriate channel bit in the MLB_MS0 or MLB_MS1 registers to be set.

Bit 26 – CRX_BREAK Control Rx Break Enable
Rx channel causes the appropriate channel bit in the MLB_MS0 or MLB_MS1 registers to be set.

Value	Description
1	A ControlBreak command received from the transmitter on a control.

49.6.39 MCAN Transmit Buffer Add Request

Name: MCAN_TXBAR
Offset: 0xD0
Reset: 0x00000000
Property: Read/Write

If an add request is applied for a Transmit Buffer with pending transmission request (corresponding MCAN_TXBRP bit already set), this Add Request is ignored.

Bit	31	30	29	28	27	26	25	24
	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
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
	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
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	15	14	13	12	11	10	9	8
	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
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
	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
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

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – ARx Add Request for Transmit Buffer x
Each Transmit 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 processor to set transmission requests for multiple Transmit Buffers with one write to MCAN_TXBAR. MCAN_TXBAR bits are set only for those Transmit Buffers configured via TXBC. When no Transmit scan is running, the bits are reset immediately, else the bits remain set until the Transmit scan process has completed.

Value	Description
0	No transmission request added.
1	Transmission requested added.

FMOD configuration to use must be FMOD = 1, to avoid spurious fault detection. Refer to the corresponding peripheral documentation for details on handling fault generation.

Fault inputs may or may not be glitch-filtered depending on the FFIL field in PWM_FMR. When the filter is activated, glitches on fault inputs with a width inferior to the PWM peripheral clock period are rejected.

A fault becomes active as soon as its corresponding fault input has a transition to the programmed polarity level. If the corresponding bit FMOD is set to '0' in PWM_FMR, the fault remains active as long as the fault input is at this polarity level. If the corresponding FMOD field is set to '1', the fault remains active until the fault input is no longer at this polarity level and until it is cleared by writing the corresponding bit FCLR in the PWM Fault Clear Register (PWM_FCR). In the PWM Fault Status Register (PWM_FSR), the field FIV indicates the current level of the fault inputs and the field FIS indicates whether a fault is currently active.

Each fault can be taken into account or not by the fault protection mechanism in each channel. To be taken into account in the channel x, the fault y must be enabled by the bit FPEx[y] in the PWM Fault Protection Enable register (PWM_FPE1). However, synchronous channels (see Synchronous Channels) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[y]).

The fault protection on a channel is triggered when this channel is enabled and when any one of the faults that are enabled for this channel is active. It can be triggered even if the PWM peripheral clock is not running but only by a fault input that is not glitch-filtered.

When the fault protection is triggered on a channel, the fault protection mechanism resets the counter of this channel and forces the channel outputs to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV) and fields FPZHx/FPZLx in the PWM Fault Protection Value Register 2, as shown in the table below. The output forcing is made asynchronously to the channel counter.

Table 51-3. Forcing Values of PWM Outputs by Fault Protection

FPZH/Lx	FPVH/Lx	Forcing Value of PWMH/Lx
0	0	0
0	1	1
1	–	High impedance state (Hi-Z)

⚠ CAUTION

- To prevent any unexpected activation of the status flag FSy in PWM_FSR, the FMODy bit can be set to '1' only if the FPOLy bit has been previously configured to its final value.
- To prevent any unexpected activation of the Fault Protection on the channel x, the bit FPEx[y] can be set to '1' only if the FPOLy bit has been previously configured to its final value.

If a comparison unit is enabled (see PWM Comparison Units) and if a fault is triggered in the channel 0, then the comparison cannot match.

As soon as the fault protection is triggered on a channel, an interrupt (different from the interrupt generated at the end of the PWM period) can be generated but only if it is enabled and not masked. The interrupt is reset by reading the interrupt status register, even if the fault which has caused the trigger of the fault protection is kept active.

51.7.26 PWM Fault Clear Register

Name: PWM_FCR
Offset: 0x64
Reset: –
Property: Write-only

See Fault Inputs for details on fault generation.

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
	FCLR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 7:0 – FCLR[7:0] Fault Clear
For each bit y of FCLR, where y is the fault input number:

0: No effect.

1: If bit y of FMOD field is set to ‘1’ and if the fault input y is not at the level defined by the bit y of FPOL field, the fault y is cleared and becomes inactive (FMOD and FPOL fields belong to PWM Fault Mode Register), else writing this bit to ‘1’ has no effect.

Moreover, when a DMA channel is connected to the AFEC, a resolution lower than 16 bits sets the transfer request size to 16 bits.

Note: If ADTRG is asynchronous to the AFEC peripheral clock, the internal resynchronization introduces a jitter of 1 peripheral clock. This jitter may reduce the resolution of the converted signal. Refer to the formula below, where f_{IN} is the frequency of the analog signal to convert and t_J is the half-period of 1 peripheral clock.

$$\mathrm{SNR} = 20 \times \log 10\left(\frac{1}{2\pi f_{\mathrm{IN}} t_J}\right)$$

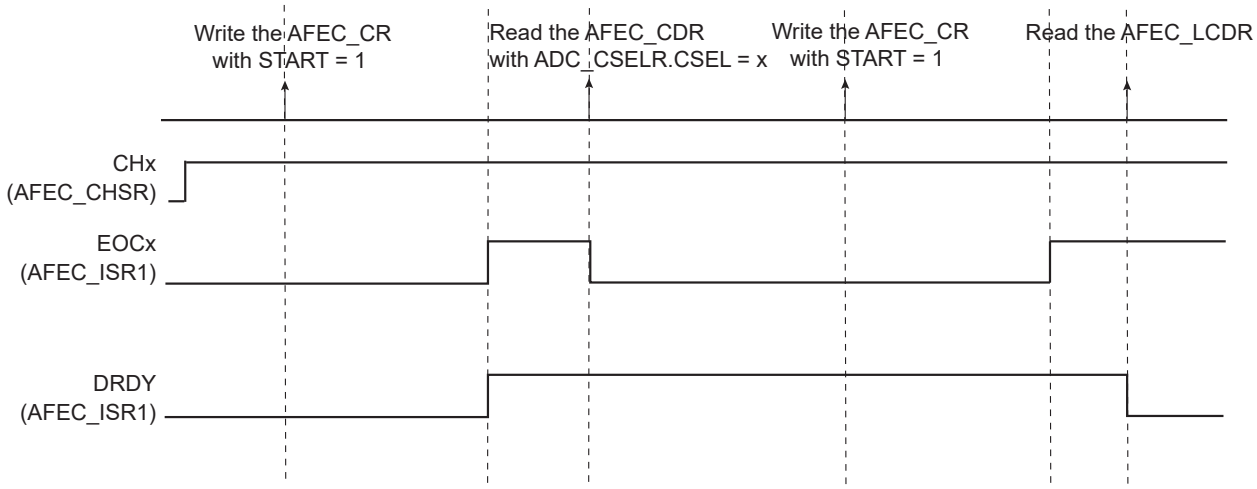
52.6.4 Conversion Results

When a conversion is completed, the resulting 12-bit digital value is stored in an internal register (one register for each channel) that can be read by means of the Channel Data Register (AFEC_CDR) and the Last Converted Data Register (AFEC_LCDR). By setting the bit TAG in the AFEC_EMR, the AFEC_LCDR presents the channel number associated with the last converted data in the CHNB field.

The bits EOCx, where 'x' corresponds to the value programmed in the CSEL bit of AFEC_CSELR, and DRDY in the Interrupt Status Register (AFEC_ISR) are set. In the case of a connected DMA channel, DRDY rising triggers a data transfer request. In any case, either EOCx or DRDY can trigger an interrupt.

Reading the AFEC_CDR clears the EOCx bit. Reading AFEC_LCDR clears the DRDY bit and the EOCx bit corresponding to the last converted channel.

Figure 52-4. EOCx and DRDY Flag Behavior



If AFEC_CDR is not read before further incoming data is converted, the corresponding OVREx flag is set in the Overrun Status Register (AFEC_OVER).

New data converted when DRDY is high sets the GOVRE bit in AFEC_ISR.

The OVREx flag is automatically cleared when AFEC_OVER is read, and the GOVRE flag is automatically cleared when AFEC_ISR is read.

The AFEC can apply different gain and offset on each channel.

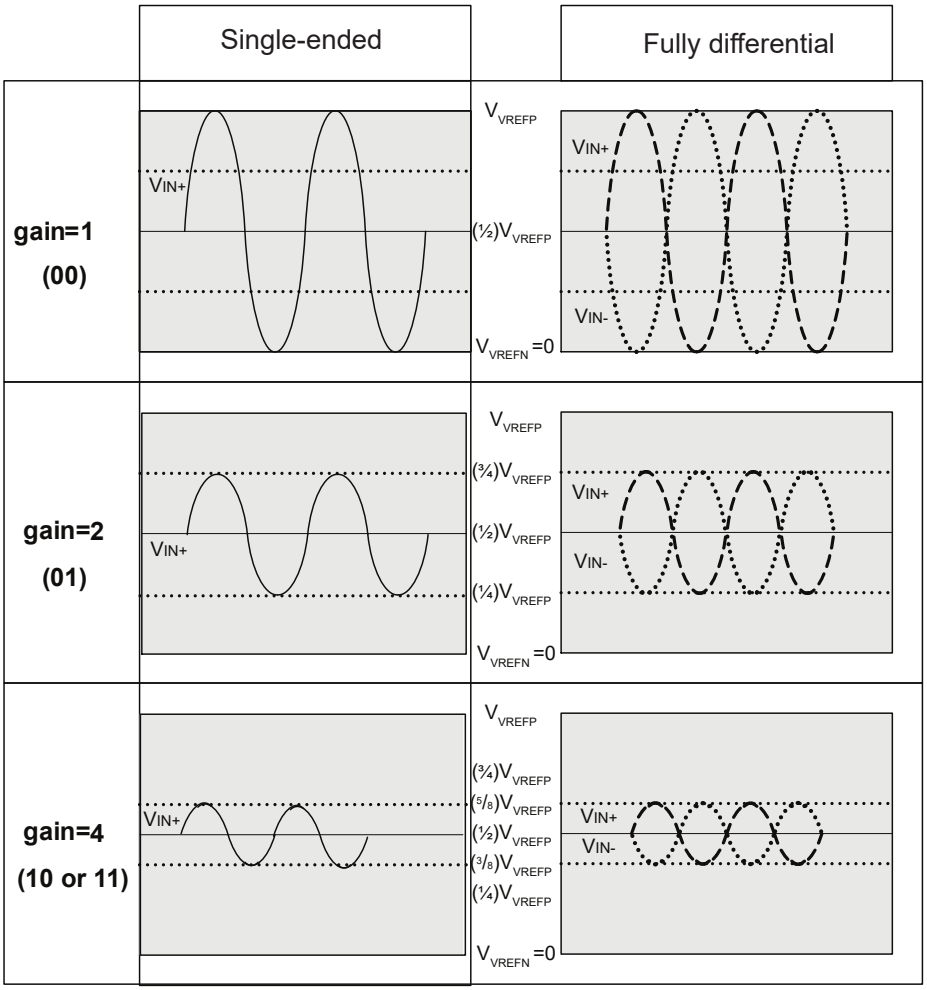
The gain is configured in the GAIN field of the Channel Gain Register (AFEC_CGR) as shown in the following table.

Table 52-5. Gain of the Sample-and-Hold Unit

GAIN	GAIN (DIFFx = 0)	GAIN (DIFFx = 1)
0	1	1
1	2	2
2	4	4
3	4	4

The analog offset of the AFE is configured in the AOFF field in the Channel Offset Compensation register (AFEC_COCR). The offset is only available in Single-ended mode. The field AOFF must be configured to 512 (mid scale of the DAC) when there is no offset error to compensate.To compensate for an offset error of n LSB (positive or negative), the field AOFF must be configured to 512 + n.

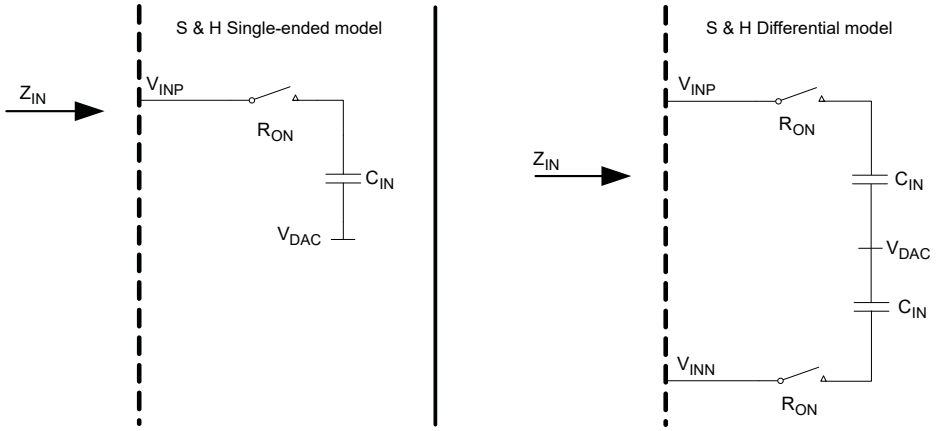
Figure 52-7. Analog Full Scale Ranges in Single-Ended/Differential Applications Versus Gain



Symbol	Parameter	Conditions	Min	Typ(1)	Max	Unit
		Gain=2	-0.3	0.3	1.4	
		Gain=4	-0.3	0.7	3.3	
Single-Ended Mode						
E_O	Single-ended Offset Error (see **Note 1**)	Gain=1	-20	–	35	LSB
E_G	Single-ended Gain Error	Gain=1	0.3	0.7	1.8	%
		Gain=2	0.3	1.3	3.6	
		Gain=4	0.3	1.7	4.7	

58.8.6 AFE Channel Input Impedance

Figure 58-15. Input Channel Model



where:

- Z_{IN} is input impedance in Single-ended or Differential mode
- C_{IN} = 2 to 8 pF ±20% depending on the gain value and mode (SE or DIFF); temperature dependency is negligible
- R_{ON} is typical 2 kΩ and 8 kΩ max (worst case process and high temperature)

The following formula is used to calculate input impedance:

$$Z_{\mathrm{IN}} = \frac{1}{f_S \times C_{\mathrm{IN}}}$$

where:

- f_S is the sampling frequency of the AFE channel
- Typ values are used to compute AFE input impedance Z_{IN}

Table 58-37. Input Capacitance (C_{IN}) Values

Gain Selection	Single-ended	Differential	Unit
1	2	2	pF
2	4	4	
4	8	8	

Table 59-12. Worst Case Power Consumption for Backup Mode with 1 Kbyte BACKUP SRAM Off

Total Consumption	Worst Case Value			Unit
	at 25°C	at 85°C	at 105°C	
Conditions	AMP1	AMP1	AMP1	
V_{DDIO} = 3.6V	5.1	16.4	24	µA
V_{DDIO} = 3.3V	3.7	14.8	23	µA
V_{DDIO} = 3.0V	3.4	13.2	22	µA
V_{DDIO} = 2.5V	2.7	12.8	21	µA
V_{DDIO} = 1.7V	1.3	9.6	18	µA

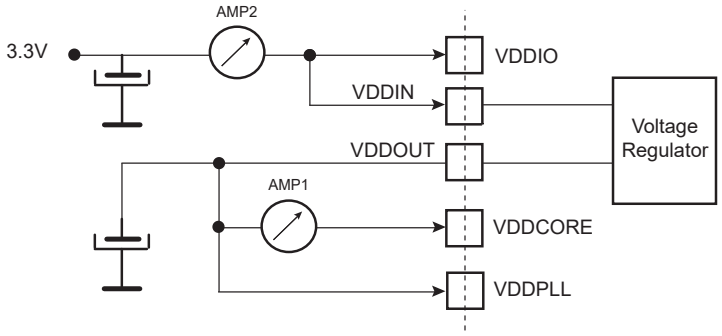
59.3.2 Sleep Mode Current Consumption and Wakeup Time

The Sleep mode configuration and measurements are defined as follows:

- Core clock OFF
- V_{DDIO} = V_{DDIN} = 3.3V
- Master Clock (MCK) running at various frequencies with PLLA or the fast RC oscillator
- Fast startup through WKUP0–13 pins
- Current measurement as shown below and associated wakeup time(1)
- All peripheral clocks deactivated
- T_A = 25°C

  Note: 1. Wakeup time is defined as the delay between the WKUP event and the execution of the first instruction.

Figure 59-6. Measurement Setup for Sleep Mode



The following tables give current consumption and wakeup time in Sleep mode.

Table 59-13. Typical Sleep Mode Current Consumption vs. Master Clock (MCK) Variation with PLLA

Core Clock/MCK (MHz)	VDDCORE Consumption (AMP1)	Total Consumption (AMP2)	Unit	Wakeup Time	Unit
300/150	20	24	mA	0.85	µs
250/125	17	20		1.05	
150/150	20	24		0.9	