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	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	1MB (1M 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 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70n20a-cnt

31.15 Main Crystal Oscillator Failure Detection

The Main crystal oscillator failure detector monitors the Main crystal oscillator against the Slow RC oscillator and provides an automatic switchover of the MAINCK source to the Main RC oscillator in case of failure detection.

The failure detector can be enabled or disabled by configuring the CKGR_MOR.CFDEN, and it can also be disabled in either of the following cases:

- After a VDDCORE reset
- When the Main crystal oscillator is disabled (MOSCXTEN = 0)

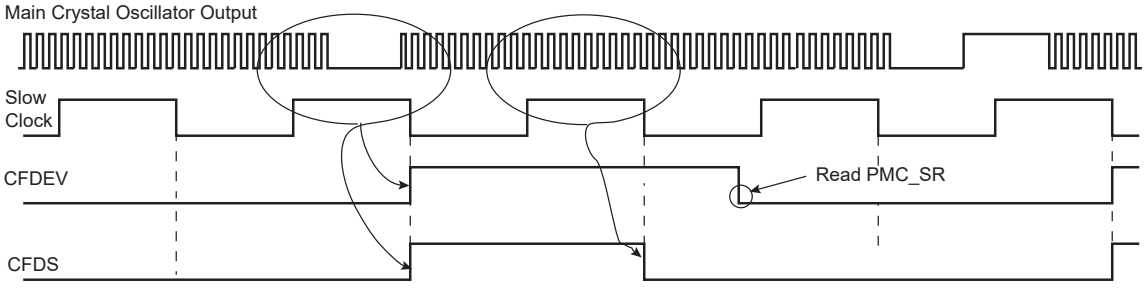
A failure is detected by means of a counter incrementing on the Main crystal oscillator output and detection logic is triggered by the Slow RC oscillator which is automatically enabled when CFDEN = 1.

The counter is cleared when the Slow RC oscillator clock signal is low and enabled when the signal is high. Thus, the failure detection time is one Slow RC oscillator period. If, during the high level period of the Slow RC oscillator clock signal, less than eight Main crystal oscillator clock periods have been counted, then a failure is reported. Note that when enabling the failure detector, up to two cycles of the Slow RC oscillator are needed to detect a failure of the Main crystal oscillator.

If a failure of Main crystal oscillator is detected, PMC_SR.CFDEV and PMC_SR.FOS both indicate a failure event. PMC_SR.CFDEV is cleared on read of PMC_SR, and PMC_SR.FOS is cleared by writing a ‘1’ to the FOCLR bit in the PMC Fault Output Clear Register (PMC_FOCR).

Only PMC_SR.CFDEV can generate an interrupt if the corresponding interrupt source is enabled in PMC_IER. The current status of the clock failure detection can be read at any time from PMC_SR.CFDS.

Figure 31-5. Clock Failure Detection Example



Note: Ratio of clock periods is for illustration purposes only.

If the Main crystal oscillator is selected as the source clock of MAINCK (CKGR_MOR.MOSCSEL = 1), and if the MCK source is PLLACK or UPLLCKDIV (CSS = 2 or 3), a clock failure detection automatically forces MAINCK to be the source clock for MCK. Then, regardless of the PMC configuration, a clock failure detection automatically forces the Main RC oscillator to be the source clock for MAINCK. If the Main RC oscillator is disabled when a clock failure detection occurs, it is automatically re-enabled by the clock failure detection mechanism.

Two Slow RC oscillator clock cycles are necessary to detect and switch from the Main crystal oscillator to the Main RC oscillator if the source of MCK is MAINCK, or three Slow RC oscillator clock cycles if the source of MCK is PLLACK or UPLLCKDIV.

A clock failure detection activates a fault output that is connected to the Pulse Width Modulator (PWM) Controller. With this connection, the PWM controller is able to force its outputs and to protect the driven device, if a clock failure is detected.

These additional modes are:

- Rising edge detection
- Falling edge detection
- Low-level detection
- High-level detection

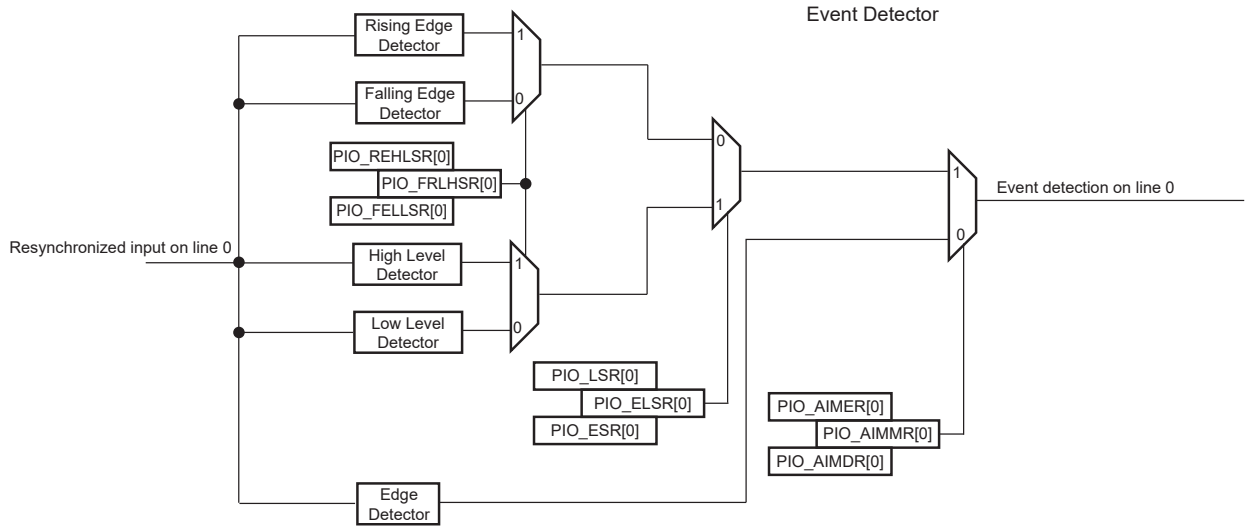
In order to select an additional interrupt mode:

- The type of event detection (edge or level) must be selected by writing in the Edge Select Register (PIO_ESR) and Level Select Register (PIO_LSR) which select, respectively, the edge and level detection. The current status of this selection is accessible through the Edge/Level Status Register (PIO_ELSR).
- The polarity of the event detection (rising/falling edge or high/low-level) must be selected by writing in the Falling Edge/Low-Level Select Register (PIO_FELLSR) and Rising Edge/High-Level Select Register (PIO_REHLSR) which allow to select falling or rising edge (if edge is selected in PIO_ELSR) edge or high- or low-level detection (if level is selected in PIO_ELSR). The current status of this selection is accessible through the Fall/Rise - Low/High Status Register (PIO_FRLHSR).

When an input edge or level is detected on an I/O line, the corresponding bit in the Interrupt Status Register (PIO_ISR) is set. If the corresponding bit in PIO_IMR is set, the PIO Controller interrupt line is asserted.The interrupt signals of the 32 channels are ORed-wired together to generate a single interrupt signal to the interrupt controller.

When the software reads PIO_ISR, all the interrupts are automatically cleared. This signifies that all the interrupts that are pending when PIO_ISR is read must be handled. When an Interrupt is enabled on a “level”, the interrupt is generated as long as the interrupt source is not cleared, even if some read accesses in PIO_ISR are performed.

Figure 32-6. Event Detector on Input Lines (Figure Represents Line 0)



Example of interrupt generation on following lines:

- Rising edge on PIO line 0
- Falling edge on PIO line 1
- Rising edge on PIO line 2
- Low-level on PIO line 3

34.7.2 SDRAMC Refresh Timer Register

Name: SDRAMC_TR
Offset: 0x04
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
					COUNT[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[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 11:0 – COUNT[11:0] SDRAMC Refresh Timer Count
This 12-bit field is loaded into a timer that generates the refresh pulse. Each time the refresh pulse is generated, a refresh burst is initiated. The SDRAM device requires a refresh every 15.625 µs or 7.81 µs. With a 100 MHz frequency, the Refresh Timer Counter Register must be set with the value 1562 (15.625 µs x 100 MHz) or 781 (7.81 µs x 100 MHz).

To refresh the SDRAM device, this 12-bit field must be written. If this condition is not satisfied, no refresh command is issued and no refresh of the SDRAM device is carried out.

Offset	Name	Bit Pos.								
0x0644	USBHS_HSTPIPIDR9 (INTPIPES)	7:0	SHORTPACKETIEC	RXSTALLDEC	OVERFIEC	NAKEDEC	PERREC	UNDERFIEC	TXOUTEC	RXINEC
		15:8		FIFOCONC		NBUSYBKEC				
		23:16							PFREEZEC	PDISHDMAC
		31:24								
0x0644	USBHS_HSTPIPIDR9 (ISOPIPES)	7:0	SHORTPACKETIEC	CRCERREC	OVERFIEC	NAKEDEC	PERREC	UNDERFIEC	TXOUTEC	RXINEC
		15:8		FIFOCONC		NBUSYBKEC				
		23:16							PFREEZEC	PDISHDMAC
		31:24								
0x0648 ... 0x064F	Reserved									
0x0650	USBHS_HSTPIPINRQ0	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0654	USBHS_HSTPIPINRQ1	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0658	USBHS_HSTPIPINRQ2	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x065C	USBHS_HSTPIPINRQ3	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0660	USBHS_HSTPIPINRQ4	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0664	USBHS_HSTPIPINRQ5	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0668	USBHS_HSTPIPINRQ6	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x066C	USBHS_HSTPIPINRQ7	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								

42.7.2 QSPI Mode Register

Name: QSPI_MR
Offset: 0x04
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
	DLYCS[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	23	22	21	20	19	18	17	16
	DLYBCT[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
					NBBITS[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
			CSMODE[1:0]			WDRBT	LLB	SMM
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 31:24 – DLYCS[7:0] Minimum Inactive QCS Delay
This field defines the minimum delay between the deactivation and the activation of QCS. The DLYCS time guarantees the slave minimum deselect time.

If DLYCS written to ‘0’, one peripheral clock period is inserted by default.

Otherwise, the following equation determines the delay:

DLYCS = Minimum inactive × $f_{peripheral\ clock}$

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.

When DLYBCT is written to ‘0’, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers. In Serial Memory mode (SMM = 1), DLYBCT must be written to ‘0’ and no delay is inserted.

Otherwise, the following equation determines the delay:

DLYBCT = (Delay Between Consecutive Transfers × $f_{peripheral\ clock}$) / 32

42.7.11 QSPI Instruction Code Register

Name: QSPI_ICR
Offset: 0x34
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
	OPT[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

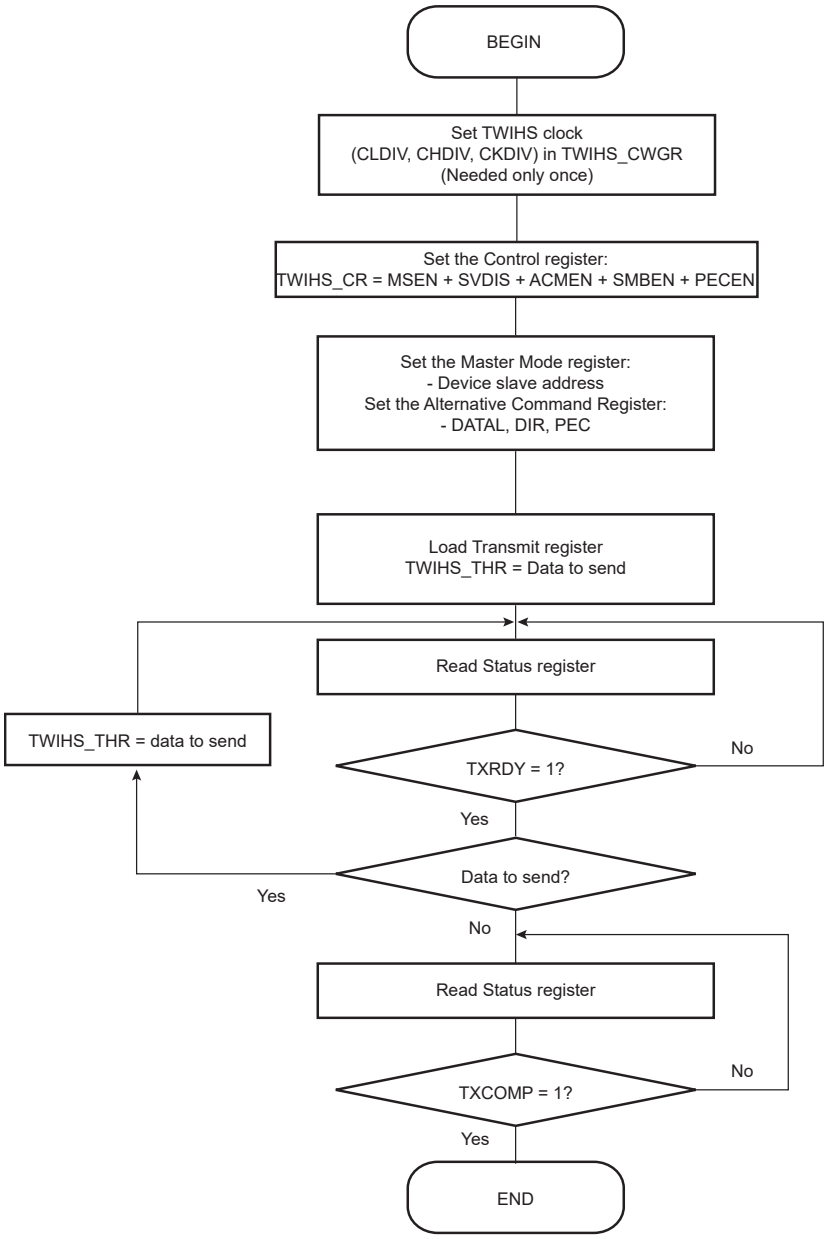
Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INST[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 23:16 – OPT[7:0] Option Code
Option code to send to the serial Flash memory.

Bits 7:0 – INST[7:0] Instruction Code
Instruction code to send to the serial Flash memory.

Figure 43-20. TWIHS Write Operation with Multiple Data Bytes + Read Operation and Alternative Command Mode + PEC



44.9.4 SSC Receive Frame Mode Register

Name: SSC_RFMR
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	FSLEN_EXT[3:0]							FSEDGE
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	23	22	21	20	19	18	17	16
		FSOS[2:0]			FSLEN[3:0]			
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
					DATNB[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
	MSBF		LOOP	DATLEN[4:0]				
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:28 – FSLEN_EXT[3:0] FSLEN Field Extension
Extends FSLEN field. For details, see FSLEN: Receive Frame Sync Length.

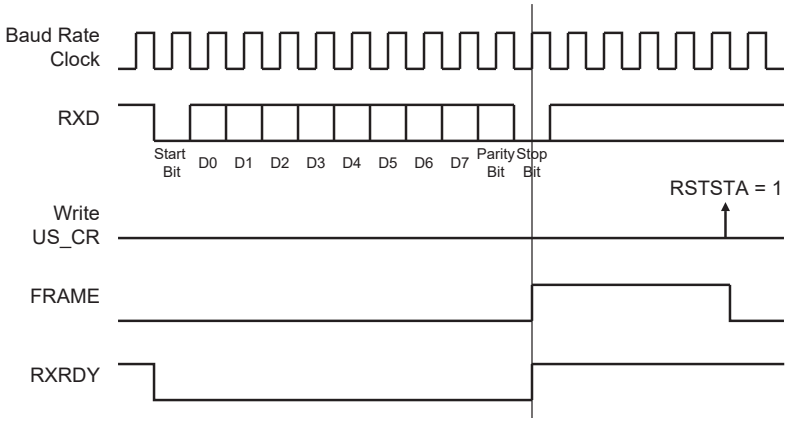
Bit 24 – FSEDGE Frame Sync Edge Detection
Determines which edge on Frame Sync will generate the interrupt RXSYN in the SSC Status Register.

Value	Name	Description
0	POSITIVE	Positive Edge Detection
1	NEGATIVE	Negative Edge Detection

Bits 22:20 – FSOS[2:0] Receive Frame Sync Output Selection

Value	Name	Description
0	NONE	None, RF pin is an input
1	NEGATIVE	Negative Pulse, RF pin is an output
2	POSITIVE	Positive Pulse, RF pin is an output
3	LOW	Driven Low during data transfer, RF pin is an output
4	HIGH	Driven High during data transfer, RF pin is an output
5	TOGGLING	Toggling at each start of data transfer, RF pin is an output

Figure 46-24. Framing Error Status



46.6.3.13 Transmit Break

The user can request the transmitter to generate a break condition on the TXD line. A break condition drives the TXD line low during at least one complete character. It appears the same as a 0x00 character sent with the parity and the stop bits at 0. However, the transmitter holds the TXD line at least during one character until the user requests the break condition to be removed.

A break is transmitted by writing a ‘1’ to US_CR.STTBRK. This can be performed at any time, either while the transmitter is empty (no character in either the Shift register or in US_THR) or when a character is being transmitted. If a break is requested while a character is being shifted out, the character is first completed before the TXD line is held low.

Once STTBRK command is requested, further STTBRK commands are ignored until the end of the break is completed.

The break condition is removed by writing a ‘1’ to US_CR.STPBRK. If the STPBRK is requested before the end of the minimum break duration (one character, including start, data, parity and stop bits), the transmitter ensures that the break condition completes.

The transmitter considers the break as though it is a character, i.e., the STTBRK and STPBRK commands are processed only if US_CSR. TXRDY = 1 and the start of the break condition clears the TXRDY and TXEMPTY bits as if a character is processed.

Writing US_CR with both STTBRK and STPBRK bits to ‘1’ can lead to an unpredictable result. All STPBRK commands requested without a previous STTBRK command are ignored. A byte written into US_THR while a break is pending, but not started, is ignored.

After the break condition, the transmitter returns the TXD line to 1 for a minimum of 12 bit times. Thus, the transmitter ensures that the remote receiver detects correctly the end of break and the start of the next character. If the timeguard is programmed with a value higher than 12, the TXD line is held high for the timeguard period.

After holding the TXD line for this period, the transmitter resumes normal operations.

The following figure illustrates the effect of both the Start Break (STTBRK) and Stop Break (STPBRK) commands on the TXD line.

NACT(slave1)=PUBLISH

NACT(slave2)=SUBSCRIBE

- Data transfer from the slave2 to the master and to the slave1:

NACT(master)=SUBSCRIBE

NACT(slave1)=SUBSCRIBE

NACT(slave2)=PUBLISH

46.6.9.11 Response Data Length

The LIN response data length is the number of data fields (bytes) of the response excluding the checksum.

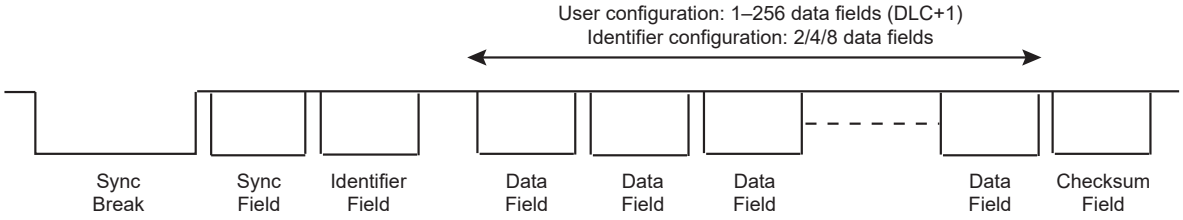
The response data length can either be configured by the user or be defined automatically by bits 4 and 5 of the Identifier (compatibility to LIN Specification 1.1). The user can choose between these two modes using the US_LINMR.DLM:

- DLM = 0: The response data length is configured by the user via US_LINMR.DLC. The response data length is equal to (DLC + 1) bytes. DLC can be programmed from 0 to 255, so the response can contain from 1 data byte up to 256 data bytes.
- DLM = 1: The response data length is defined by the Identifier (US_LINIR.IDCHR) according to the table below. The US_LINMR.DLC is discarded. The response can contain 2 or 4 or 8 data bytes.

Table 46-13. Response Data Length if DLM = 1

IDCHR[5]	IDCHR[4]	Response Data Length [Bytes]
0	0	2
0	1	2
1	0	4
1	1	8

Figure 46-43. Response Data Length



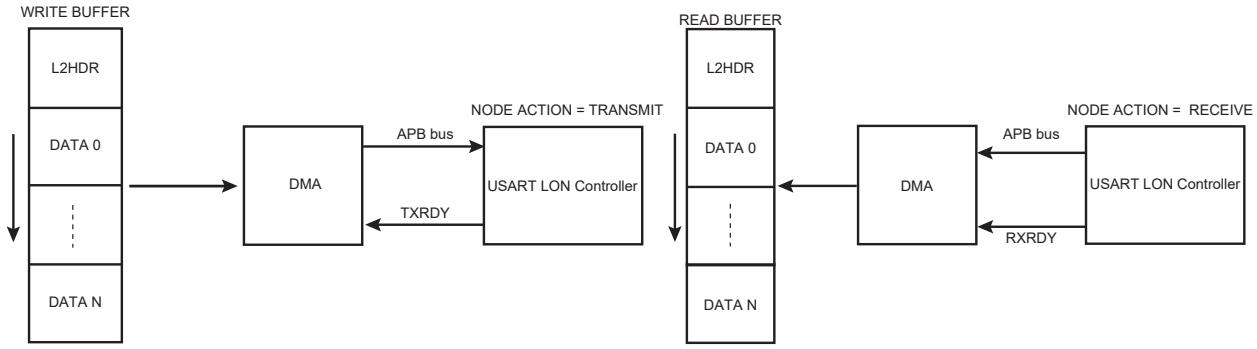
46.6.9.12 Checksum

The last field of a frame is the checksum. The checksum contains the inverted 8-bit sum with carry, over all data bytes or all data bytes and the protected identifier. Checksum calculation over the data bytes only is called classic checksum and it is used for communication with LIN 1.3 slaves. Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and it is used for communication with LIN 2.0 slaves.

The USART can be configured to:

- Send/Check an Enhanced checksum automatically (CHKDIS = 0 & CHKTYP = 0)
- Send/Check a Classic checksum automatically (CHKDIS = 0 & CHKTYP = 1)

Figure 46-64. DMAM = 0



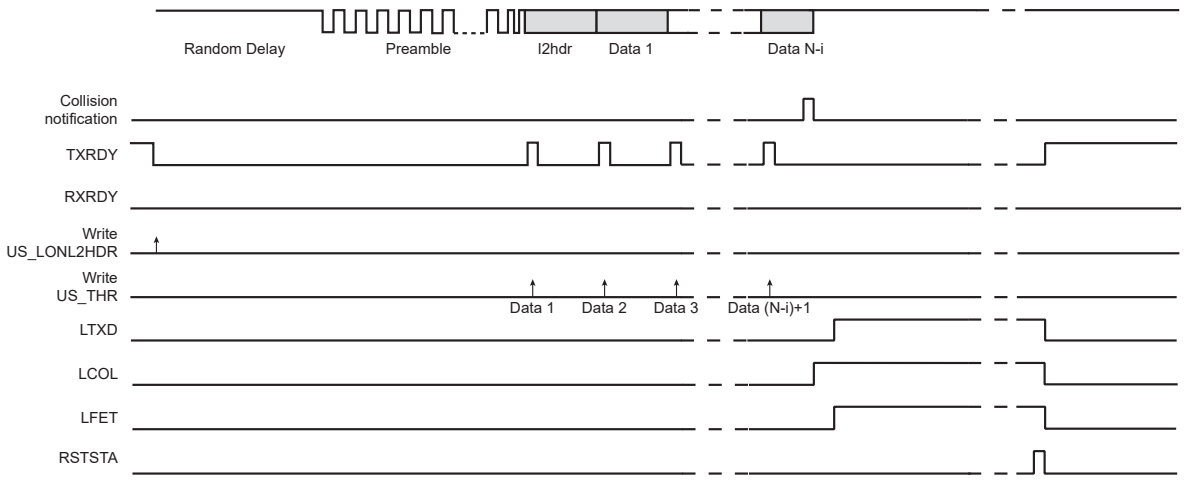
46.6.10.12.2 DMA and Collision Detection

As explained in "comm_type", depending on LON configuration the transmission may be terminated early upon collision notification which means that the DMA transfer may be stopped before its end.

In case of early end of transmission due to collision detection the USART in LON mode acts as follows:

- Send the end of frame trigger.
- Hold down TXRDY avoiding thus any additional DMA transfer.
- Set LTXD, LCOL and LFET flags in US_CSR.
- Wait that the application reconfigure the DMA.
- Wait until LCOL and LFET flags are cleared through US_CR. RSTSTA (it releases the TXRDY signal).

Figure 46-65. DMA, Collision and Early Frame Termination



46.6.11 Test Modes

The USART can be programmed to operate in three different test modes. The internal loopback capability allows on-board diagnostics. In Loopback mode, the USART interface pins are disconnected or not and reconfigured for loopback internally or externally.

46.6.11.1 Normal Mode

Normal mode connects the RXD pin on the receiver input and the transmitter output on the TXD pin.

46.7.47 USART Write Protection Status Register

Name: US_WPSR
Offset: 0x00E8
Reset: 0x0
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the US_WPSR.
1	A write protection violation has occurred since the last read of the US_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

48.7.12 HBI Channel Error 1 Register

Name: MLB_HCER1
Offset: 0x094
Reset: 0x00000000
Property: Read-only

HCERn status bits are set when hardware detects hardware errors on the given logical channel, including:

- Channel opened, but not enabled,
- Channel programmed with invalid channel type, or
- Out-of-range PML for asynchronous or control Tx channels

Bit	31	30	29	28	27	26	25	24
	CERR: Bitwise Channel Error Bit [63[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CERR: Bitwise Channel Error Bit [63[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CERR: Bitwise Channel Error Bit [63[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CERR: Bitwise Channel Error Bit [63[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CERR: Bitwise Channel Error Bit [63[31:0] 32]
CERR[n] = 1 indicates that a fatal error occurred on channel n.

Each linear divider can independently divide one of the clocks of the modulo n counter. The selection of the clock to be divided is made according to the PREA (PREB) field of the PWM Clock register (PWM_CLK). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value.

After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) are set to ‘0’. This implies that after reset clkA (clkB) are turned off.

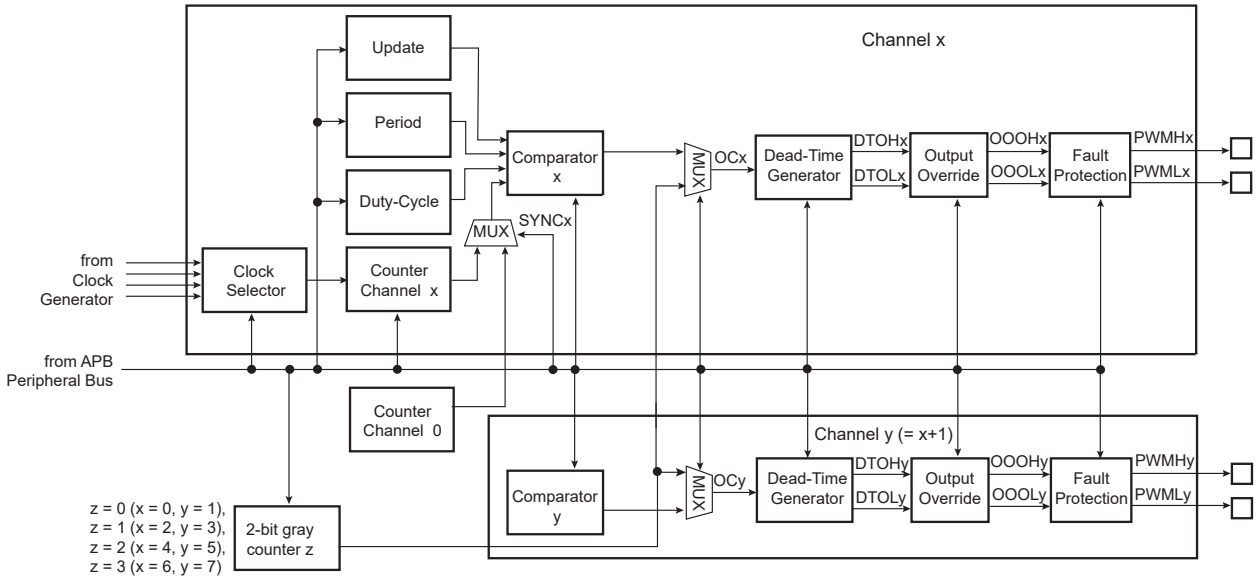
At reset, all clocks provided by the modulo n counter are turned off except the peripheral clock. This situation is also true when the PWM peripheral clock is turned off through the Power Management Controller.

CAUTION Before using the PWM controller, the programmer must first enable the peripheral clock in the Power Management Controller (PMC).

51.6.2 PWM Channel

51.6.2.1 Channel Block Diagram

Figure 51-3. Functional View of the Channel Block Diagram



Each of the 4 channels is composed of six blocks:

- A clock selector which selects one of the clocks provided by the clock generator (described in PWM Clock Generator).
- A counter clocked by the output of the clock selector. This counter is incremented or decremented according to the channel configuration and comparators matches. The size of the counter is 16 bits.
- A comparator used to compute the OCx output waveform according to the counter value and the configuration. The counter value can be the one of the channel counter or the one of the channel 0 counter according to SYNCx bit in the PWM Sync Channels Mode Register (PWM_SCM).
- A 2-bit configurable gray counter enables the stepper motor driver. One gray counter drives 2 channels.
- A dead-time generator providing two complementary outputs (DTOHx/DTOLx) which allows to drive external power control switches safely.

51.7.25 PWM Fault Status Register

Name: PWM_FSR
Offset: 0x60
Reset: 0x00000000
Property: Read-only

Refer to 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
	FS[7:0]							
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
	FIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – FS[7:0] Fault Status
For each bit y of FS, where y is the fault input number:

0: The fault y is not currently active.

1: The fault y is currently active.

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

0: The current sampled value of the fault input y is 0 (after filtering if enabled).

1: The current sampled value of the fault input y is 1 (after filtering if enabled).

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.

Value	Description
0	The SW write protection x of the register group x is disabled.
1	The SW write protection x of the register group x is enabled.

51.7.48 PWM Channel Mode Update Register

Name: PWM_CMUPDx
Offset: 0x0400 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

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

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
			CPOLINVUP				CPOLUP	
Access			W				W	
Reset			–				–	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 13 – CPOLINVUP Channel Polarity Inversion Update
If this bit is written at ‘1’, the write of the bit CPOLUP is not taken into account.

Value	Description
0	No effect.
1	The OCx output waveform (output from the comparator) is inverted.

Bit 9 – CPOLUP Channel Polarity Update
The write of this bit is taken into account only if the bit CPOLINVUP is written at ‘0’ at the same time.

Value	Description
0	The OCx output waveform (output from the comparator) starts at a low level.
1	The OCx output waveform (output from the comparator) starts at a high level.

Value	Name	Description
0	DISABLED	One data to convert is written to the FIFO per access to DACC.
1	ENABLED	Two data to convert are written to the FIFO per access to DACC (reduces the number of requests to DMA and the number of system bus accesses).

Bits 0, 1 – MAXSx Max Speed Mode for Channel x

Value	Name	Description
0	TRIG_EVENT	Trigger mode or Free-running mode enabled. (See TRGENx.DACC_TRIGR.)
1	MAXIMUM	Max speed mode enabled.

Figure 58-38. SSC Receiver, RK in Input and RF in Output

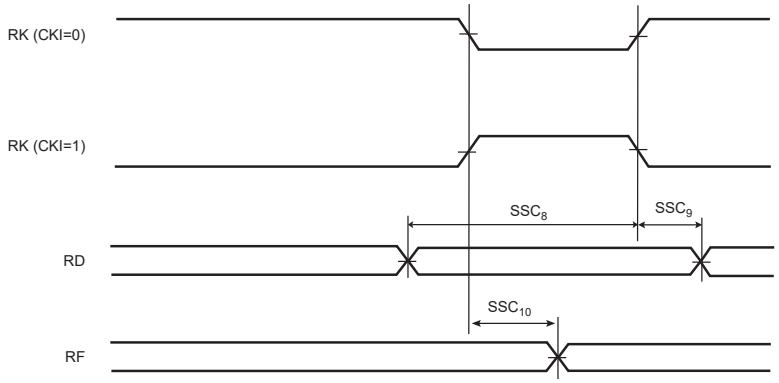


Figure 58-39. SSC Receiver, RK and RF in Output

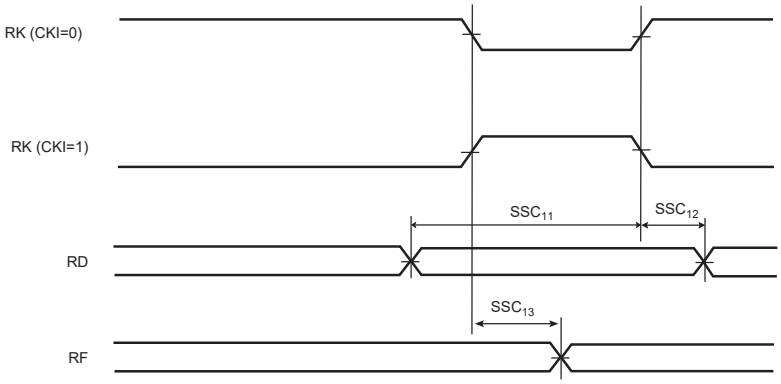


Figure 58-40. SSC Receiver, RK in Output and RF in Input

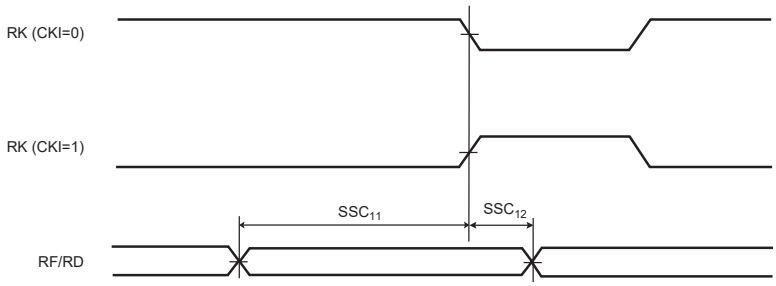


Table 58-74. SSC Timings with 3.3V Peripheral Supply

Symbol	Parameter	Condition	Min	Max	Unit
Transmitter					
SSC_0	TK edge to TF/TD (TK output, TF output)	–	-3.9(1)	4.0 (1)	ns
SSC_1	TK edge to TF/TD (TK input, TF output)	–	3.1(1)	12.7(1)	ns
SSC_2	TF setup time before TK edge (TK output)	–	13.6	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	ns