Microchip Technology - ATSAMS70Q21B-CN Datasheet

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	EBI/EMI, 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	114
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.08V ~ 3.6V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LFBGA
Supplier Device Package	144-LFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70q21b-cn

26.4.5.3 RSTC Mode Register

Name:	RSTC_MR
Offset:	0x08
Reset:	0x00000001
Property:	Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ERSTL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				URSTIEN				URSTEN
Access				R/W				R/W
Reset				0				1

Bits 31:24 – KEY[7:0] Write Access Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bits 11:8 – ERSTL[3:0] External Reset Length
This field defines the external reset length. The external reset is asserted during a time of $2^{(ERSTL+1)}$ SLCK cycles. This allows assertion duration to be programmed between 60 µs and 2 seconds. Note that synchronization cycles must also be considered when calculating the actual reset length as previously described.

Bit 4 – URSTIEN User Reset Interrupt Enable

Value	Description
0	RSTC_SR.USRTS at ‘1’ has no effect on the RSTC interrupt line.
1	RSTC_SR.USRTS at ‘1’ asserts the RSTC interrupt line if URSTEN = 0.

Bit 0 – URSTEN User Reset Enable

32.6.1.45 PIO Lock Status Register

Name: PIO_LOCKSR
Offset: 0x00E0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access								
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 – P PIO Lock Status

Value	Description
0	The I/O line is not locked.
1	The I/O line is locked.

32.6.1.51 PIO Parallel Capture Interrupt Enable Register

Name: PIO_PCIER
Offset: 0x0154
Property: Write-only

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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RXBUFF	ENDRX	OVRE	DRDY
Access								
Reset								

Bit 3 – RXBUFF Reception Buffer Full Interrupt Enable

Bit 2 – ENDRX End of Reception Transfer Interrupt Enable

Bit 1 – OVRE Parallel Capture Mode Overrun Error Interrupt Enable

Bit 0 – DRDY Parallel Capture Mode Data Ready Interrupt Enable

Offset	Name	Bit Pos.								
		31:24								
0x0604	XDMAC_CDUS22	7:0	DUBS[7:0]							
		15:8	DUBS[15:8]							
		23:16	DUBS[23:16]							
		31:24								
0x0608 ... 0x060F	Reserved									
0x0610	XDMAC_CIE23	7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
		15:8								
		23:16								
		31:24								
0x0614	XDMAC_CID23	7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
		15:8								
		23:16								
		31:24								
0x0618	XDMAC_CIM23	7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
		15:8								
		23:16								
		31:24								
0x061C	XDMAC_CIS23	7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
		15:8								
		23:16								
		31:24								
0x0620	XDMAC_CSA23	7:0	SA[7:0]							
		15:8	SA[15:8]							
		23:16	SA[23:16]							
		31:24	SA[31:24]							
0x0624	XDMAC_CDA23	7:0	DA[7:0]							
		15:8	DA[15:8]							
		23:16	DA[23:16]							
		31:24	DA[31:24]							
0x0628	XDMAC_CNDA23	7:0	NDA[5:0]							NDAIF
		15:8	NDA[13:6]							
		23:16	NDA[21:14]							
		31:24	NDA[29:22]							
0x062C	XDMAC_CNDC23	7:0				NDVIEW[1:0]		NDDUP	NDSUP	NDE
		15:8								
		23:16								
		31:24								
0x0630	XDMAC_CUBC23	7:0	UBLEN[7:0]							
		15:8	UBLEN[15:8]							
		23:16	UBLEN[23:16]							
		31:24								
0x0634	XDMAC_CBC23	7:0	BLEN[7:0]							
		15:8					BLEN[11:8]			

Timer Adjustment

The amount by which the timer increments each clock cycle is controlled by the Timer Increment register (GMAC_TI). Bits [7:0] are the default increment value in nanoseconds. Additional 16 bits of sub-nanosecond resolution are available using the Timer Increment Sub-Nanoseconds register (GMAC_TISUBN). If the rest of the register is written with zero, the timer increments by the value in [7:0], plus the value of the GMAC_TISUBN for each clock cycle.

The GMAC_TISUBN allows a resolution of approximately 15fs.

Bits [15:8] of the increment register are the alternative increment value in nanoseconds, and bits [23:16] are the number of increments after which the alternative increment value is used. If [23:16] are zero the alternative increment value will never be used.

Taking the example of 10.2MHz, there are 102 cycles every 10µs or 51 cycles every 5µs. So a timer with a 10.2MHz clock source is constructed by incrementing by 98ns for fifty cycles and then incrementing by 100ns (98ns × 50 + 100ns = 5000ns). This is programmed by writing the value 0x00326462 to the Timer Increment register (GMAC_TI).

In a second example, a 49.8 MHz clock source requires 20ns for 248 cycles, followed by an increment of 40ns (20ns × 248 + 40ns = 5000ns). This is programmed by writing the value 0x00F82814 to the GMAC_TI register.

The Number of Increments bit field in the GMAC_TI register is 8 bit in size, so frequencies up to 50MHz are supported with 200kHz resolution.

Without the alternative increment field the period of the clock would be limited to an integer number of nanoseconds, resulting in supported clock frequencies of 8, 10, 20, 25, 40, 50, 100, 125, 200 and 250 MHz.

There are eight additional 80-bit registers that capture the time at which PTP event frames are transmitted and received. An interrupt is issued when these registers are updated. The TSU timer count value can be compared to a programmable comparison value. For the comparison, the 48 bits of the seconds value and the upper 22 bits of the nanoseconds value are used. A signal (GTSUCOMP) is output from the core to indicate when the TSU timer count value is equal to the comparison value stored in the TSU timer comparison value registers (GMAC_NSC, GMAC_SCL, and GMAC_SCH). The GTSUCOMP signal can be routed to the Timer peripheral to automatically toggle pin TIOA11/PD21. This can be used as the reference clock for an external PLL to regenerate the audio clock in Ethernet AVB.An interrupt can also be generated (if enabled) when the TSU timer count value and comparison value are equal, mapped to bit 29 of the interrupt status register.

38.6.16 MAC 802.3 Pause Frame Support

Note: See Clause 31, and Annex 31A and 31B of the IEEE standard 802.3 for a full description of MAC 802.3 pause operation.

The following table shows the start of a MAC 802.3 pause frame.

Table 38-14. Start of an 802.3 Pause Frame

Address		Type (MAC Control Frame)	Pause	
Destination	Source		Opcode	Time
0x0180C2000001	6 bytes	0x8808	0x0001	2 bytes

- For a single bank, a NYET handshake is always sent to the host (on Bulk-out transaction) to indicate that the current packet is acknowledged but there is no room for the next one.
- For a double bank, the USBHS responds to the OUT/DATA transaction with an ACK handshake when the endpoint accepted the data successfully and has room for another data payload (the second bank is free).

39.5.2.14 Underflow

This error only exists for isochronous IN/OUT endpoints. It sets the Underflow Interrupt (USBHS_DEVEPTISRx.UNDERFI) bit, which triggers a PEP_x interrupt if the Underflow Interrupt Enable (USBHS_DEVEPTIMRx.UNDERFE) bit is one.

- An underflow can occur during the IN stage if the host attempts to read from an empty bank. A zero-length packet is then automatically sent by the USBHS.
- An underflow cannot occur during the OUT stage on a CPU action, since the user may only read if the bank is not empty (USBHS_DEVEPTISRx.RXOUTI = 1 or USBHS_DEVEPTISRx.RWALL = 1).
- An underflow can also occur during the OUT stage if the host sends a packet while the bank is already full. Typically, the CPU is not fast enough. The packet is lost.
- An underflow cannot occur during the IN stage on a CPU action, since the user may only write if the bank is not full (USBHS_DEVEPTISRx.TXINI = 1or USBHS_DEVEPTISRx.RWALL = 1).

39.5.2.15 Overflow

This error exists for all endpoint types. It sets the Overflow interrupt (USBHS_DEVEPTISRx.OVERFI) bit, which triggers a PEP_x interrupt if the Overflow Interrupt Enable (USBHS_DEVEPTIMRx.OVERFE) bit is one.

- An overflow can occur during the OUT stage if the host attempts to write into a bank which is too small for the packet. The packet is acknowledged and the USBHS_DEVEPTISRx.RXOUTI bit is set as if no overflow had occurred. The bank is filled with all the first bytes of the packet that fit in.
- An overflow cannot occur during the IN stage on a CPU action, since the user may only write if the bank is not full (USBHS_DEVEPTISRx.TXINI = 1 or USBHS_DEVEPTISRx.RWALL = 1).

39.5.2.16 HB IsoIn Error

This error only exists for high-bandwidth isochronous IN endpoints.

At the end of the microframe, if at least one packet has been sent to the host and fewer banks than expected have been validated (by clearing the USBHS_DEVEPTIMRx.USBHS_DEVEPTIMRx.FIFOCON) for this microframe, it sets the USBHS_DEVEPTISRx.HBISOINERRORI bit, which triggers a PEP_x interrupt if the High Bandwidth Isochronous IN Error Interrupt Enable (HBISOINERRORE) bit is one.

For example, if the Number of Transactions per MicroFrame for Isochronous Endpoint (NBTRANS) field in USBHS_DEVEPTCFGx is three (three transactions per microframe), only two banks are filled by the CPU (three expected) for the current microframe. Then, the HBISOINERRI interrupt is generated at the end of the microframe. Note that an UNDERFI interrupt is also generated (with an automatic zero-length-packet), except in the case of a missing IN token.

39.5.2.17 HB IsoFlush

This error only exists for high-bandwidth isochronous IN endpoints.

At the end of the microframe, if at least one packet has been sent to the host and there is a missing IN token during this microframe, the bank(s) destined to this microframe is/are flushed out to ensure a good data synchronization between the host and the device.

For example, if NBTRANS is three (three transactions per microframe) and if only the first IN token (among three) is well received by the USBHS, the last two banks are discarded.

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								

Bit 5 – OVERFIES Overflow Interrupt Enable

Bit 4 – NAKEDES NAKed Interrupt Enable

Bit 3 – PERRES Pipe Error Interrupt Enable

Bit 2 – UNDERFIES Underflow Interrupt Enable

Bit 1 – TXOUTES Transmitted OUT Data Interrupt Enable

Bit 0 – RXINES Received IN Data Interrupt Enable

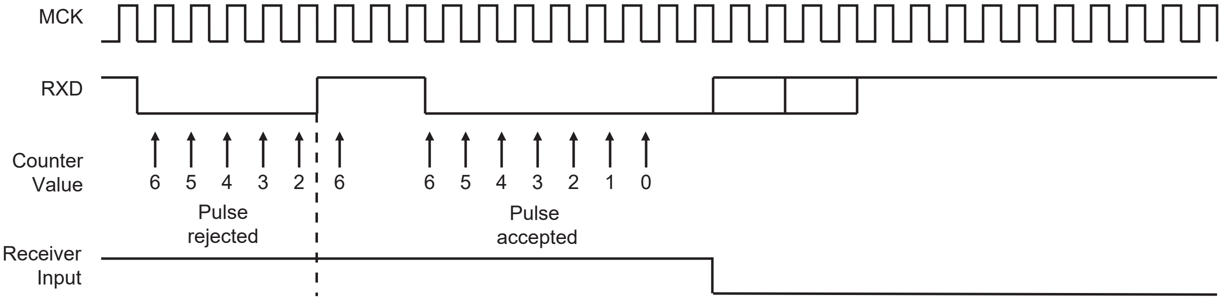
Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (µs)
20,000,000	9,600	130	0.16%	19.53
32,768,000	9,600	213	0.16%	19.53
40,000,000	9,600	260	0.16%	19.53
3,686,400	2,400	96	0.00%	78.13
20,000,000	2,400	521	0.03%	78.13
32,768,000	2,400	853	0.04%	78.13

46.6.5.3 IrDA Demodulator

The demodulator is based on the IrDA Receive filter comprised of an 8-bit down counter which is loaded with the value programmed in US_IF. When a falling edge is detected on the RXD pin, the Filter Counter starts counting down at the peripheral clock speed. If a rising edge is detected on the RXD pin, the counter stops and is reloaded with US_IF. If no rising edge is detected when the counter reaches 0, the input of the receiver is driven low during one bit time.

The following figure illustrates the operations of the IrDA demodulator.

Figure 46-34. IrDA Demodulator Operations



The programmed value in the US_IF register must always meet the following criterion:

$t_{\text{peripheral clock}} \times (\text{IRDA\_FILTER} + 3) < 1.41\ \mu s$

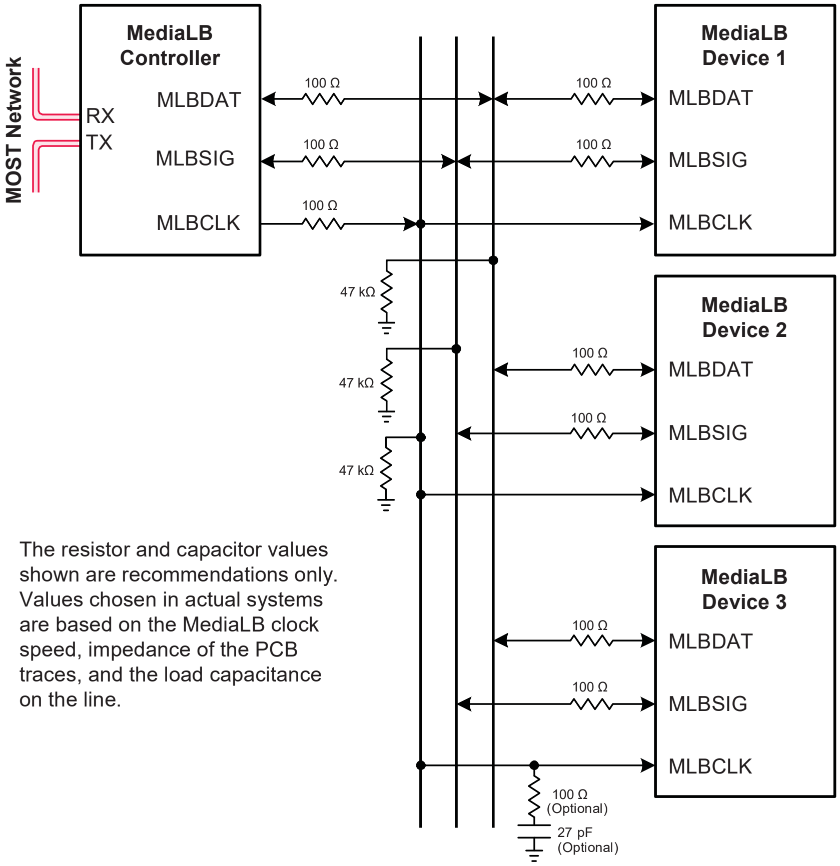
As the IrDA mode uses the same logic as the ISO7816, note that the FI_DI_RATIO field in US_FIDI must be set to a value higher than 0 in order to ensure IrDA communications operate correctly.

46.6.6 RS485 Mode

The USART features the RS485 mode to enable line driver control. While operating in RS485 mode, the USART behaves as though in Asynchronous or Synchronous mode and configuration of all the parameters is possible. The difference is that the RTS pin is driven high when the transmitter is operating. The behavior of the RTS pin is controlled by the TXEMPTY bit. A typical connection of the USART to an RS485 bus is shown in Figure 46-35.

optionally have AC-parallel termination near the farthest Device from the Controller to ensure a clean clock by minimizing reflections.

Figure 48-2. 3-pin MediaLB Connection Diagram



The resistor and capacitor values shown are recommendations only. Values chosen in actual systems are based on the MediaLB clock speed, impedance of the PCB traces, and the load capacitance on the line.

48.6 Functional Description

48.6.1 Link Layer

The MediaLB link layer uses the concept of ChannelAddress, Command, RxStatus, and Data to transport all MOST Network data types and manage MediaLB.

These terms are defined as follows:

- ChannelAddress:

A 16-bit token, which is sent on the MLBS line by the MediaLB Controller at the end of a physical channel. A unique ChannelAddress defines a logical channel and grants a particular physical channel to a transmitting (Tx) and a receiving (Rx) MediaLB Device.

- Command:

A byte-wide value sent by the transmitting (Tx) MediaLB Device on the MLBS line at the start of a physical channel. This command byte indicates the data type and additional control information to the Rx MediaLB Device. The Tx Device also outputs data on the MLBD signal during the same physical channel that Command is sent.

- RxStatus:

- Finish any active MLB transfer
- Disable MLB (clear the MLBEN and MLBPEN bits in MLB_MLBC0)
- Disable HBI (clear all bits in MLB_HCMR0 and MLB_HCMR1, clear EN bit in MLB_HCTL)
- Mask AHB interrupts (clear all bits in MLB_ACMR0 and MLB_ACMR1)

For information on configuring the MLB IP if the clocks are re-enabled, see Section “Configure the Hardware”.

49. Controller Area Network (MCAN)

49.1 Description

The Controller Area Network (MCAN) performs communication according to ISO 11898-1:2015 and to Bosch CAN-FD specification. Additional transceiver hardware is required for connection to the physical layer.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN core to the Message RAM, as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN core, as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 128 filter elements, where each element can be configured as a range, as a bit mask, or as a dedicated ID filter.

49.2 Embedded Characteristics

- Compliant with CAN Protocol Version 2.0 Part A, B and ISO 11898-1
- CAN-FD with up to 64 Data Bytes Supported
- CAN Error Logging
- AUTOSAR Optimized
- SAE J1939 Optimized
- Improved Acceptance Filtering
- Two Configurable Receive FIFOs
- Separate Signalling on Reception of High Priority Messages
- Up to 64 Dedicated Receive Buffers
- Up to 32 Dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM Access for Processor
- Multiple MCANs May Share the Same Message RAM
- Programmable Loop-back Test Mode
- Maskable Module Interrupts
- Support for Asynchronous CAN and System Bus Clocks
- Power-down Support
- Debug on CAN Support

49.5.6 FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index in the registers MCAN_RXF0A, MCAN_RXF1A and MCAN_TXEFA. Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO’s Get Index.

Due to the fact that the processor has free access to the MCAN’s Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO’s Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO’s Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN does not check for erroneous values.

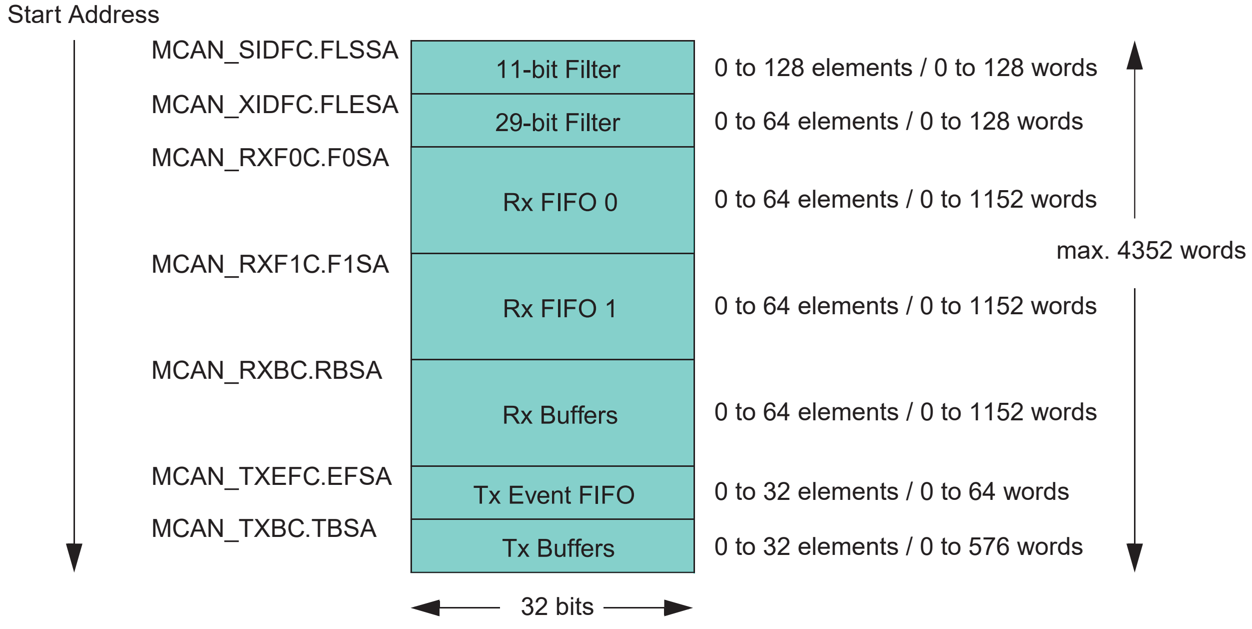
49.5.7 Message RAM

49.5.7.1 Message RAM Configuration

The Message RAM has a width of 32 bits. The MCAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in the figure below, nor is there any restriction with respect to the sequence of the sections.

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

Figure 49-12. Message RAM Configuration


Start Address
MCAN_SIDFC.FLSSA
MCAN_XIDFC.FLESA
MCAN_RXF0C.F0SA
MCAN_RXF1C.F1SA
MCAN_RXBC.RBSA
MCAN_TXEFC.EFSA
MCAN_TXBC.TBSA
11-bit Filter
29-bit Filter
Rx FIFO 0
Rx FIFO 1
Rx Buffers
Tx Event FIFO
Tx Buffers
0 to 128 elements / 0 to 128 words
0 to 64 elements / 0 to 128 words
0 to 64 elements / 0 to 1152 words
0 to 64 elements / 0 to 1152 words
0 to 64 elements / 0 to 1152 words
0 to 32 elements / 0 to 64 words
0 to 32 elements / 0 to 576 words
max. 4352 words
32 bits

50. Timer Counter (TC)

50.1 Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multipurpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

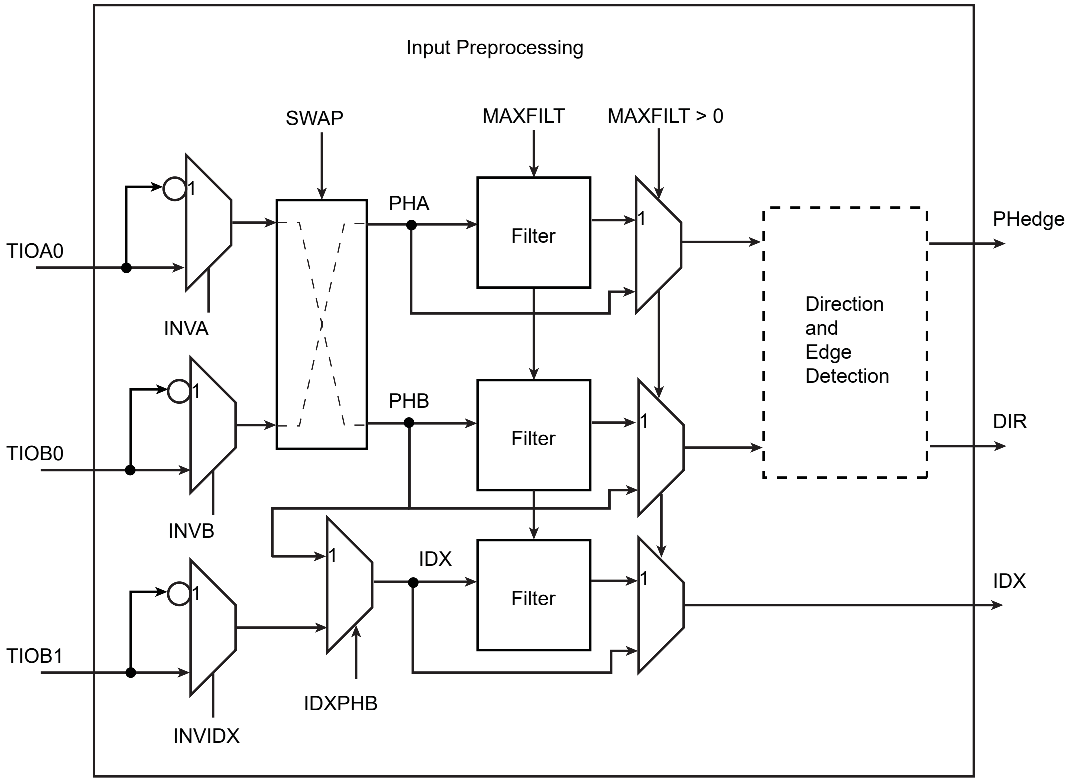
The TC block has two global registers which act upon all TC channels:

- Block Control register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

50.2 Embedded Characteristics

- Total of 12 Channels
- 16-bit Channel Size
- Wide Range of Functions Including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder
  - 2-bit Gray up/down count for stepper motor
- Each Channel is User-Configurable and Contains:
  - Three external clock inputs
  - Five Internal clock inputs
  - Two multipurpose input/output signals acting as trigger event
  - Trigger/capture events can be directly synchronized by PWM signals
- Internal Interrupt Signal

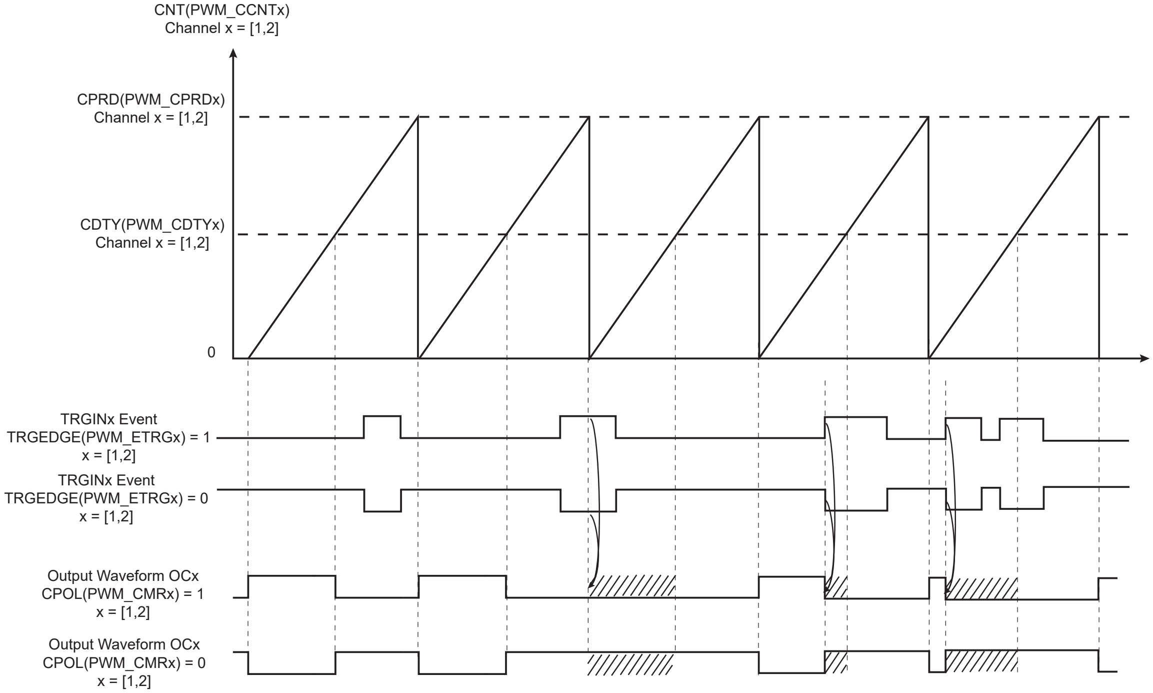
Figure 50-18. Input Stage



Input filtering can efficiently remove spurious pulses that might be generated by the presence of particulate contamination on the optical or magnetic disk of the rotary sensor.

Spurious pulses can also occur in environments with high levels of electromagnetic interference. Or, simply if vibration occurs even when rotation is fully stopped and the shaft of the motor is in such a position that the beginning of one of the reflective or magnetic bars on the rotary sensor disk is aligned with the light or magnetic (Hall) receiver cell of the rotary sensor. Any vibration can make the PHA, PHB signals toggle for a short duration.

Figure 51-31. Cycle-By-Cycle Duty Mode



51.6.5.3.2 Application Example

The figure below illustrates an application example of the Cycle-by-cycle Duty mode.

In an LED string control circuit, Cycle-by-cycle Duty mode can be used to automatically limit the current in the LED string.

57.5.5 AES Interrupt Mask Register

Name: AES_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

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

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TAGRDY
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 16 – TAGRDY GCM Tag Ready Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

Symbol	Parameter	Digital Code	Min	Typ	Max	Unit
		1110	–	3.28	–	
		1111	–	3.4	–	

Figure 58-3. VDDIO Supply Monitor

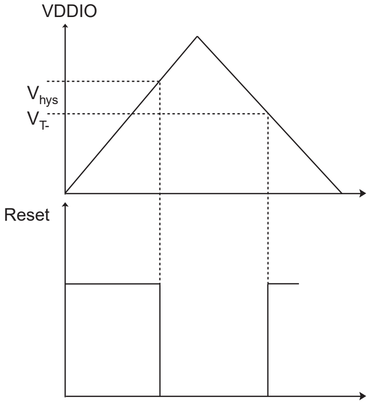
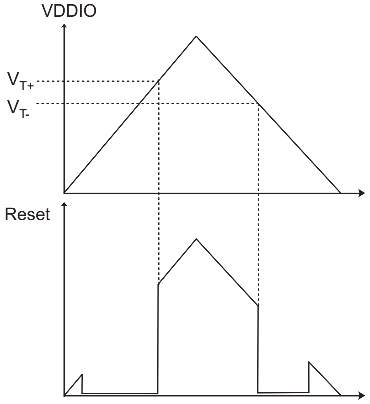


Table 58-10. VDDIO Power-On Reset Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{T+}	Threshold Voltage Rising	–	1.45	1.53	1.61	V
V_{T-}	Threshold Voltage Falling	–	1.37	1.46	–	V
V_{hys}	Hysteresis	–	40	80	130	mV
t_{RES}	Reset Time-out Period	–	240	320	800	μs

Figure 58-4. VDDIO Power-On Reset Characteristics



58.3 Power Consumption

- Power consumption of the device depending on the different Low-Power modes (Backup, Wait, Sleep) and Active mode
- Power consumption on power supply in different modes: Backup, Wait, Sleep and Active
- Power consumption by peripheral:

Table 59-55. QSPI Timings

Symbol	Parameter	Conditions	Min	Max	Unit
QSPI$_0$	QIOx data in to QSCK rising edge (input setup time)	3.3V domain	2.5	–	ns
		1.8V domain	2.9	–	ns
QSPI$_1$	QIOx data in to QSCK rising edge (input hold time)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
QSPI$_2$	QSCK rising edge to QIOx data out valid	3.3V domain	-1.3	1.9	ns
		1.8V domain	-2.5	3.0	ns
QSPI$_3$	QIOx data in to QSCK falling edge (input setup time)	3.3V domain	2.9	–	ns
		1.8V domain	3.2	–	ns
QSPI$_4$	QIOx data in to QSCK falling edge(input hold time)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
QSPI$_5$	QSCK falling edge to QIOx data out valid	3.3V domain	-1.6	1.8	ns
		1.8V domain	-2.7	3.1	ns

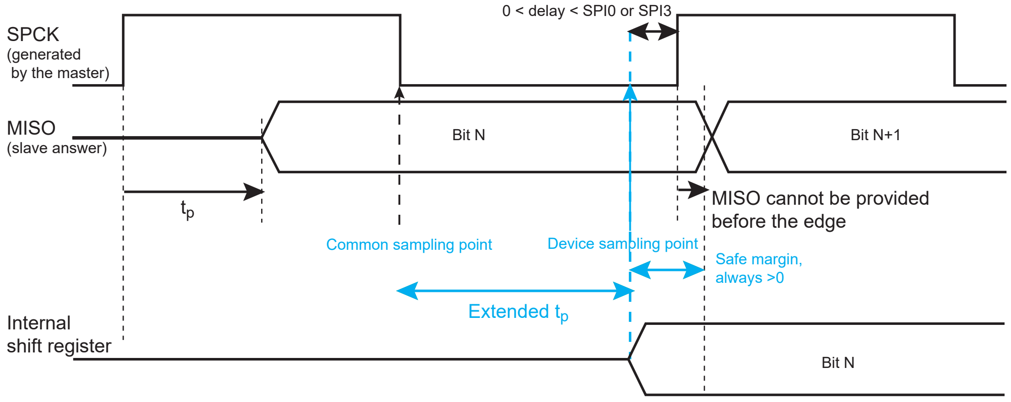
Timings are given for the 3.3V domain, with VDDIO from 2.85V to 3.6V, maximum external capacitor = 40 pF.

59.13.1.6 SPI Characteristics

In the figures below, the MOSI line shifting edge is represented with a hold time equal to 0. However, it is important to note that for this device, the MISO line is sampled prior to the MOSI line shifting edge. As shown further below, the device sampling point extends the propagation delay (t_p) for slave and routing delays to more than half the SPI clock period, whereas the common sampling point allows only less than half the SPI clock period.

As an example, an SPI Slave working in Mode 0 can be safely driven if the SPI Master is configured in Mode 0.

Figure 59-19. MISO Capture in Master Mode



Symbol	Parameter	Condition	Min	Max	Unit
V_{hys}	Hysteresis of Schmitt Trigger Inputs	–	0.150	–	V
V_{OL}	Low-level Output Voltage	3 mA sink current	–	0.4	V
t_R	Rise Time for both TWD and TWCK		$20 + 0.1C_b$(1)(2)	300	ns
t_{OF}	Output Fall Time from V_{IHmin} to V_{ILmax}	10 pF < C_b < 400 pF see the figure below	$20 + 0.1C_b$(1)(2)	250	ns
C_i(1)	Capacitance for each I/O Pin	–	–	10	pF
f_{TWCK}	TWCK Clock Frequency	–	0	400	kHz
R_P	Value of Pull-up resistor	f_{TWCK} ≤ 100 kHz	(V_{DDIO} - 0.4V) ÷ 3mA	1000ns ÷ C_b	Ω
		f_{TWCK} > 100 kHz		300ns ÷ C_b	Ω
t_{LOW}	Low Period of the TWCK clock	f_{TWCK} ≤ 100 kHz	(3)	–	µs
		f_{TWCK} > 100 kHz	(3)	–	µs
t_{HIGH}	High period of the TWCK clock	f_{TWCK} ≤ 100 kHz	(4)	–	µs
		f_{TWCK} > 100 kHz	(4)	–	µs
$t_{HD;STA}$	Hold Time (repeated) START Condition	f_{TWCK} ≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs
$t_{SU;STA}$	Set-up time for a repeated START condition	f_{TWCK} ≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs
$t_{HD;DAT}$	Data hold time	f_{TWCK} ≤ 100 kHz	0	3 × t_{CPMCK}(5)	µs
		f_{TWCK} > 100 kHz	0	3 ×t_{CPMCK}(5)	µs
$t_{SU;DAT}$	Data setup time	f_{TWCK} ≤ 100 kHz	t_{LOW} - 3 × t_{CPMCK}(5)	–	ns
		f_{TWCK} > 100 kHz	t_{LOW} - 3 × t_{CPMCK}(5)	–	ns
$t_{SU;STO}$	Setup time for STOP condition	f_{TWCK} ≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs
$t_{HD;STA}$	Hold Time (repeated) START Condition	f_{TWCK} ≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs

Note:

1. Required only for f_{TWCK} > 100 kHz.
2. C_b = capacitance of one bus line in pF. Per I^2C standard, C_b max = 400pF.
3. The TWCK low period is defined as follows: $t_{LOW} = ((CLDIV \times 2^{CKDIV}) + 4) \times t_{MCK}$.
4. The TWCK high period is defined as follows: $t_{HIGH} = ((CHDIV \times 2^{CKDIV}) + 4) \times t_{MCK}$.