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	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	144-UFBGA
Supplier Device Package	144-UFBGA (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70q21b-cfn

19.4.10 SMC NAND Flash Chip Select Configuration Register

Name: CCFG_SMCNFCS
Offset: 0x0124
Reset: 0x00000000
Property: Read/Write

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
				SDRAMEN	SMC_NFCS3	SMC_NFCS2	SMC_NFCS1	SMC_NFCS0
Access								
Reset				0	0	0	0	0

Bit 4 – SDRAMEN SDRAM Enable

WARNING This bit must not be used if SMC_NFCS1 is set.

WARNING: This must not be used if SMC_NFCS1 is set.

Value	Description
0	NCS1 is not assigned to SDRAM.
1	NCS1 is assigned to SDRAM.

Bit 3 – SMC_NFCS3 SMC NAND Flash Chip Select 3 Assignment

Value	Description
0	NCS3 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS3).
1	NCS3 is assigned to a NAND Flash (NANDOE and NANWE used for NCS3).

Bit 2 – SMC_NFCS2 SMC NAND Flash Chip Select 2 Assignment

Value	Description
0	NCS2 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS2).
1	NCS2 is assigned to a NAND Flash (NANDOE and NANWE used for NCS2).

Value	Description
0	The minute-matching alarm is disabled.
1	The minute-matching alarm is enabled.

Bits 14:8 – MIN[6:0] Minute Alarm
This field is the alarm field corresponding to the BCD-coded minute counter.

Bit 7 – SECEN Second Alarm Enable

Value	Description
0	The second-matching alarm is disabled.
1	The second-matching alarm is enabled.

Bits 6:0 – SEC[6:0] Second Alarm
This field is the alarm field corresponding to the BCD-coded second counter.

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AHB Error

Bit 5 – RLEX Retry Limit Exceeded

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

38.8.25 GMAC Wake on LAN Register

Name: GMAC_WOL
Offset: 0x0B8
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MTI	SA1	ARP	MAG
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IP[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
	IP[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 19 – MTI Multicast Hash Event Enable

Value	Description
0	Wake on LAN multicast hash Event disabled
1	Wake on LAN multicast hash Event enabled

Bit 18 – SA1 Specific Address Register 1 Event Enable

Value	Description
0	Wake on Specific Address Register 1 Event disabled
1	Wake on Specific Address Register 1 Event enabled

Bit 17 – ARP ARP Request Event Enable

Value	Description
0	Wake on LAN ARP request Event disabled
1	Wake on LAN ARP request Event enabled

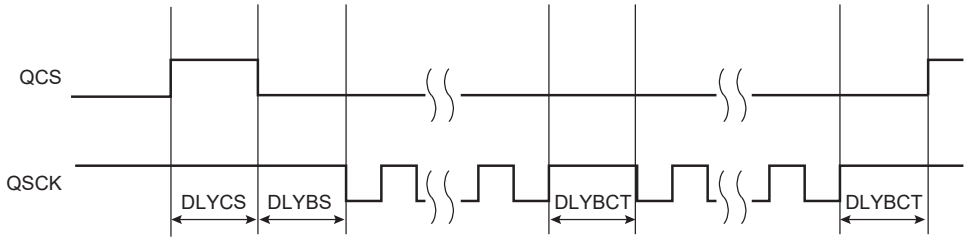
Bit 16 – MAG Magic Packet Event Enable

Offset	Name	Bit Pos.								
0x0180	USBHS_DEVEPTICR8	7:0	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0180	USBHS_DEVEPTICR8 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0184	USBHS_DEVEPTICR9	7:0	SHORTPACKETC	STALLEDIC	OVERFIC	NAKINIC	NAKOUTIC	RXSTPIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0184	USBHS_DEVEPTICR9 (ISOENPT)	7:0	SHORTPACKETC	CRCERRIC	OVERFIC	HBISOFLUSHIC	HBISOINERRIC	UNDERFIC	RXOUTIC	TXINIC
		15:8								
		23:16								
		31:24								
0x0188 ... 0x018F	Reserved									
0x0190	USBHS_DEVEPTIFR0	7:0	SHORTPACKETS	STALLEDIS	OVERFIS	NAKINIS	NAKOUTIS	RXSTPIS	RXOUTIS	TXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0190	USBHS_DEVEPTIFR0 (ISOENPT)	7:0	SHORTPACKETS	CRCERRIS	OVERFIS	HBISOFLUSHIS	HBISOINERRIS	UNDERFIS	RXOUTIS	TXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0194	USBHS_DEVEPTIFR1	7:0	SHORTPACKETS	STALLEDIS	OVERFIS	NAKINIS	NAKOUTIS	RXSTPIS	RXOUTIS	TXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0194	USBHS_DEVEPTIFR1 (ISOENPT)	7:0	SHORTPACKETS	CRCERRIS	OVERFIS	HBISOFLUSHIS	HBISOINERRIS	UNDERFIS	RXOUTIS	TXINIS
		15:8				NBUSYBKS				
		23:16								
		31:24								
0x0198	USBHS_DEVEPTIFR2	7:0	SHORTPACKETS	STALLEDIS	OVERFIS	NAKINIS	NAKOUTIS	RXSTPIS	RXOUTIS	TXINIS
		15:8				NBUSYBKS				
		23:16								

- The delay between consecutive transfers, programmed by writing QSPI_MR.DLYBCT. Allows insertion of a delay between two consecutive transfers. In Serial Memory mode, this delay is not programmable and DLYBCT is ignored. In this mode, DLYBCT must be written to ‘0’.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 42-4. Programmable Delays



42.6.4 QSPI SPI Mode

In SPI mode, the QSPI acts as a standard SPI Master.

To activate this mode, QSPI_MR.SMM must be written to ‘0’ in QSPI_MR.

42.6.4.1 SPI Mode Operations

The QSPI in standard SPI mode operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the slave connected to the SPI bus. The QSPI drives the chip select line to the slave (QCS) and the serial clock signal (QSCK).

The QSPI features two holding registers, the Transmit Data register (QSPI_TDR) and the Receive Data register (QSPI_RDR), and a single internal shift register. The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to the QSPI_TDR. The written data is immediately transferred to the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted to the internal shift register. Receiving data cannot occur without transmitting data. If receiving mode is not needed, for example when communicating with a slave receiver only (such as an LCD), the receive status flags in the Status register (QSPI_SR) can be discarded.

If new data is written in QSPI_TDR during the transfer, it is retained there until the current transfer is completed. Then, the received data is transferred from the internal shift register to the QSPI_RDR, the data in QSPI_TDR is loaded in the internal shift register and a new transfer starts.

The transfer of a data written in QSPI_TDR in the internal shift register is indicated by the Transmit Data Register Empty (TDRE) bit in the QSPI_SR. When new data is written in QSPI_TDR, this bit is cleared. QSPI_SR.TDRE is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the TXEMPTY flag in the QSPI_SR. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, QSPI_SR.TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

The transfer of received data from the internal shift register in QSPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in the QSPI_SR. When the received data is read, QSPI_SR.RDRF bit is cleared.

The external data lines can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the microcontroller or the QSPI slave device (e.g., memory).

The scrambling/unscrambling function can be enabled by writing a ‘1’ to the SCREN bit in the QSPI Scrambling Mode Register (QSPI_SMR).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable user scrambling key (field USRK) in the QSPI Scrambling Key Register (QSPI_SKR). QSPI_SKR is only accessible in Write mode.

If QSPI_SMR.RVDIS is written to ‘0’, the scrambling/unscrambling algorithm includes the user scrambling key plus a random value depending on device processing characteristics. Data scrambled by a given microcontroller cannot be unscrambled by another.

If QSPI_SMR.RVDIS is written to ‘1’, the scrambling/unscrambling algorithm includes only the user scrambling key. No random value is part of the key.

The user scrambling key or the seed for key generation must be securely stored in a reliable nonvolatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

42.6.7 Register Write Protection

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

If a write access to a write-protected register is detected, the WPVS flag in the QSPI Write Protection Status Register (QSPI_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 QSPI_WPSR.

The following registers can be write-protected when WPEN is set in QSPI_WPMR:

- QSPI Mode Register
- QSPI Serial Clock Register
- QSPI Scrambling Mode Register
- QSPI Scrambling Key Register

42.7.9 QSPI Serial Clock Register

Name: QSPI_SCR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DLYBS[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	15	14	13	12	11	10	9	8
	SCBR[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
							CPHA	CPOL
Access							R/W	R/W
Reset							0	0

Bits 23:16 – DLYBS[7:0] Delay Before QSCK
This field defines the delay from QCS valid to the first valid QSCK transition.

When DLYBS equals zero, the QCS valid to QSCK transition is 1/2 the QSCK clock period.

Otherwise, the following equation determines the delay:

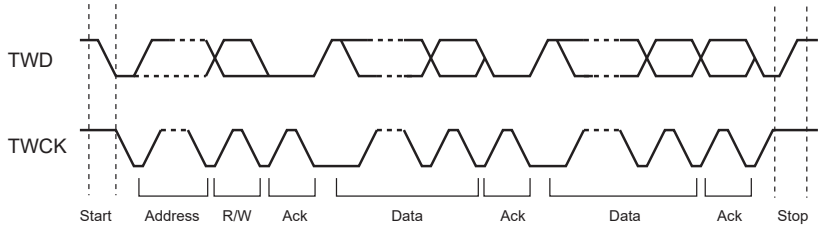
DLYBS = Delay Before QSCK × $f_{peripheral\ clock}$

Bits 15:8 – SCBR[7:0] Serial Clock Baud Rate
The QSPI uses a modulus counter to derive the QSCK baud rate from the peripheral clock. The baud rate is selected by writing a value from 0 to 255 in the SCBR field. The following equation determines the QSCK baud rate:

SCBR = ($f_{peripheral\ clock}$ / QSCK Baudrate) - 1

Bit 1 – CPHA Clock Phase
CPHA determines which edge of QSCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between master and slave devices.

Figure 43-3. Transfer Format



43.6.2 Modes of Operation

The TWIHS has different modes of operation:

- Master Transmitter mode (Standard and Fast modes only)
- Master Receiver mode (Standard and Fast modes only)
- Multimaster Transmitter mode (Standard and Fast modes only)
- Multimaster Receiver mode (Standard and Fast modes only)
- Slave Transmitter mode (Standard, Fast and High-speed modes)
- Slave Receiver mode (Standard, Fast and High-speed modes)

These modes are described in the following sections.

43.6.3 Master Mode

43.6.3.1 Definition

The master is the device that starts a transfer, generates a clock and stops it. This operating mode is not available if High-speed mode is selected.

43.6.3.2 Programming Master Mode

The following registers must be programmed before entering Master mode:

1. TWIHS_MMR.DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access slave devices in Read or Write mode.
2. TWIHS_CWGR.CKDIV + CHDIV + CLDIV: Clock Waveform register
3. TWIHS_CR.SVDIS: Disables the Slave mode
4. TWIHS_CR.MSEN: Enables the Master mode

Note: If the TWIHS is already in Master mode, the device address (DADR) can be configured without disabling the Master mode.

43.6.3.3 Master Transmitter Mode

This operating mode is not available if High-speed mode is selected.

After the master initiates a START condition when writing into the Transmit Holding register (TWIHS_THR), it sends a 7-bit slave address, configured in the Master Mode register (DADR in TWIHS_MMR), to notify the slave device. The bit following the slave address indicates the transfer direction, 0 in this case (MREAD = 0 in TWIHS_MMR).

The TWIHS transfers require the slave to acknowledge each received byte. During the acknowledge clock pulse (9th pulse), the master releases the data line (HIGH), enabling the slave to pull it down in order to generate the acknowledge. If the slave does not acknowledge the byte, then the Not Acknowledge flag (NACK) is set in the TWIHS Status Register (TWIHS_SR) of the master and a STOP condition is sent. The NACK flag must be cleared by reading TWIHS_SR before the next write into TWIHS_THR. As with the other status bits, an interrupt can be generated if enabled in the Interrupt

shapes) occur only during a physical channel (PCn) associated with the logical channel defined by a single ChannelAddress.

The flow diagram contains four states: Idle, Start, Continue, and End. Each state uses a different command when sending the data. The Idle state is the starting point, waiting for the application to initiate a packet transfer. When a quadlet is ready to be transferred, the flow diagram moves to the Start state.

Note: If a ControlEnd command is sent in the physical channel preceding a ReceiverProtocolError RxStatus (in either the Idle or Start state), the ReceiverProtocolError status response must be assigned to the previous packet transmitted. Alternatively, a status response of ReceiverProtocolError (in either the Idle or Start state) must not be assigned to the previous packet transmitted unless ControlEnd was sent in the preceding physical channel.

Once a quadlet has been sent successfully, the flow diagram moves to the Continue state, depicted in Figure 48-7, and stays there until all but the last quadlet has been transmitted. The last quadlet is transmitted in the End state, which is depicted in Figure 48-8.

The protocol flow for an Rx Device is illustrated in Figure 48-9. This flow diagram consists of only two states: Idle and Continue. The Idle state is the starting point where the Rx Device is waiting for a packet start command. Once a start command has been received (ControlStart or AsyncStart), the flow diagram moves to the Continue state. The reception of a ControlEnd command completes the transfer and moves the flow diagram back to the Idle state, where it waits for the next packet.

The protocol flow for an Rx Device, as described in Figure 48-9, should be used as a reference for standard MediaLB Devices. According to this flow, a ReceiverProtocolError status response may be sent by an Rx Device only in the Continue state; however, more enhanced MediaLB Devices can also conduct protocol checks in the Idle state. In this case, a ReceiverProtocolError status response could be sent for example, if a logical channel is setup for control data and an isochronous or synchronous command is received. Protocol checks in the Continue state may be expanded beyond the flow shown in Figure 48-9 when required by specific implementations.

48.7 Register Summary

Offset	Name	Bit Pos.								
0x00	MLB_MLBC0	7:0	MLBLK		ZERO	MLBCLK[2:0]				MLBEN
		15:8	FCNT[0:0]	CTLRETRY		ASYRETRY				
		23:16							FCNT[2:1]	
		31:24								
0x04 ... 0x0B	Reserved									
0x0C	MLB_MS0	7:0	MCS: MediaLB Channel Status [31[7:0]							
		15:8	MCS: MediaLB Channel Status [31[15:8]							
		23:16	MCS: MediaLB Channel Status [31[23:16]							
		31:24	MCS: MediaLB Channel Status [31[31:24]							
0x10 ... 0x13	Reserved									
0x14	MLB_MS1	7:0	MCS: MediaLB Channel Status [63[7:0]							
		15:8	MCS: MediaLB Channel Status [63[15:8]							
		23:16	MCS: MediaLB Channel Status [63[23:16]							
		31:24	MCS: MediaLB Channel Status [63[31:24]							
0x18 ... 0x1F	Reserved									
0x20	MLB_MSS	7:0			SERVREQ	SWSYSCMD	CSSYSCMD	ULKSYSCMD	LKSYSCMD	RSTSYSCMD
		15:8								
		23:16								
		31:24								
0x24	MLB_MSD	7:0	SD0[7:0]							
		15:8	SD1[7:0]							
		23:16	SD2[7:0]							
		31:24	SD3[7:0]							
0x28 ... 0x2B	Reserved									
0x2C	MLB_MIEN	7:0							ISOC_BUFO	ISOC_PE
		15:8								
		23:16		ATX_BREAK	ATX_PE	ATX_DONE	ARX_BREAK	ARX_PE	ARX_DONE	SYNC_PE
		31:24			CTX_BREAK	CTX_PE	CTX_DONE	CRX_BREAK	CRX_PE	CRX_DONE
0x30 ... 0x3B	Reserved									
0x3C	MLB_MLBC1	7:0	CLKM	LOCK						
		15:8	NDA[7:0]							
		23:16								
		31:24								
0x40	Reserved									

49.5.4.1 Acceptance Filtering

The MCAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC)
- Standard ID Filter Configuration (MCAN_SIDFC)
- Extended ID Filter Configuration (MCAN_XIDFC)
- Extended ID and Mask (MCAN_XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag (MCAN_IR.HPM)
- Set High Priority Message interrupt flag (MCAN_IR.HPM) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the effected Rx Buffer or Rx FIFO:

- Rx Buffer
  New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC.
- Rx FIFO
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC. In case the matching Rx FIFO is operated in Overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode have to be considered.

49.6.29 MCAN Receive FIFO 0 Acknowledge

Name: MCAN_RXF0A
Offset: 0xA8
Reset: 0x00000000
Property: Read/Write

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
			F0AI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – F0AI[5:0] Receive FIFO 0 Acknowledge Index
After the processor has read a message or a sequence of messages from Receive FIFO 0 it has to write the buffer index of the last element read from Receive FIFO 0 to F0AI. This will set the Receive FIFO 0 Get Index MCAN_RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level MCAN_RXF0S.F0FL.

Figure 51-25. Event Line Block Diagram

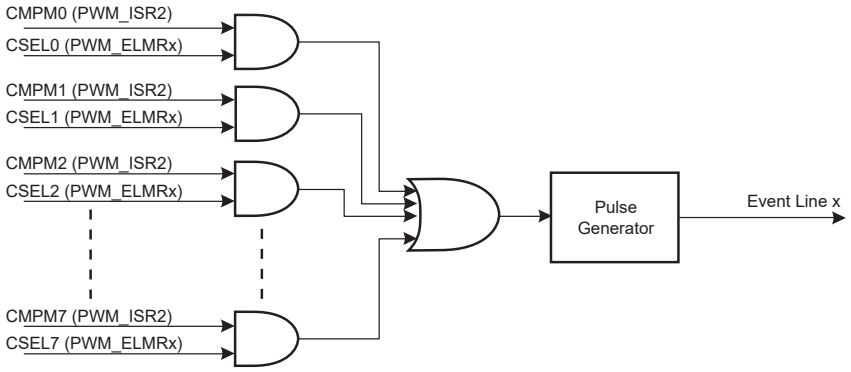
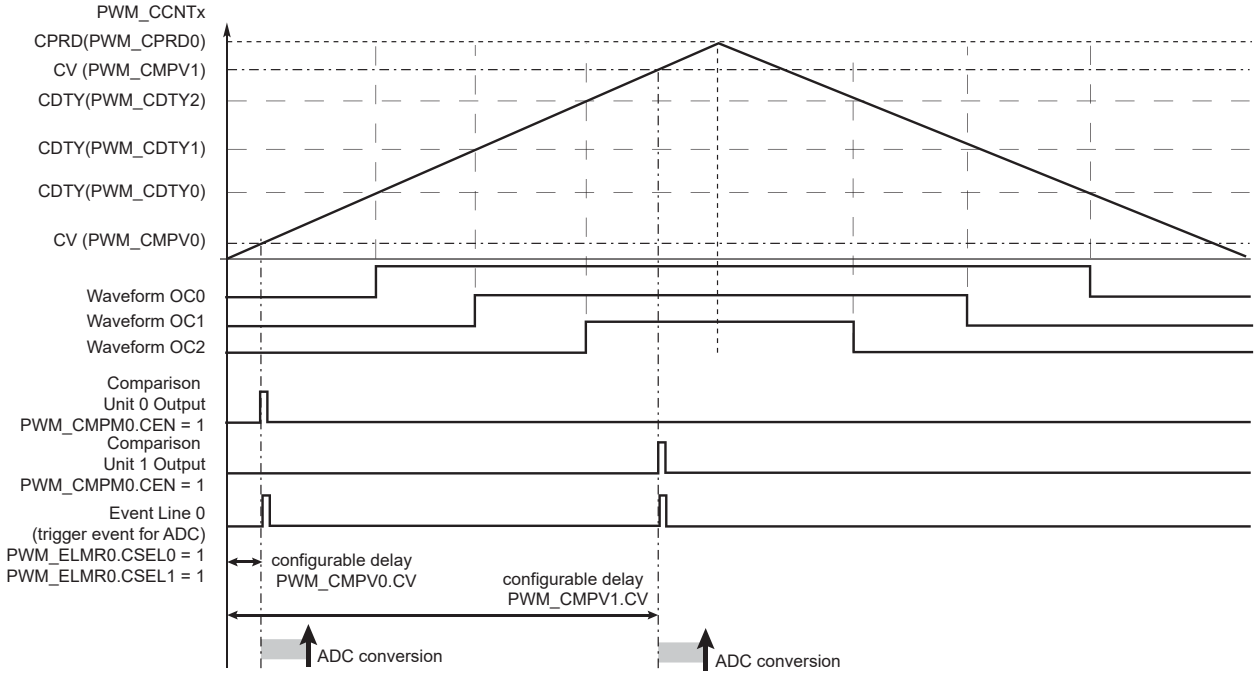


Figure 51-26. Event Line Generation Waveform (Example)



51.6.5 PWM External Trigger Mode

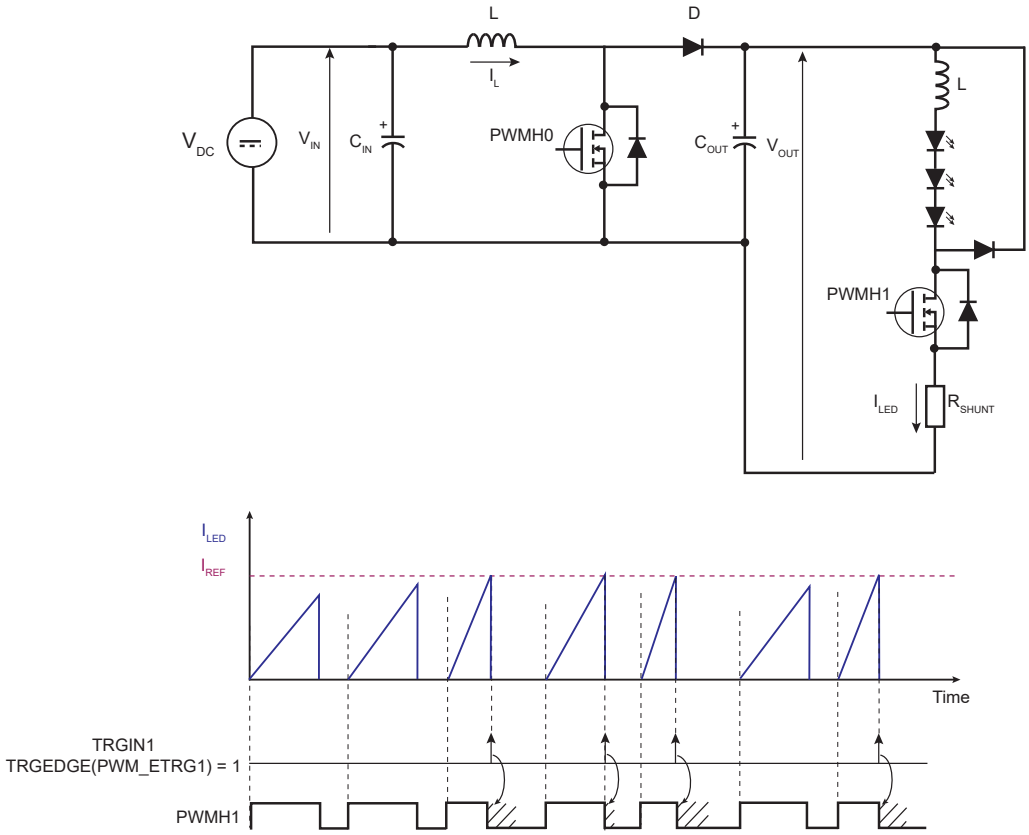
The PWM channels 1 and 2 can be configured to use an external trigger for generating specific PWM signals. The external trigger source can be selected through the bit TRGSRC of the PWM External Trigger Register (see the table below).

Table 51-5. External Event Source Selection

Channel	Trigger Source Selection	Trigger Source
1	PWM_ETRG1.TRGSRC = 0	From PWMEXTRG1 input
	PWM_ETRG1.TRGSRC = 1	From Analog Comparator Controller
2	PWM_ETRG2.TRGSRC = 0	From PWMEXTRG2 input
	PWM_ETRG2.TRGSRC = 1	From Analog Comparator Controller

Each external trigger source can be filtered by writing a one to the TRGFILT bit in the corresponding PWM External Trigger Register (PWM_ETRGx).

Figure 51-32. Cycle-By-Cycle Duty Mode: LED String Control



51.6.5.4 Leading-Edge Blanking (LEB)

PWM channels 1 and 2 support leading-edge blanking. Leading-edge blanking masks the external trigger input when a transient occurs on the corresponding PWM output. It masks potential spurious external events due to power transistor switching.

The blanking delay on each external trigger input is configured by programming the LEBDELAYx in the PWM Leading-Edge Blanking Register.

The LEB can be enabled on both the rising and the falling edges for the PWMH and PWML outputs through the bits PWMLFEN, PWMLREN, PWMHFEN, PWMHREN.

Any event on the PWMEXTRGx input which occurs during the blanking time is ignored.

51.7.6 PWM Interrupt Disable Register 1

Name: PWM_IDR1
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FCHID3	FCHID2	FCHID1	FCHID0
Access					W	W	W	W
Reset					0	0	0	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					0	0	0	–

Bits 16, 17, 18, 19 – FCHIDx Fault Protection Trigger on Channel x Interrupt Disable

Bits 0, 1, 2, 3 – CHIDx Counter Event on Channel x Interrupt Disable

51.7.44 PWM Channel Period Update Register

Name: PWM_CPRDUPDx
Offset: 0x0210 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPRD value. This prevents an unexpected waveform when modifying the waveform period.

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CPRDUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPRDUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPRDUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 23:0 – CPRDUPD[23:0] Channel Period Update
If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:

– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times \mathrm{CPRDUPD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times \mathrm{CPRDUPD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or } \frac{(X \times \mathrm{CPRDUPD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$

52.7.15 AFEC Compare Window Register

Name: AFEC_CWR
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	HIGHTHRES[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	23	22	21	20	19	18	17	16
	HIGHTHRES[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	15	14	13	12	11	10	9	8
	LOWTHRES[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
	LOWTHRES[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 31:16 – HIGHTHRES[15:0] High Threshold
High threshold associated to compare settings of AFEC_EMR. For comparisons lower than 16 bits and signed, the sign should be extended up to the bit 15.

Bits 15:0 – LOWTHRES[15:0] Low Threshold
Low threshold associated to compare settings of AFEC_EMR. For comparisons lower than 16 bits and signed, the sign should be extended up to the bit 15.

Figure 58-2. VDDCORE Power-On Reset Characteristics

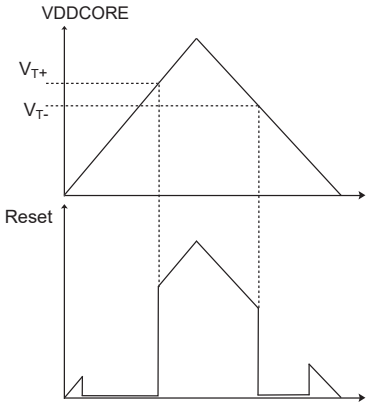


Table 58-8. VDDIO Supply Monitor

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_T	Supply Monitor Threshold	16 selectable steps (see the Threshold Selection table below)	–	–	–	V
T_{ACC}	Threshold Accuracy	–	-4	–	4	%
V_{hys}	Hysteresis Voltage	–	–	38	45	mV
t_{START}	Startup Time	From disabled state to enabled state	–	–	300	µs

Table 58-9. Threshold Selection

Symbol	Parameter	Digital Code	Min	Typ	Max	Unit
V_T	Supply Monitor Threshold	0	–	1.6	–	V
		1	–	1.72	–	
		10	–	1.84	–	
		11	–	1.96	–	
		100	–	2.08	–	
		101	–	2.2	–	
		110	–	2.32	–	
		111	–	2.44	–	
		1000	–	2.56	–	
		1001	–	2.68	–	
		1010	–	2.8	–	
		1011	–	2.92	–	
		1100	–	3.04	–	
		1101	–	3.16	–	

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
		Normal mode with one output on, DACC_ACR.IBCTLCHx =1 (see **Note 1**) FS = 500 KSps, no R_{LOAD}, V_{DDIN} = 3.3V	–	100	400	
		Bypass mode (output buffer off) with one output on, DACC_ACR.IBCTLCHx =0 (see **Note 1**) FS = 500 KSps, no R_{LOAD}, V_{DDIN} = 3.3V	–	10	30	
PSRR	Power Supply RejectionRatio (V_{DDIN})	V_{DDIN} ±10 mV Up to 10 kHz	–	70	–	dB

Note:

1. The maximum conversion rate versus the configuration of DACC_ACR.IBCTL is shown in the following table.

Table 59-44. Maximum Conversion Rate vs. Configuration of DACC_ACR.IBCTL

DACC_ACR.IBCTLCHx	Maximum Conversion Rate
0	Bypass
1	500 ks/s
2	N/A
3	1 Ms/s

Table 59-45. Voltage Reference

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{VREFP}	Positive Voltage Reference	Externally decoupled 1 µF	1.7	–	V_{DDIN}	V
I_{VREFP}	DC Current on VREFP	–	–	2.5	–	µA

Note: V_{REFP} is the positive reference shared with AFE and may have a different value for AFE. Refer to the AFE electrical characteristics if AFE is used. The V_{REFN} pin must be connected to ground.

Table 59-46. DAC Clock

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{DAC}	DAC Clock Frequency	–	–	–	12	MHz
f_S	Sampling Frequency	–	–	f_{DAC} / 12	–	MHz