Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, SPI, SSC, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	44
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 5x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70j19a-ant

Offset	Name	Bit Pos.								
		15:8			M3PR[1:0]				M2PR[1:0]	
		23:16			M5PR[1:0]				M4PR[1:0]	
		31:24			M7PR[1:0]				M6PR[1:0]	
0xAC	MATRIX_PRBS5	7:0			M9PR[1:0]				M8PR[1:0]	
		15:8			M11PR[1:0]				M10PR[1:0]	
		23:16							M12PR[1:0]	
		31:24								
0xB0	MATRIX_PRAS6	7:0			M1PR[1:0]				M0PR[1:0]	
		15:8			M3PR[1:0]				M2PR[1:0]	
		23:16			M5PR[1:0]				M4PR[1:0]	
		31:24			M7PR[1:0]				M6PR[1:0]	
0xB4	MATRIX_PRBS6	7:0			M9PR[1:0]				M8PR[1:0]	
		15:8			M11PR[1:0]				M10PR[1:0]	
		23:16							M12PR[1:0]	
		31:24								
0xB8	MATRIX_PRAS7	7:0			M1PR[1:0]				M0PR[1:0]	
		15:8			M3PR[1:0]				M2PR[1:0]	
		23:16			M5PR[1:0]				M4PR[1:0]	
		31:24			M7PR[1:0]				M6PR[1:0]	
0xBC	MATRIX_PRBS7	7:0			M9PR[1:0]				M8PR[1:0]	
		15:8			M11PR[1:0]				M10PR[1:0]	
		23:16							M12PR[1:0]	
		31:24								
0xC0	MATRIX_PRAS8	7:0			M1PR[1:0]				M0PR[1:0]	
		15:8			M3PR[1:0]				M2PR[1:0]	
		23:16			M5PR[1:0]				M4PR[1:0]	
		31:24			M7PR[1:0]				M6PR[1:0]	
0xC4	MATRIX_PRBS8	7:0			M9PR[1:0]				M8PR[1:0]	
		15:8			M11PR[1:0]				M10PR[1:0]	
		23:16							M12PR[1:0]	
		31:24								
0xC8 ... 0xFF	Reserved									
0x0100	MATRIX_MRCR	7:0	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
		15:8				RCB12	RCB11	RCB10	RCB9	RCB8
		23:16								
		31:24								
0x0104 ... 0x010F	Reserved									
0x0110	CCFG_CAN0	7:0	Reserved[7:0]							
		15:8								Reserved[8:8]
		23:16	CAN0DMABA[7:0]							
		31:24	CAN0DMABA[15:8]							
0x0114	CCFG_SYSIO	7:0	SYSIO7	SYSIO6	SYSIO5	SYSIO4				

Command	Value	Mnemonic
Get GPNVM Bit	0x0D	GGPB
Start Read Unique Identifier	0x0E	STUI
Stop Read Unique Identifier	0x0F	SPUI
Get CALIB Bit	0x10	GCALB
Erase Sector	0x11	ES
Write User Signature	0x12	WUS
Erase User Signature	0x13	EUS
Start Read User Signature	0x14	STUS
Stop Read User Signature	0x15	SPUS

To execute one of these commands, select the required command using the FCMD field in the Flash Command register (EEFC_FCR). As soon as EEFC_FCR is written, the FRDY flag and the FVALUE field in the Flash Result register (EEFC_FRR) are automatically cleared. Once the current command has completed, the FRDY flag is automatically set. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated. (Note that this is true for all commands except for the STUI command. The FRDY flag is not set when the STUI command has completed.)

All the commands are protected by the same keyword, which must be written in the eight highest bits of EEFC_FCR.

Writing EEFC_FCR with data that does not contain the correct key and/or with an invalid command has no effect on the whole memory plane, but the FCMDE flag is set in the Flash Status register (EEFC_FSR). This flag is automatically cleared by a read access to EEFC_FSR.

When the current command writes or erases a page in a locked region, the command has no effect on the whole memory plane, but the FLOCKE flag is set in EEFC_FSR. This flag is automatically cleared by a read access to EEFC_FSR.

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

Bits 20:16 – MONTH[4:0] Month Alarm
This field is the alarm field corresponding to the BCD-coded month counter.

- The frequency of peripheral clock must be strictly superior to two times the frequency of the clock of the device which generates the parallel data.

32.5.14.4 Programming Sequence

32.5.14.4.1 Without DMA

1. Write PIO_PCIDR and PIO_PCIER in order to configure the Parallel Capture mode interrupt mask.
2. Write PIO_PCMR to set the fields DSIZE, ALWYS, HALFS and FRSTS in order to configure the Parallel Capture mode WITHOUT enabling the Parallel Capture mode.
3. Write PIO_PCMR to set the PCEN bit to one in order to enable the Parallel Capture mode WITHOUT changing the previous configuration.
4. Wait for a data ready by polling the DRDY flag in PIO_PCISR or by waiting for the corresponding interrupt.
5. Check OVRE flag in PIO_PCISR.
6. Read the data in PIO_PCRHR.
7. If new data are expected, go to step 4.
8. Write PIO_PCMR to set the PCEN bit to zero in order to disable the Parallel Capture mode WITHOUT changing the previous configuration.

32.5.14.4.2 With DMA

1. Write PIO_PCIDR and PIO_PCIER in order to configure the Parallel Capture mode interrupt mask.
2. Configure DMA transfer in DMA registers.
3. Write PIO_PCMR to set the fields DSIZE, ALWYS, HALFS and FRSTS in order to configure the Parallel Capture mode WITHOUT enabling the Parallel Capture mode.
4. Write PIO_PCMR to set PCEN bit to one in order to enable the Parallel Capture mode WITHOUT changing the previous configuration.
5. Wait for the DMA status flag to indicate that the buffer transfer is complete.
6. Check OVRE flag in PIO_PCISR.
7. If a new buffer transfer is expected, go to step 5.
8. Write PIO_PCMR to set the PCEN bit to zero in order to disable the Parallel Capture mode WITHOUT changing the previous configuration.

32.5.15 I/O Lines Programming Example

The programming example shown in the following table is used to obtain the following configuration:

- 4-bit output port on I/O lines 0 to 3 (should be written in a single write operation), open-drain, with pullup resistor
- Four output signals on I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pullup resistor, no pulldown resistor
- Four input signals on I/O lines 8 to 11 (to read push-button states for example), with pullup resistors, glitch filters and input change interrupts
- Four input signals on I/O line 12 to 15 to read an external device status (polled, thus no input change interrupt), no pullup resistor, no glitch filter
- I/O lines 16 to 19 assigned to peripheral A functions with pullup resistor
- I/O lines 20 to 23 assigned to peripheral B functions with pulldown resistor
- I/O lines 24 to 27 assigned to peripheral C with input change interrupt, no pullup resistor and no pulldown resistor
- I/O lines 28 to 31 assigned to peripheral D, no pullup resistor and no pulldown resistor

37.6.1 ISI Configuration 1 Register

Name: ISI_CFG1
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SFD[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
	SLD[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
		THMASK[1:0]		FULL	DISCR	FRATE[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRC_SYNC	EMB_SYNC	GRAYLE	PIXCLK_POL	VSYNC_POL	HSYNC_POL		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:24 – SFD[7:0] Start of Frame Delay
SFD lines are skipped at the beginning of the frame.

Bits 23:16 – SLD[7:0] Start of Line Delay
SLD pixel clock periods to wait before the beginning of a line.

Bits 14:13 – THMASK[1:0] Threshold Mask

Value	Name	Description
0	BEATS_4	Only 4 beats AHB burst allowed
1	BEATS_8	Only 4 and 8 beats AHB burst allowed
2	BEATS_16	4, 8 and 16 beats AHB burst allowed

Bit 12 – FULL Full Mode is Allowed

Value	Description
0	The codec frame is transferred to memory when an available frame slot is detected.
1	Both preview and codec DMA channels are operating simultaneously.

Bit 11 – DISCR Disable Codec Request

37.6.21 DMA Codec Base Address Register

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

Bit	31	30	29	28	27	26	25	24
	C_ADDR[29:22]							
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
	C_ADDR[21:14]							
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
	C_ADDR[13:6]							
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
	C_ADDR[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – C_ADDR[29:0] Codec Image Base Address
This address is word-aligned.

Offset	Name	Bit Pos.								
0x01E8	GMAC_EFRSL	7:0	RUD[7:0]							
		15:8	RUD[15:8]							
		23:16	RUD[23:16]							
		31:24	RUD[31:24]							
0x01EC	GMAC_EFRN	7:0	RUD[7:0]							
		15:8	RUD[15:8]							
		23:16	RUD[23:16]							
		31:24			RUD[29:24]					
0x01F0	GMAC_PEFTSL	7:0	RUD[7:0]							
		15:8	RUD[15:8]							
		23:16	RUD[23:16]							
		31:24	RUD[31:24]							
0x01F4	GMAC_PEFTN	7:0	RUD[7:0]							
		15:8	RUD[15:8]							
		23:16	RUD[23:16]							
		31:24			RUD[29:24]					
0x01F8	GMAC_PEFRSL	7:0	RUD[7:0]							
		15:8	RUD[15:8]							
		23:16	RUD[23:16]							
		31:24	RUD[31:24]							
0x01FC	GMAC_PEFRN	7:0	RUD[7:0]							
		15:8	RUD[15:8]							
		23:16	RUD[23:16]							
		31:24			RUD[29:24]					
0x0200 ... 0x026F	Reserved									
0x0270	GMAC_RXLPI	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16								
		31:24								
0x0274	GMAC_RXLPITIME	7:0	LPITIME[7:0]							
		15:8	LPITIME[15:8]							
		23:16	LPITIME[23:16]							
		31:24								
0x0278	GMAC_TXLPI	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24								
0x027C	GMAC_TXLPITIME	7:0	LPITIME[7:0]							
		15:8	LPITIME[15:8]							
		23:16	LPITIME[23:16]							
		31:24								
0x0280 ... 0x03FF	Reserved									

38.8.10 GMAC Interrupt Status Register

Name: GMAC_ISR
Offset: 0x024
Reset: 0x00000000
Property: Read-only

This register indicates the source of the interrupt. An interrupt source must be enabled in the mask register first so the corresponding bits of this register will be set and the GMAC interrupt signal will be asserted in the system.

Bit	31	30	29	28	27	26	25	24
			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8
		PFTR	PTZ	PFNZ	HRESP	ROVR		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 29 – TSUTIMCMP TSU Timer Comparison
Indicates when TSU timer count value is equal to programmed value.

Cleared on read.

Bit 28 – WOL Wake On LAN
WOL interrupt. Indicates a WOL message has been received.

Bit 27 – RXLPISBC Receive LPI indication Status Bit Change
Receive LPI indication status bit change.

Cleared on read.

Bit 26 – SRI TSU Seconds Register Increment
Indicates the register has incremented.

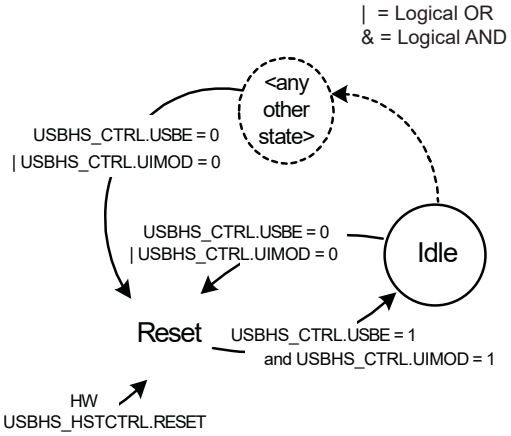
Cleared on read.

Bit 25 – PDRSFT PDelay Response Frame Transmitted
Indicates a PTP pdelay_resp frame has been transmitted.

39.5.2.2 Power-On and Reset

The following figure describes the USBHS Device mode main states.

Figure 39-7. Device Mode Main States



After a hardware reset, the USBHS Device mode is in Reset state. In this state:

- the USBHS clock is stopped to minimize power consumption (USBHS_CTRL.FRZCLK = 1),
- the internal registers of the Device mode are reset,
- the endpoint banks are de-allocated,
- neither D+ nor D- is pulled up (USBHS_DEVCTRL.DETACH = 1).

D+ or D- is pulled up according to the selected speed as soon as the USBHS_DEVCTRL.DETACH bit is written to zero. See "Device Mode" for further details.

When the USBHS is enabled (USBHS_CTRL.USBE = 1) in Device mode (USBHS_CTRL.UIMOD = 1), its Device mode state enters Idle state with minimal power consumption. This does not require the USB clock to be activated.

The USBHS Device mode can be disabled and reset at any time by disabling the USBHS (by writing a zero to USBHS_CTRL.USBE) or when the Host mode is enabled (USBHS_CTRL.UIMOD = 0).

39.5.2.3 USB Reset

The USB bus reset is managed by hardware. It is initiated by a connected host.

When a USB reset is detected on the USB line, the following operations are performed by the controller:

- All endpoints are disabled, except the default control endpoint.
- The default control endpoint is reset (see 39.5.2.4 Endpoint Reset for more details).
- The data toggle sequence of the default control endpoint is cleared.
- At the end of the reset process, the End of Reset (USBHS_DEVISR.EORST) bit is set.
- During a reset, the USBHS automatically switches to High-speed mode if the host is High-speed-capable (the reset is called High-speed reset). The user should observe the USBHS_SR.SPEED field to know the speed running at the end of the reset (USBHS_DEVISR.EORST = 1).

39.5.2.4 Endpoint Reset

An endpoint can be reset at any time by writing a one to the Endpoint x Reset bit USBHS_DEVEPT.EPRSTx. This is recommended before using an endpoint upon hardware reset or when a USB bus reset has been received. This resets:

- the internal state machine of the endpoint,

39.6.9 Device Global Interrupt Mask Register

Name: USBHS_DEVIMR
Offset: 0x0010
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
	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
Access								
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
			PEP_9	PEP_8	PEP_7	PEP_6	PEP_5	PEP_4
Access								
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PEP_3	PEP_2	PEP_1	PEP_0				
Access								
Reset	0	0	0	0				

Bit	7	6	5	4	3	2	1	0
		UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	MSOFE	SUSPE
Access								
Reset		0	0	0	0	0	0	0

Bits 25, 26, 27, 28, 29, 30, 31 – DMA_ DMA Channel x Interrupt Mask

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 – PEP_ Endpoint x Interrupt Mask

Bit 6 – UPRSME Upstream Resume Interrupt Mask

Bit 5 – EORSME End of Resume Interrupt Mask

Bit 4 – WAKEUPE Wakeup Interrupt Mask

Bit 3 – EORSTE End of Reset Interrupt Mask

Bit 2 – SOFE Start of Frame Interrupt Mask

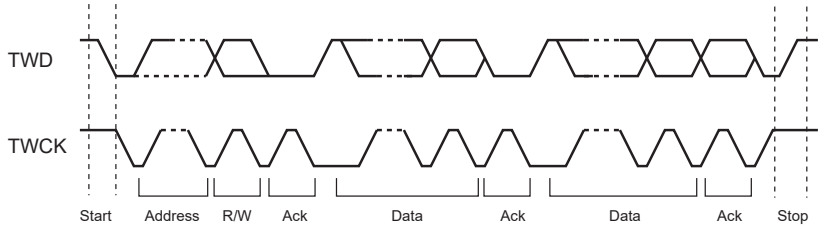
Bit 1 – MSOFE Micro Start of Frame Interrupt Mask

Bit 0 – SUSPE Suspend Interrupt Mask

Value	Name	Description
6	65536	CSTOCYC x 65536
7	1048576	CSTOCYC x 1048576

Bits 3:0 – CSTOCYC[3:0] Completion Signal Timeout Cycle Number
This field determines the maximum number of Master Clock cycles that the HSMCI waits between two data block transfers. Its value is calculated by (CSTOCYC x Multiplier).

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

Arbitration starts as soon as two or more masters place information on the bus at the same time, and stops (arbitration is lost) for the master that intends to send a logical one while the other master sends a logical zero.

As soon as arbitration is lost by a master, it stops sending data and listens to the bus in order to detect a stop. When the stop is detected, the master that has lost arbitration may put its data on the bus by respecting arbitration.

Arbitration is illustrated in Arbitration Cases.

43.6.4.2 Different Multimaster Modes

Two Multimaster modes may be distinguished:

1. The TWIHS is considered as a master only and is never addressed.
2. The TWIHS may be either a master or a slave and may be addressed.

Note: Arbitration in supported in both Multimaster modes.

43.6.4.2.1 TWIHS as Master Only

In this mode, the TWIHS is considered as a master only (MSEN is always at one) and must be driven like a master with the ARBLST (Arbitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the user must reinitiate the data transfer.

If starting a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWIHS automatically waits for a STOP condition on the bus to initiate the transfer (see User Sends Data While the Bus is Busy).

Note: The state of the bus (busy or free) is not indicated in the user interface.

43.6.4.2.2 TWIHS as Master or Slave

The automatic reversal from master to slave is not supported in case of a lost arbitration.

Then, in the case where TWIHS may be either a master or a slave, the user must manage the pseudo Multimaster mode described in the steps below:

1. Program the TWIHS in Slave mode (SADR + MSDIS + SVEN) and perform a slave access (if TWIHS is addressed).
2. If the TWIHS has to be set in Master mode, wait until TXCOMP flag is at 1.
3. Program the Master mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as the Master mode is enabled, the TWIHS scans the bus in order to detect if it is busy or free. When the bus is considered free, the TWIHS initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is set to 1), the user must program the TWIHS in Slave mode in case the master that won the arbitration needs to access the TWIHS.
7. If the TWIHS has to be set in Slave mode, wait until the TXCOMP flag is at 1 and then program the Slave mode.

Note: If the arbitration is lost and the TWIHS is addressed, the TWIHS does not acknowledge, even if it is programmed in Slave mode as soon as ARBLST is set to 1. Then the master must repeat SADR.

43.7.12 TWIHS Receive Holding Register

Name: TWIHS_RHR
Offset: 0x30
Reset: 0x00000000
Property: Read-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
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXDATA[7:0] Master or Slave Receive Holding Data

Figure 44-3. Audio Application Block Diagram

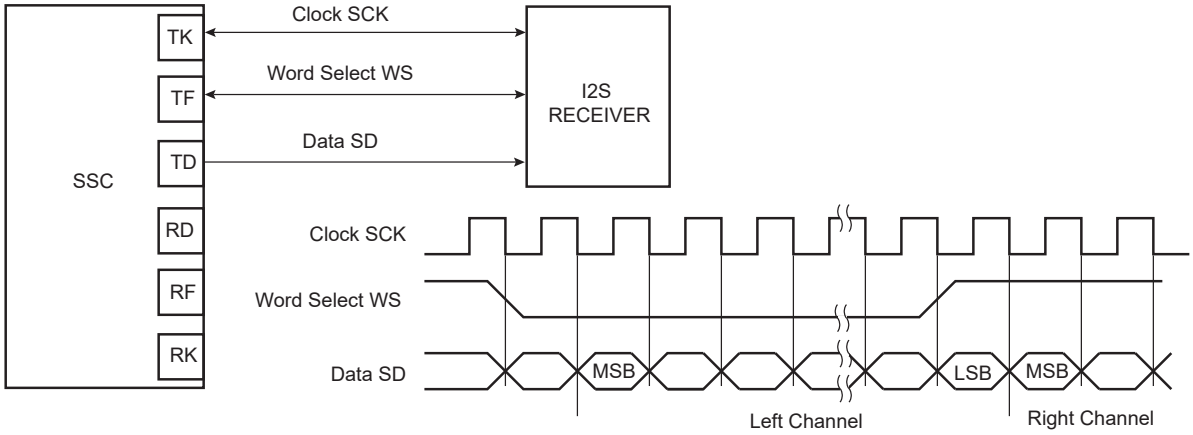


Figure 44-4. Codec Application Block Diagram

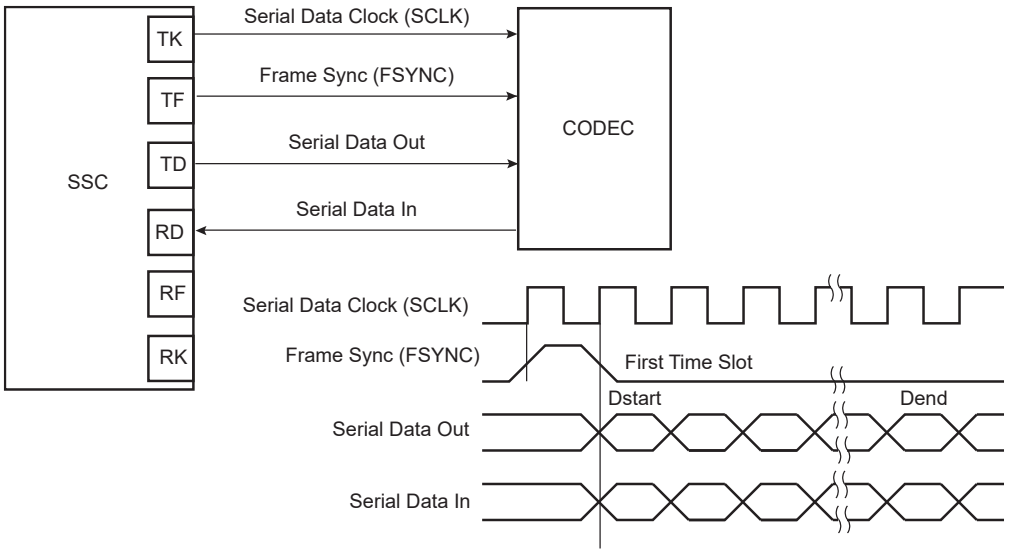


Table 48-5. MediaLB RxStatus Responses

Value (see Note)	Command	Description
Normal Commands (TX Device sends in non-system channels):		
00h	NoData	No data to send out in this physical channel.
02h...0Eh	rsvd	Reserved
10h	SyncData	Tx Device sends out SyncData command to indicate synchronous stream data.
12h...1Eh	rsvd	Reserved
20h	AsyncStart	Asynchronous logical channel. Start of a packet.
22h	AsyncContinue	Asynchronous logical channel. Middle of a packet.
24h	AsyncEnd	Asynchronous logical channel. End of a packet.
26h	AsyncBreak	Asynchronous logical channel. Indicates a packet stop. No valid data present on the MLBD line.
28h...2Eh	rsvd	Reserved
30h	ControlStart	Control logical channel. Start of a message.
32h	ControlContinue	Control logical channel. Middle of a message.
34h	ControlEnd	Control logical channel. End of a message.
36h	ControlBreak	Control logical channel. Indicates a message stop. No valid data present on the MLBD line.
38h...3Eh	rsvd	Reserved
40h	IsoNoData	Isochronous logical channel, no data valid.
42h	Iso1Byte	Isochronous logical channel, one data byte valid. First byte (MSB) transmitted/received is valid. Last three bytes in physical channel are empty.
44h	Iso2Bytes	Isochronous logical channel, first two data bytes valid. First byte transmitted/received is the MSB. Last two bytes in physical channel are empty.
46h	Iso3Bytes	Isochronous logical channel, first three data bytes valid. First byte transmitted/received is the MSB. Last byte in physical channel is empty.
48h	Iso4Bytes	Isochronous logical channel, all four data bytes valid. First byte transmitted/received is the MSB.
4Ah...4Eh	rsvd	Reserved
50h	IsoSync1Byte	Isochronous logical channel, one data byte valid and start of a block. First byte transmitted/received is valid. Last three bytes in physical channel are empty.

49.6.42 MCAN Transmit Buffer Cancellation Finished

Name: MCAN_TXBCF
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
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
	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
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
	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
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
	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
Access	R	R	R	R	R	R	R	R
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 – CFx Cancellation Finished for Transmit Buffer x
Each Transmit Buffer has its own Cancellation Finished bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a cancellation was requested via MCAN_TXBCR. In case the corresponding MCAN_TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit of register MCAN_TXBAR.

Value	Description
0	No transmit buffer cancellation.
1	Transmit buffer cancellation finished.

53.7.8 DACC Interrupt Enable Register

Name: DACC_IER
Offset: 0x24
Reset: –
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
			EOC1	EOC0			TXRDY1	TXRDY0
Access			W	W			W	W
Reset			0	–			0	–

Bits 4, 5 – EOCx End of Conversion Interrupt Enable of channel x

Bits 0, 1 – TXRDYx Transmit Ready Interrupt Enable of channel x

Table 58-14. Typical Sleep Mode Current Consumption vs. Master Clock (MCK) Variation with Fast RC

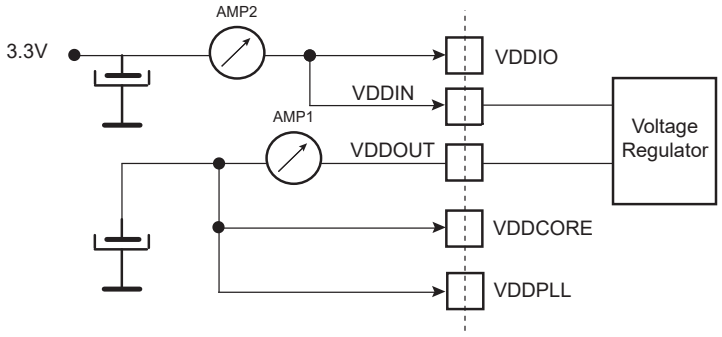
Core Clock/MCK (MHz)	VDDCORE Consumption (AMP1)	Total Consumption (AMP2)	Unit	Wakeup Time	Unit
12	2.0	2.0	mA	12	µs
8	1.5	1.5		18	
4	1.0	1.1		31	
2	0.8	0.8		62	
1	0.6	0.7		123	
0.5	0.6	0.6		247	
0.25	0.5	0.5		494	

58.3.3 Wait Mode Current Consumption and Wakeup Time

The Wait mode configuration and measurements are defined as follows:

- Core clock and Master clock stopped
- Current measurement as shown below
- All peripheral clocks deactivated
- BOD disabled
- RTT enabled

Figure 58-7. Measurement Setup for Wait Mode



The following tables give current consumption and wakeup time(1) in Wait mode.

Table 58-15. Typical Current Consumption in Wait Mode

Wait Mode Consumption	Typical Value				Unit
	at 25°C		at 85°C	at 105°C	
	VDDIO = 3.3V		VDDIO = 3.3V	VDDIO = 3.3V	
Conditions	VDDOUT Consumption AMP1	Total Consumption AMP2	Total Consumption AMP2	Total Consumption AMP2	
No activity on the I/Os of the device	–	0.3	3.8	7.5	mA

61. Marking

All devices are marked with the Atmel logo and the ordering code.

Additional marking is as follows:

where

- “YY”: Manufacturing year
- “WW”: Manufacturing week
- “V”: Revision
- “XXXXXXXXX”: Lot number