Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

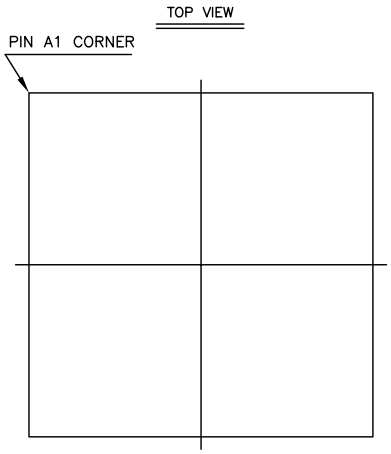
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, EBI/EMI, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	114
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 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-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamv70q20b-aabt

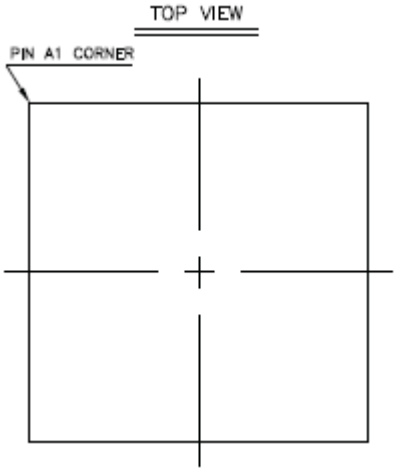
6.1.2 144-ball LFBGA/TFBGA Package Outline

Figure 6-2. Orientation of the 144-ball LFBGA/TFBGA Package



6.1.3 144-ball UFBGA Package Outline

Figure 6-3. Orientation of the 144-ball UFBGA Package



6.2 144-lead Package Pinout

Table 6-1. 144-lead Package Pinout

LQFP Pin	LFBGA/ TFBGA Ball	UFBGA Ball	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral A		PIO Peripheral B		PIO Peripheral C		PIO Peripheral D		Reset State
					Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST
102	C11	E11	VDDIO	GPIO_AD	PA0	I/O	WKUP0(1)	I	PWMC0_PWMH0	O	TIOA0	I/O	A17/BA1	O	I2SC0_MCK	O	PIO, I, PU, ST
99	D12	F11	VDDIO	GPIO_AD	PA1	I/O	WKUP1(1)	I	PWMC0_PWML0	O	TIOB0	I/O	A18	O	I2SC0_CK	I/O	PIO, I, PU, ST
93	E12	G12	VDDIO	GPIO	PA2	I/O	WKUP2(1)	I	PWMC0_PWMH1	O	–	–	DATRG	I	–	–	PIO, I, PU, ST
91	F12	G11	VDDIO	GPIO_AD	PA3	I/O	PIODC0(2)	I	TWD0	I/O	LONCOL1	I	PCK2	O	–	–	PIO, I, PU, ST

9. Interconnect

The system architecture is based on the ARM Cortex-M7 processor connected to the main AHB Bus Matrix, the embedded Flash, the multi-port SRAM and the ROM.

The 32-bit AHBP interface is a single 32-bit wide interface that accesses the peripherals connected on the main Bus Matrix. It is used only for data access. Instruction fetches are never performed on the AHBP interface. The bus, AHBP or AXIM, accessing the peripheral memory area [0x40000000 to 0x60000000] is selected in the AHBP control register.

The 32-bit AHBS interface provides system access to the ITCM, D1TCM, and D0TCM. It is connected on the main Bus Matrix and allows the XDMA to transfer from memory or peripherals to the instruction or data TCMs.

The 64-bit AXIM interface is a single 64-bit wide interface connected through two ports of the AXI Bridge to the main AHB Bus Matrix and to two ports of the multi-port SRAM. The AXIM interface allows:

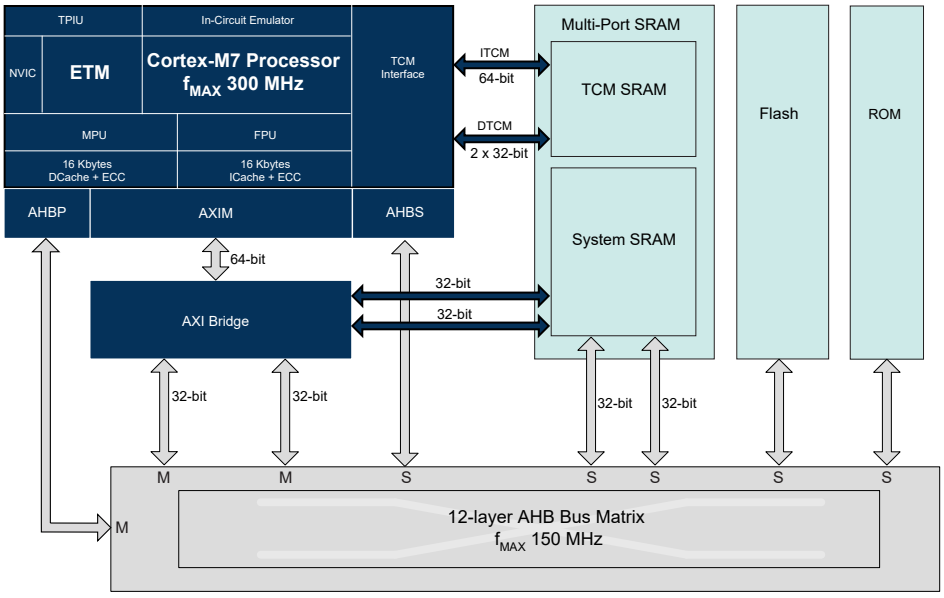
- Instruction fetches
- Data cache linefills and evictions
- Non-cacheable normal-type memory data accesses
- Device and strongly-ordered type data accesses, generally to peripherals

The interleaved multi-port SRAM optimizes the Cortex-M7 accesses to the internal SRAM.

The interconnect of the other masters and slaves is described in 19. Bus Matrix (MATRIX).

The figure below shows the connections of the different Cortex-M7 ports.

Figure 9-1. Interconnect Block Diagram



1. Round-robin Arbitration (default)
2. Fixed Priority Arbitration

Each algorithm may be complemented by selecting a default master configuration for each slave.

When re-arbitration is required, specific conditions apply. Refer to the "Arbitration Rules" section.

19.3.3.1 Arbitration Rules

Each arbiter has the ability to arbitrate between requests from two or more masters. To avoid burst breaking and to provide maximum throughput for slave interfaces, arbitration should take place during the following cycles:

- Idle cycles: When a slave is not connected to any master or is connected to a master which is not currently accessing it
- Single cycles: When a slave is performing a single access
- End of Burst cycles: When the current cycle is the last cycle of a burst transfer. For a defined length burst, predicted end of burst matches the size of the transfer but is managed differently for undefined length burst. Refer to the "Undefined Length Burst Arbitration" section.
- Slot cycle limit: When the slot cycle counter has reached the limit value indicating that the current master access is too long and must be broken. Refer to the "Slot Cycle Limit Arbitration" section.

19.3.3.1.1 Undefined Length Burst Arbitration

In order to prevent slave handling during undefined length bursts, the user can trigger the re-arbitration before the end of the incremental bursts. The re-arbitration period can be selected from the following Undefined Length Burst Type (ULBT) possibilities:

1. Unlimited: no predetermined end of burst is generated. This value enables 1-Kbyte burst lengths.
2. 1-beat bursts: predetermined end of burst is generated at each single transfer during the INCR transfer.
3. 4-beat bursts: predetermined end of burst is generated at the end of each 4-beat boundary during INCR transfer.
4. 8-beat bursts: predetermined end of burst is generated at the end of each 8-beat boundary during INCR transfer.
5. 16-beat bursts: predetermined end of burst is generated at the end of each 16-beat boundary during INCR transfer.
6. 32-beat bursts: predetermined end of burst is generated at the end of each 32-beat boundary during INCR transfer.
7. 64-beat bursts: predetermined end of burst is generated at the end of each 64-beat boundary during INCR transfer.
8. 128-beat bursts: predetermined end of burst is generated at the end of each 128-beat boundary during INCR transfer.

The use of undefined length16-beat bursts, or less, is discouraged since this decreases the overall bus bandwidth due to arbitration and slave latencies at each first access of a burst.

If the master does not permanently and continuously request the same slave or has an intrinsically limited average throughput, the ULBT should be left at its default unlimited value, knowing that the AHB specification natively limits all word bursts to 256 beats and double-word bursts to 128 beats because of its 1-Kbyte address boundaries.

Unless duly needed, the ULBT should be left at its default value of 0 for power saving.

This selection is made through the ULBT field of the Master Configuration Registers (MATRIX_MCFG).

Offset	Register	Name	Access	Reset
0x0028	PLLA Register	CKGR_PLLAR	Read/Write	0x0000_3F00
0x002C	Reserved	–	–	–
0x0030	Master Clock Register	PMC_MCKR	Read/Write	0x0000_0001
0x0034	Reserved	–	–	–
0x0038	USB Clock Register	PMC_USB	Read/Write	0x0000_0000
0x003C	Reserved	–	–	–
0x0040+chid*0x04	Programmable Clock Register	PMC_PCK	Read/Write	0x0000_0000
0x005C	Reserved	–	–	–
0x0060	Interrupt Enable Register	PMC_IER	Write-only	–
0x0064	Interrupt Disable Register	PMC_IDR	Write-only	–
0x0068	Status Register	PMC_SR	Read-only	0x0003_0008
0x006C	Interrupt Mask Register	PMC_IMR	Read-only	0x0000_0000
0x0070	Fast Startup Mode Register	PMC_FSMR	Read/Write	0x0000_0000
0x0074	Fast Startup Polarity Register	PMC_FSPR	Read/Write	0x0000_0000
0x0078	Fault Output Clear Register	PMC_FOCR	Write-only	–
0x007C–0x00E0	Reserved	–	–	–
0x00E4	Write Protection Mode Register	PMC_WPMR	Read/Write	0x0
0x00E8	Write Protection Status Register	PMC_WPSR	Read-only	0x0
0x00EC–0x00FC	Reserved	–	–	–
0x0100	Peripheral Clock Enable Register 1	PMC_PCER1	Write-only	–
0x0104	Peripheral Clock Disable Register 1	PMC_PCDR1	Write-only	–
0x0108	Peripheral Clock Status Register 1	PMC_PCSR1	Read-only	0x0000_0000
0x010C	Peripheral Control Register	PMC_PCR	Read/Write	0x0000_0000
0x0110	Oscillator Calibration Register	PMC_OCR	Read/Write	(See **Note 2**)
0x0114	SleepWalking Enable Register 0	PMC_SLPWK_ER0	Write-only	–
0x0118	SleepWalking Disable Register 0	PMC_SLPWK_DR0	Write-only	–
0x011C	SleepWalking Status Register 0	PMC_SLPWK_SR0	Read-only	0x00000000
0x0120	SleepWalking Activity Status Register 0	PMC_SLPWK_ASR0	Read-only	0x00000000

32.6.1.49 PIO I/O Drive Register 1

Name: PIO_DRIVER1
Offset: 0x0118
Property: Read/Write

Register Reset value: 0x000000000xAAAAAAAA

Bit	31	30	29	28	27	26	25	24
	LINE31	LINE30	LINE29	LINE28	LINE27	LINE26	LINE25	LINE24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LINE23	LINE22	LINE21	LINE20	LINE19	LINE18	LINE17	LINE16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0
Access								
Reset								

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 – LINE Drive of PIO Line

Value	Name	Description
0	LOW_DRIVE	Lowest drive
1	HIGH_DRIVE	Highest drive

Table 37-5. RGB Format in Default Mode, RGB_CFG = 00, Swap Activated

Mode	Byte	D7	D6	D5	D4	D3	D2	D1	D0
RGB 8:8:8	Byte 0	R0(i)	R1(i)	R2(i)	R3(i)	R4(i)	R5(i)	R6(i)	R7(i)
	Byte 1	G0(i)	G1(i)	G2(i)	G3(i)	G4(i)	G5(i)	G6(i)	G7(i)
	Byte 2	B0(i)	B1(i)	B2(i)	B3(i)	B4(i)	B5(i)	B6(i)	B7(i)
	Byte 3	R0(i+1)	R1(i+1)	R2(i+1)	R3(i+1)	R4(i+1)	R5(i+1)	R6(i+1)	R7(i+1)
RGB 5:6:5	Byte 0	G3(i)	G4(i)	G5(i)	R0(i)	R1(i)	R2(i)	R3(i)	R4(i)
	Byte 1	B0(i)	B1(i)	B2(i)	B3(i)	B4(i)	G0(i)	G1(i)	G2(i)
	Byte 2	G3(i+1)	G4(i+1)	G5(i+1)	R0(i+1)	R1(i+1)	R2(i+1)	R3(i+1)	R4(i+1)
	Byte 3	B0(i+1)	B1(i+1)	B2(i+1)	B3(i+1)	B4(i+1)	G0(i+1)	G1(i+1)	G2(i+1)

The RGB 5:6:5 input format is processed to be displayed as RGB 5:6:5 format, compliant with the 16-bit mode of the LCD controller.

37.5.3 Clocks

The sensor master clock (ISI_MCK) can be generated either by the Advanced Power Management Controller (APMC) through a Programmable Clock output or by an external oscillator connected to the sensor.

None of the sensors embed a power management controller, so providing the clock by the APMC is a simple and efficient way to control power consumption of the system.

Care must be taken when programming the system clock. The ISI has two clock domains, the sensor master clock and the pixel clock provided by sensor. The two clock domains are not synchronized, but the sensor master clock must be faster than the pixel clock.

37.5.4 Preview Path

37.5.4.1 Scaling, Decimation (Subsampling)

This module resizes captured 8-bit color sensor images to fit the LCD display format. The resize module performs only downscaling. The same ratio is applied for both horizontal and vertical resize, then a fractional decimation algorithm is applied.

The decimation factor is a multiple of 1/16; values 0 to 15 are forbidden.

Table 37-6. Decimation Factor

Decimation Value	0–15	16	17	18	19	...	124	125	126	127
Decimation Factor	—	1	1.063	1.125	1.188	...	7.750	7.813	7.875	7.938

Table 37-7. Decimation and Scaler Offset Values

OUTPUT	INPUT	352 × 288	640 × 480	800 × 600	1280 × 1024	1600 × 1200	2048 × 1536
VGA 640 × 480	F	—	16	20	32	40	51
QVGA	F	16	32	40	64	80	102

38.8.70 GMAC 1519 to Maximum Byte Frames Received Register

Name: GMAC_TMXBFR
Offset: 0x180
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

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

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

Bits 31:0 – NFRX[31:0] 1519 to Maximum Byte Frames Received without Error
This bit field counts the number of 1519 Byte or above frames successfully received without error. Maximum frame size is determined by the Maximum Frame Size bit (MAXFS, 1536 Bytes) or Jumbo Frame Size bit (JFRAME, 10240 Bytes) in the Network Configuration Register (GMAC_NCFGR). Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

39.6.14 Device Endpoint x Configuration Register

Name: USBHS_DEVEPTCFGx
Offset: 0x0100 + x*0x04 [x=0..9]
Reset: 0
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
		NBTRANS[1:0]		EPTYPE[1:0]			AUTOSW	EPDIR
Access								
Reset		0	0	0	0		0	0

Bit	7	6	5	4	3	2	1	0
		EPSIZE[2:0]			EPBK[1:0]		ALLOC	
Access								
Reset		0	0	0	0	0	0	

Bits 14:13 – NBTRANS[1:0] Number of transactions per microframe for isochronous endpoint
This field should be written with the number of transactions per microframe to perform high-bandwidth isochronous transfer.

It can be written only for endpoints that have this capability (see USBHS_FEATURES.ENHBISOx bit). Otherwise, this field is 0.

This field is irrelevant for non-isochronous endpoints.

Value	Name	Description
0	0_TRANS	Reserved to endpoint that does not have the high-bandwidth isochronous capability.
1	1_TRANS	Default value: one transaction per microframe.
2	2_TRANS	Two transactions per microframe. This endpoint should be configured as double-bank.
3	3_TRANS	Three transactions per microframe. This endpoint should be configured as triple-bank.

Bits 12:11 – EPTYPE[1:0] Endpoint Type
This field should be written to select the endpoint type:

This field is cleared upon receiving a USB reset.

Value	Description
0	Disables the ping protocol.
1	Enables the ping mechanism according to the USB 2.0 Standard.

Bits 19:16 – PEPNUM[3:0] Pipe Endpoint Number
This field contains the number of the endpoint targeted by the pipe. This value is from 0 to 10.

This field is cleared upon sending a USB reset.

Bits 13:12 – PTYPE[1:0] Pipe Type
This field contains the pipe type.

This field is cleared upon sending a USB reset.

Value	Name	Description
0	CTRL	Control
1	Reserved	
2	BLK	Bulk
3	Reserved	

Bit 10 – AUTOSW Automatic Switch
This bit is cleared upon sending a USB reset.

Value	Description
0	The automatic bank switching is disabled.
1	The automatic bank switching is enabled.

Bits 9:8 – PTOKEN[1:0] Pipe Token
This field contains the pipe token.

Value	Name	Description
0	SETUP	SETUP
1	IN	IN
2	OUT	OUT
3	Reserved	

Bits 6:4 – PSIZE[2:0] Pipe Size
This field contains the size of each pipe bank.

This field is cleared upon sending a USB reset.

Value	Name	Description
0	8_BYTE	8 bytes
1	16_BYTE	16 bytes
2	32_BYTE	32 bytes
3	64_BYTE	64 bytes
4	128_BYTE	128 bytes
5	256_BYTE	256 bytes
6	512_BYTE	512 bytes
7	1024_BYTE	1024 bytes

43.7.14 TWIHS Transmit Holding Register

Name: TWIHS_THR
Offset: 0x34
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

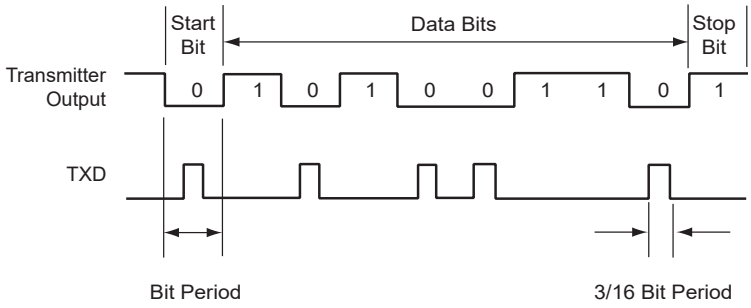
Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TXDATA[7:0] Master or Slave Transmit Holding Data

Baud Rate	Pulse Duration (3/16)
57.6 kbit/s	3.26 µs
115.2 kbit/s	1.63 µs

The following figure shows an example of character transmission.

Figure 46-33. IrDA Modulation



46.6.5.2 IrDA Baud Rate

The following table provides examples of CD values, baud rate error, and pulse duration. Note that the requirement on the maximum acceptable error of ±1.87% must be met.

Table 46-10. IrDA Baud Rate Error

Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (µs)
3,686,400	115,200	2	0.00%	1.63
20,000,000	115,200	11	1.38%	1.63
32,768,000	115,200	18	1.25%	1.63
40,000,000	115,200	22	1.38%	1.63
3,686,400	57,600	4	0.00%	3.26
20,000,000	57,600	22	1.38%	3.26
32,768,000	57,600	36	1.25%	3.26
40,000,000	57,600	43	0.93%	3.26
3,686,400	38,400	6	0.00%	4.88
20,000,000	38,400	33	1.38%	4.88
32,768,000	38,400	53	0.63%	4.88
40,000,000	38,400	65	0.16%	4.88
3,686,400	19,200	12	0.00%	9.77
20,000,000	19,200	65	0.16%	9.77
32,768,000	19,200	107	0.31%	9.77
40,000,000	19,200	130	0.16%	9.77
3,686,400	9,600	24	0.00%	19.53

USART Pin	V24	CCITT	Direction
RXD	3	104	From modem to terminal
CTS	5	106	From terminal to modem
DSR	6	107	From terminal to modem
DCD	8	109	From terminal to modem
RI	22	125	From terminal to modem

The control of the DTR output pin is performed by writing a ‘1’ to US_CR.DTRDIS and US_CR.DTREN. The disable command forces the corresponding pin to its inactive level, i.e., high. The enable command forces the corresponding pin to its active level, i.e., low. The RTS output pin is automatically controlled in this mode.

The level changes are detected on the RI, DSR, DCD and CTS pins. If an input change is detected, the bits RIIC, DSRIC, DCDIC and CTSIC in US_CSR are set and can trigger an interrupt. The status is automatically cleared when US_CSR is read. Furthermore, the CTS automatically disables the transmitter when it is detected at its inactive state. If a character is being transmitted when the CTS rises, the character transmission is completed before the transmitter is actually disabled.

46.6.8 SPI Mode

The Serial Peripheral Interface (SPI) mode is a synchronous serial data link that provides communication with external devices in Master or Slave mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master” which controls the data flow, while the other devices act as “slaves" which have data shifted into and out by the master. Different CPUs can take turns being masters and one master may simultaneously shift data into multiple slaves. (Multiple master protocol is the opposite of single master protocol, where one CPU is always the master while all of the others are always slaves.) However, only one slave may drive its output to write data back to the master at any given time.

A slave device is selected when its NSS signal is asserted by the master. The USART in SPI Master mode can address only one SPI slave because it can generate only one NSS signal.

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This data line supplies the output data from the master shifted into the input of the slave.
- Master In Slave Out (MISO): This data line supplies the output data from a slave to the input of the master.
- Serial Clock (SCK): This control line is driven by the master and regulates the flow of the data bits. The master may transmit data at a variety of baud rates. The SCK line cycles once for each bit that is transmitted.
- Slave Select (NSS): This control line allows the master to select or deselect the slave.

46.6.8.1 Modes of Operation

The USART can operate in SPI Master mode or in SPI Slave mode.

SPI Master mode is enabled by writing 0xE to US_MR.USART_MODE. In this case, the SPI lines must be connected as described below:

Figure 46-46. Master Node Configuration, NACT = SUBSCRIBE

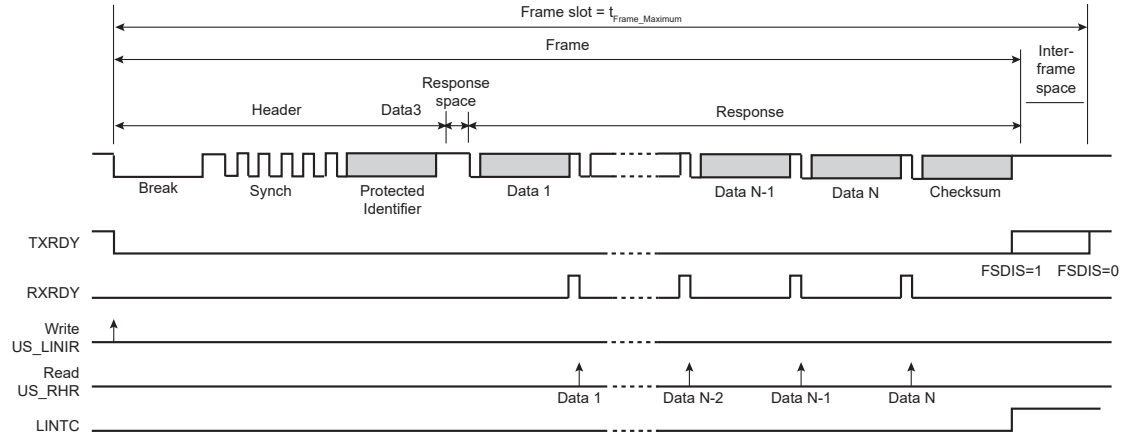
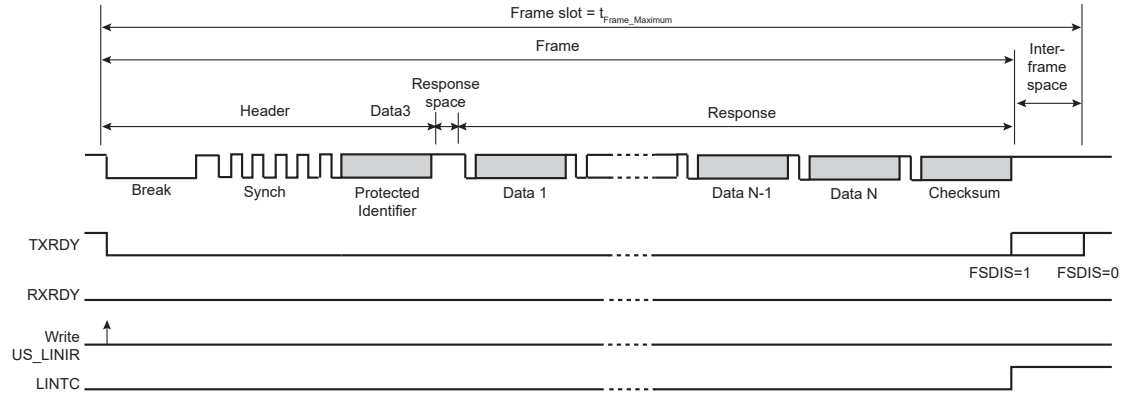


Figure 46-47. Master Node Configuration, NACT = IGNORE



46.6.9.15.2 Slave Node Configuration

- Write TXEN and RXEN in US_CR to enable both the transmitter and the receiver.
- Write USART_MODE in US_MR to select the LIN mode and the slave node configuration.
- Write CD and FP in US_BRGR to configure the baud rate.
- Wait until LINID in US_CSR rises.
- Check LINISFE and LINPE errors.
- Read IDCHR in US_RHR.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM and DLC in US_LINMR to configure the frame transfer.

IMPORTANT: If the NACT configuration for this frame is PUBLISH, the US_LINMR must be written with NACT = PUBLISH even if this field is already correctly configured, in order to set the TXREADY flag and the corresponding write transfer request.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the LIN controller sends the response
  - Wait until TXRDY in US_CSR rises.
  - Write TCHR in US_THR to send a byte.
  - If all the data have not been written, redo the two previous steps.
  - Wait until LINTC in US_CSR rises.
  - Check the LIN errors.

Figure 48-11. Synchronous Data Tx Device Protocol

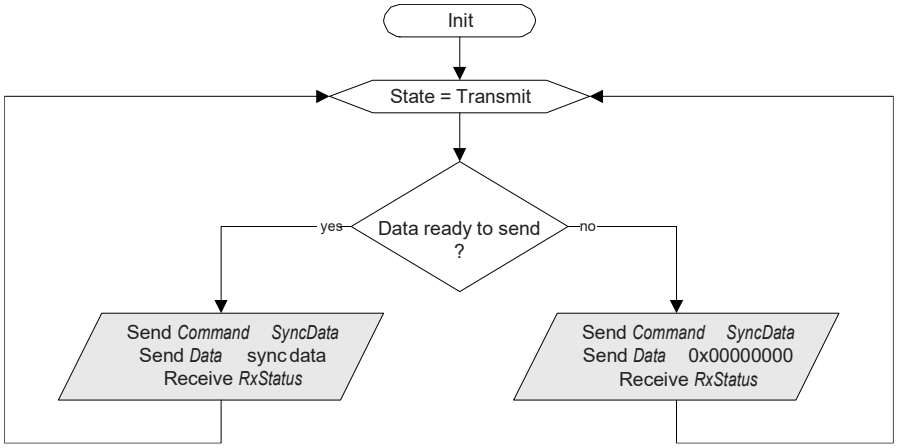
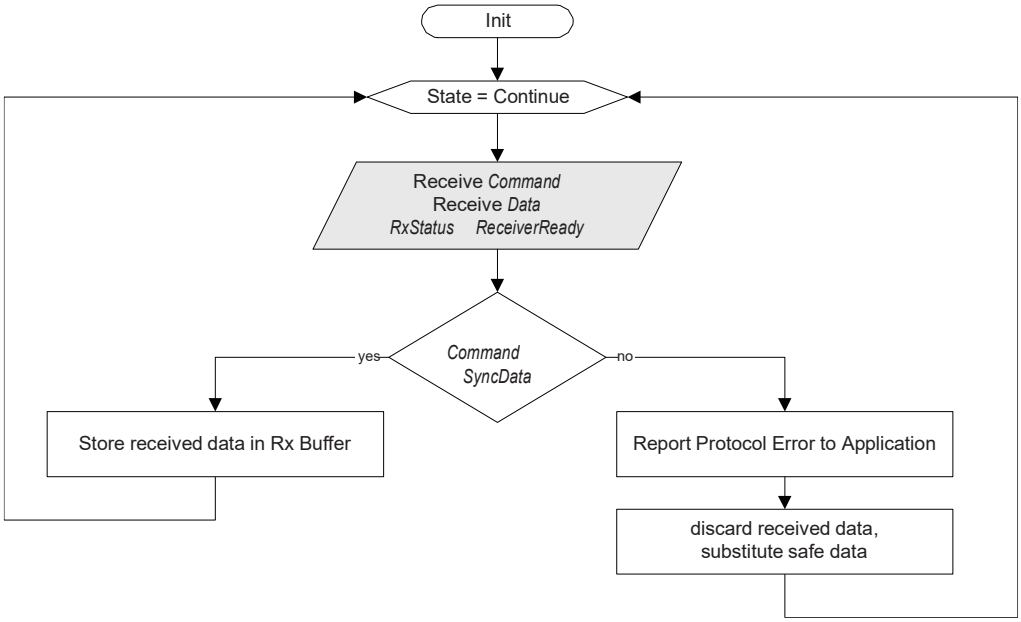


Figure 48-12. Synchronous Data Rx Device Protocol



Isochronous

Isochronous data is sent in a streaming fashion, similar to synchronous data. However, the isochronous commands indicate the start of a block and how many bytes are valid in the concurrent transmitted quadlet. Valid bytes are left-justified in the quadlet, as illustrated in Figure 48-13. When isochronous data is being transported (channel active), but no data is available for the current quadlet, the IsoNoData command is sent by the Tx Device.

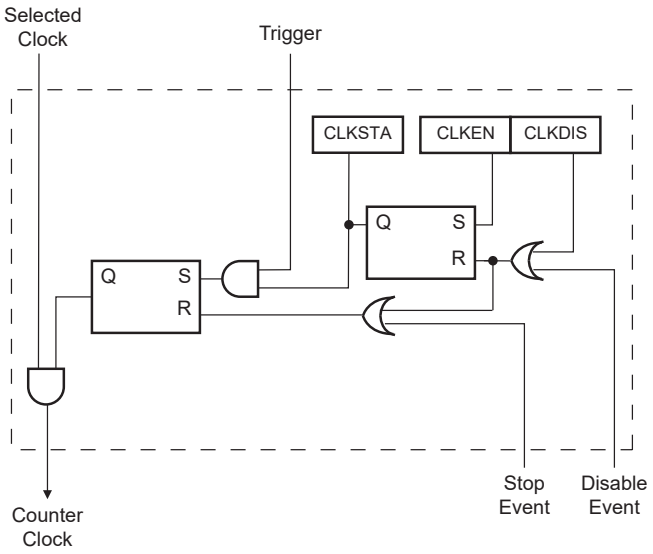
Table 48-15. Synchronous CDT Entry Field Definitions

Field	Description	Details	Accessibility
BA	Buffer Base Address	- BA can start at any byte in the 16k DBR	r,w
BD	Buffer Depth	- BD = size of buffer in bytes - 1 - Buffer end address = BA + BD - BD = 4 x m x bpf - 1, where: m = frames per sub-buffer (for MFE = 0, m = 1) bpf = bytes per frame.	r,w
RPTR	Read Pointer	- Software initializes to zero, hardware updates - Counts the read address offset within a buffer - DMA read address = BA + RPTR	r,w,u(1)
WPTR	Write Pointer	- Software initializes to zero, hardware updates - Counts the write address offset within a buffer - DMA write address = BA + WPTR	r,w,u (1)
RSBC	Read Sub-buffer Counter	- Software initializes to zero, hardware updates - Counts the read sub-buffer offset - DMA uses for pointer management	r,w,u (1)
WSBC	Write Sub-buffer Counter	- Software initializes to zero, hardware updates - Counts the write sub-buffer offset - DMA uses for pointer management	r,w,u (1)
RSTS	Read Status	- Software initializes to zero, hardware updates - RSTS states:(2) xxx0 = normal operation (no mute) xxx1 = normal operation (mute) xx0x = idle	r,w,u (1)
WSTS	Write Status	- Software initializes to zero, hardware updates - WSTS states:(2) xxx0 = normal operation (no mute) xxx1 = normal operation (mute) xx0x = idle 1xxx = command protocol error	r,w,u (1)
Reserved	Reserved	- Software writes a zero to all reserved bits when the entry is initialized. The reserved bits are Read-only after initialization.	r,w,u (1)

Notes: 1. “u” means “Updated periodically by hardware”.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the Channel Control register (TC_CCR). In Capture mode it can be disabled by an RB load event if TC_CMRx.LDBDIS is set to ‘1’. In Waveform mode, it can be disabled by an RC Compare event if TC_CMRx.CPCDIS is set to ‘1’. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can reenable the clock. When the clock is enabled, TC_SR.CLKSTA is set.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (TC_CMRx.LDBSTOP = 1) or an RC compare event in Waveform mode (TC_CMRx.CPCSTOP = 1). The start and the stop commands are effective only if the clock is enabled.

Figure 50-4. Clock Control



50.6.5 Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with TC_CMRx.WAVE.

In Capture mode, TIOAx and TIOBx are configured as inputs.

In Waveform mode, TIOAx is always configured to be an output and TIOBx is an output if it is not selected to be the external trigger.

50.6.6 Trigger

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

The following triggers are common to both modes:

- Software Trigger: Each channel has a software trigger, available by setting TC_CCR.SWTRG.

51.7.21 PWM Output Selection Clear Register

Name: PWM_OSC
Offset: 0x50
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					OSCL3	OSCL2	OSCL1	OSCL0
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
					OSCH3	OSCH2	OSCH1	OSCH0
Access					W	W	W	W
Reset					0	0	0	–

Bits 16, 17, 18, 19 – OSCLx Output Selection Clear for PWML output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOLx selected as PWML output of channel x.

Bits 0, 1, 2, 3 – OSCHx Output Selection Clear for PWMH output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOHx selected as PWMH output of channel x.

If conversion results are signed and resolution is less than 16 bits, the sign is extended up to the bit 15 (e.g., 0xF43 for 12-bit resolution will be read as 0xFF43 and 0x467 will be read as 0x0467).

52.6.6 Conversion Triggers

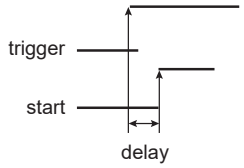
Conversions of the active analog channels are started with a software or hardware trigger. The software trigger is provided by writing a '1' to the bit START in the Control Register (AFEC_CR).

The hardware trigger can be one of the TIOA outputs of the Timer Counter channels, PWM Event line, or the external trigger input of the AFEC (ADTRG). The hardware trigger is selected with AFEC_MR.TRGSEL. The selected hardware trigger is enabled with AFEC_MR.TRGEN

The minimum time between two consecutive trigger events must be strictly greater than the duration of the longest conversion sequence according to configuration of registers AFEC_MR, AFEC_CHSR, AFEC_SEQ1R, AFEC_SEQ2R.

If a hardware trigger is selected, the start of a conversion is triggered after a delay starting at each rising edge of the selected signal. Due to asynchronous handling, the delay may vary in a range of two peripheral clock periods to one AFE clock period. This delay varies from trigger to trigger and so introduces a jitter error leading to a reduced Signal-to-Noise ratio performance.

Figure 52-6. Conversion Start with the Hardware Trigger



If one of the TIOA outputs is selected, the corresponding Timer Counter channel must be programmed in Waveform mode.

Only one start command is necessary to initiate a conversion sequence on all the channels. The AFEC hardware logic automatically performs the conversions on the active channels, then waits for a new request. The Channel Enable (AFEC_CHER) and Channel Disable (AFEC_CHDR) registers permit the analog channels to be enabled or disabled independently.

If the AFEC is used with a DMA, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

52.6.7 Sleep Mode and Conversion Sequencer

The AFEC Sleep mode maximizes power saving by automatically deactivating the AFE when it is not being used for conversions. Sleep mode is selected by setting AFEC_MR.SLEEP.

Sleep mode is managed by a conversion sequencer, which automatically processes the conversions of all channels at lowest power consumption.

This mode can be used when the minimum period of time between two successive trigger events is greater than the startup period of the AFEC. Refer to the AFE Characteristics in the section "Electrical Characteristics".

When a start conversion request occurs, the AFE is automatically activated. As the analog cell requires a startup time, the logic waits during this lapse and starts the conversion on the enabled channels. When all conversions are complete, the AFE is deactivated until the next trigger. Triggers occurring during the sequence are not taken into account.

A fast wakeup mode is available in the AFEC_MR as a compromise between power-saving strategy and responsiveness. Setting the FWUP bit enables the Fast Wakeup mode. In Fast Wakeup mode, the AFE is

55.6.9 ICM Descriptor Area Start Address Register

Name: ICM_DSCR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DASA[25:18]							
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

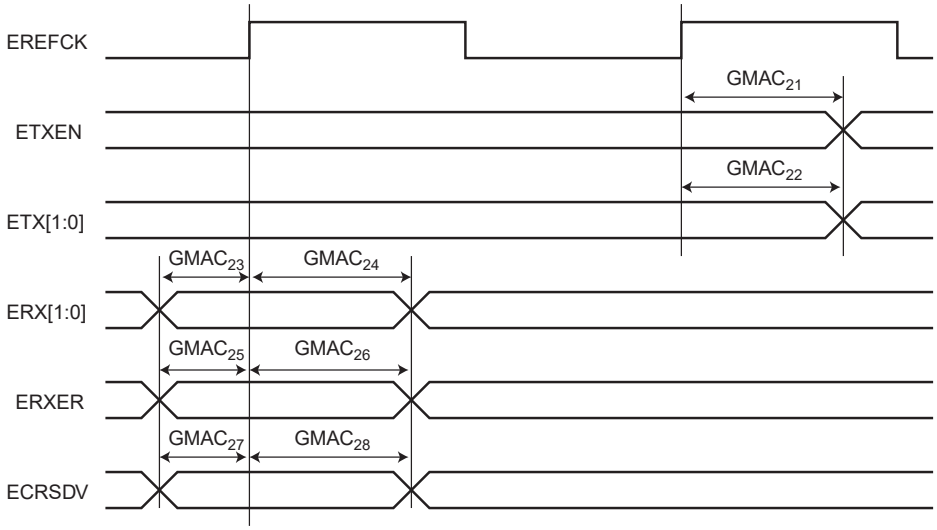
Bit	23	22	21	20	19	18	17	16
	DASA[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DASA[9:2]							
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
	DASA[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bits 31:6 – DASA[25:0] Descriptor Area Start Address
The start address is a multiple of the total size of the data structure (64 bytes).

Figure 58-32. GMAC RMII Mode Signals



58.13.1.14 SSC Timings

58.13.1.14.1 Timing Conditions

Timings are given assuming the load capacitance in the following table.

Table 58-73. Load Capacitance

Supply	C_L Max
3.3V	30 pF
1.8V	20 pF

58.13.1.14.2 Timing Extraction

Figure 58-33. SSC Transmitter, TK and TF in Output