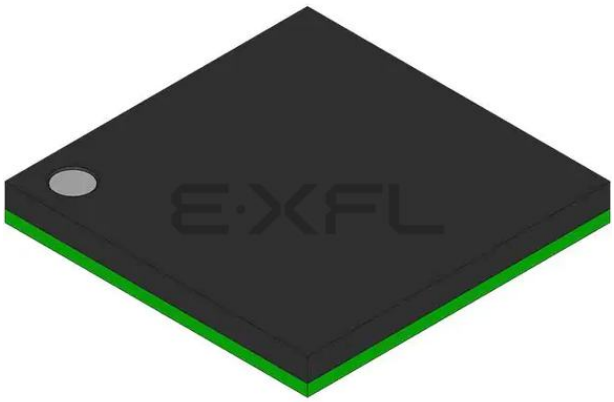


Welcome to E-XFL.COM

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	Obsolete
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	96MHz
Connectivity	EBI/EMI, I²C, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	57
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 4x10b, 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/atmel/atsam3u1ca-cu

- If the returned status bit from the second step indicates that the Store-Exclusive succeeded then the software has claimed the semaphore. However, if the Store-Exclusive failed, another process might have claimed the semaphore after the software performed the first step.

The Cortex-M3 includes an exclusive access monitor, that tags the fact that the processor has executed a Load-Exclusive instruction. If the processor is part of a multiprocessor system, the system also globally tags the memory locations addressed by exclusive accesses by each processor.

The processor removes its exclusive access tag if:

- It executes a CLREX instruction
- It executes a Store-Exclusive instruction, regardless of whether the write succeeds.
- An exception occurs. This means the processor can resolve semaphore conflicts between different threads.

In a multiprocessor implementation:

- executing a CLREX instruction removes only the local exclusive access tag for the processor
- executing a Store-Exclusive instruction, or an exception. removes the local exclusive access tags, and all global exclusive access tags for the processor.

For more information about the synchronization primitive instructions, see “LDREX and STREX” on page 102 and “CLREX” on page 104.

12.4.8 Programming hints for the synchronization primitives

ANSI C cannot directly generate the exclusive access instructions. Some C compilers provide intrinsic functions for generation of these instructions:

Table 12-8. C compiler intrinsic functions for exclusive access instructions

Instruction	Intrinsic function
LDREX, LDREXH, or LDREXB	unsigned int __ldrex(volatile void *ptr)
STREX, STREXH, or STREXB	int __strex(unsigned int val, volatile void *ptr)
CLREX	void __clrex(void)

The actual exclusive access instruction generated depends on the data type of the pointer passed to the intrinsic function. For example, the following C code generates the require LDREXB operation:

```
__ldrex((volatile char *) 0xFF);
```

12.5 Exception model

This section describes the exception model.

12.5.1 Exception states

Each exception is in one of the following states:

12.5.1.1 Inactive

The exception is not active and not pending.

12.5.1.2 Pending

The exception is waiting to be serviced by the processor.

An interrupt request from a peripheral or from software can change the state of the corresponding interrupt to pending.

Table 12-13. Cortex-M3 instructions (Continued)

Mnemonic	Operands	Brief description	Flags	Page
LDMDB, LDMEA	Rn{!}, reglist	Load Multiple registers, decrement before	-	page 99
LDMFD, LDMIA	Rn{!}, reglist	Load Multiple registers, increment after	-	page 99
LDR	Rt, [Rn, #offset]	Load Register with word	-	page 94
LDRB, LDRBT	Rt, [Rn, #offset]	Load Register with byte	-	page 94
LDRD	Rt, Rt2, [Rn, #offset]	Load Register with two bytes	-	page 94
LDREX	Rt, [Rn, #offset]	Load Register Exclusive	-	page 94
LDREXB	Rt, [Rn]	Load Register Exclusive with byte	-	page 94
LDREXH	Rt, [Rn]	Load Register Exclusive with halfword	-	page 94
LDRH, LDRHT	Rt, [Rn, #offset]	Load Register with halfword	-	page 94
LDRSB, LDRSBT	Rt, [Rn, #offset]	Load Register with signed byte	-	page 94
LDRSH, LDRSHT	Rt, [Rn, #offset]	Load Register with signed halfword	-	page 94
LDRT	Rt, [Rn, #offset]	Load Register with word	-	page 94
LSL, LSLS	Rd, Rm, <Rs|#n>	Logical Shift Left	N,Z,C	page 109
LSR, LSRS	Rd, Rm, <Rs|#n>	Logical Shift Right	N,Z,C	page 109
MLA	Rd, Rn, Rm, Ra	Multiply with Accumulate, 32-bit result	-	page 119
MLS	Rd, Rn, Rm, Ra	Multiply and Subtract, 32-bit result	-	page 119
MOV, MOVS	Rd, Op2	Move	N,Z,C	page 113
MOVT	Rd, #imm16	Move Top	-	page 115
MOVW, MOV	Rd, #imm16	Move 16-bit constant	N,Z,C	page 113
MRS	Rd, spec_reg	Move from special register to general register	-	page 141
MSR	spec_reg, Rm	Move from general register to special register	N,Z,C,V	page 142
MUL, MULS	{Rd,} Rn, Rm	Multiply, 32-bit result	N,Z	page 119
MVN, MVNS	Rd, Op2	Move NOT	N,Z,C	page 113
NOP	-	No Operation	-	page 143
ORN, ORNS	{Rd,} Rn, Op2	Logical OR NOT	N,Z,C	page 108
ORR, ORRS	{Rd,} Rn, Op2	Logical OR	N,Z,C	page 108
POP	reglist	Pop registers from stack	-	page 101
PUSH	reglist	Push registers onto stack	-	page 101
RBIT	Rd, Rn	Reverse Bits	-	page 116
REV	Rd, Rn	Reverse byte order in a word	-	page 116
REV16	Rd, Rn	Reverse byte order in each halfword	-	page 116
REVSH	Rd, Rn	Reverse byte order in bottom halfword and sign extend	-	page 116
ROR, RORS	Rd, Rm, <Rs|#n>	Rotate Right	N,Z,C	page 109
RRX, RRXS	Rd, Rm	Rotate Right with Extend	N,Z,C	page 109
RSB, RSBS	{Rd,} Rn, Op2	Reverse Subtract	N,Z,C,V	page 106
SBC, SBCS	{Rd,} Rn, Op2	Subtract with Carry	N,Z,C,V	page 106
SBFX	Rd, Rn, #lsb, #width	Signed Bit Field Extract	-	page 126

12.12.5 CMP and CMN

Compare and Compare Negative.

12.12.5.1 Syntax

```
CMP{cond} Rn, Operand2
CMN{cond} Rn, Operand2
```

where:

cond is an optional condition code, see “Conditional execution” on page 87.

Rn is the register holding the first operand.

Operand2 is a flexible second operand. See “Flexible second operand” on page 82 for details of the options.

12.12.5.2 Operation

These instructions compare the value in a register with *Operand2*. They update the condition flags on the result, but do not write the result to a register.

The CMP instruction subtracts the value of *Operand2* from the value in *Rn*. This is the same as a SUBS instruction, except that the result is discarded.

The CMN instruction adds the value of *Operand2* to the value in *Rn*. This is the same as an ADDS instruction, except that the result is discarded.

12.12.5.3 Restrictions

In these instructions:

- do not use PC
- *Operand2* must not be SP.

12.12.5.4 Condition flags

These instructions update the N, Z, C and V flags according to the result.

12.12.5.5 Examples

```
CMP     R2, R9
CMN     R0, #6400
CMPGT   SP, R7, LSL #2
```

12.19.3 Interrupt Clear-enable Registers

The ICER0 register disables interrupts, and shows which interrupts are enabled. See:

- the register summary in Table 12-27 on page 149 for the register attributes
- Table 12-28 on page 150 for which interrupts are controlled by each register

The bit assignments are:

31	30	29	28	27	26	25	24
CLRENA							

23	22	21	20	19	18	17	16
CLRENA							

15	14	13	12	11	10	9	8
CLRENA							

7	6	5	4	3	2	1	0
CLRENA							

- **CLRENA**

Interrupt clear-enable bits.

Write:

0 = no effect

1 = disable interrupt.

Read:

0 = interrupt disabled

1 = interrupt enabled.

12.19.5 Interrupt Clear-pending Registers

The ICPR0 register removes the pending state from interrupts, and show which interrupts are pending. See:

- the register summary in Table 12-27 on page 149 for the register attributes
- Table 12-28 on page 150 for which interrupts are controlled by each register.

The bit assignments are:

31	30	29	28	27	26	25	24
CLRPEND							

23	22	21	20	19	18	17	16
CLRPEND							

15	14	13	12	11	10	9	8
CLRPEND							

7	6	5	4	3	2	1	0
CLRPEND							

- **CLRPEND**

Interrupt clear-pending bits.

Write:

0 = no effect.

1 = removes pending state an interrupt.

Read:

0 = interrupt is not pending.

1 = interrupt is pending.

Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt.

Table 12-35. MPU registers summary

Address	Name	Type	Required privilege	Reset value	Description
0xE000ED90	TYPE	RO	Privileged	0x00000800	“MPU Type Register” on page 196
0xE000ED94	CTRL	RW	Privileged	0x00000000	“MPU Control Register” on page 197
0xE000ED98	RNR	RW	Privileged	0x00000000	“MPU Region Number Register” on page 199
0xE000ED9C	RBAR	RW	Privileged	0x00000000	“MPU Region Base Address Register” on page 200
0xE000EDA0	RASR	RW	Privileged	0x00000000	“MPU Region Attribute and Size Register” on page 201
0xE000EDA4	RBAR_A1	RW	Privileged	0x00000000	Alias of RBAR, see “MPU Region Base Address Register” on page 200
0xE000EDA8	RASR_A1	RW	Privileged	0x00000000	Alias of RASR, see “MPU Region Attribute and Size Register” on page 201
0xE000EDAC	RBAR_A2	RW	Privileged	0x00000000	Alias of RBAR, see “MPU Region Base Address Register” on page 200
0xE000EDB0	RASR_A2	RW	Privileged	0x00000000	Alias of RASR, see “MPU Region Attribute and Size Register” on page 201
0xE000EDB4	RBAR_A3	RW	Privileged	0x00000000	Alias of RBAR, see “MPU Region Base Address Register” on page 200
0xE000EDB8	RASR_A3	RW	Privileged	0x00000000	Alias of RASR, see “MPU Region Attribute and Size Register” on page 201

30.5 Pin Name List

Table 30-1. I/O Lines Description

Pin Name	Pin Description	Type
RF	Receiver Frame Synchro	Input/Output
RK	Receiver Clock	Input/Output
RD	Receiver Data	Input
TF	Transmitter Frame Synchro	Input/Output
TK	Transmitter Clock	Input/Output
TD	Transmitter Data	Output

30.6 Product Dependencies

30.6.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines.

Before using the SSC receiver, the PIO controller must be configured to dedicate the SSC receiver I/O lines to the SSC peripheral mode.

Before using the SSC transmitter, the PIO controller must be configured to dedicate the SSC transmitter I/O lines to the SSC peripheral mode.

Table 30-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
SSC	RD	PA27	A
SSC	RF	PA31	A
SSC	RK	PA29	A
SSC	TD	PA26	A
SSC	TF	PA30	A
SSC	TK	PA28	A

30.6.2 Power Management

The SSC is not continuously clocked. The SSC interface may be clocked through the Power Management Controller (PMC), therefore the programmer must first configure the PMC to enable the SSC clock.

30.6.3 Interrupt

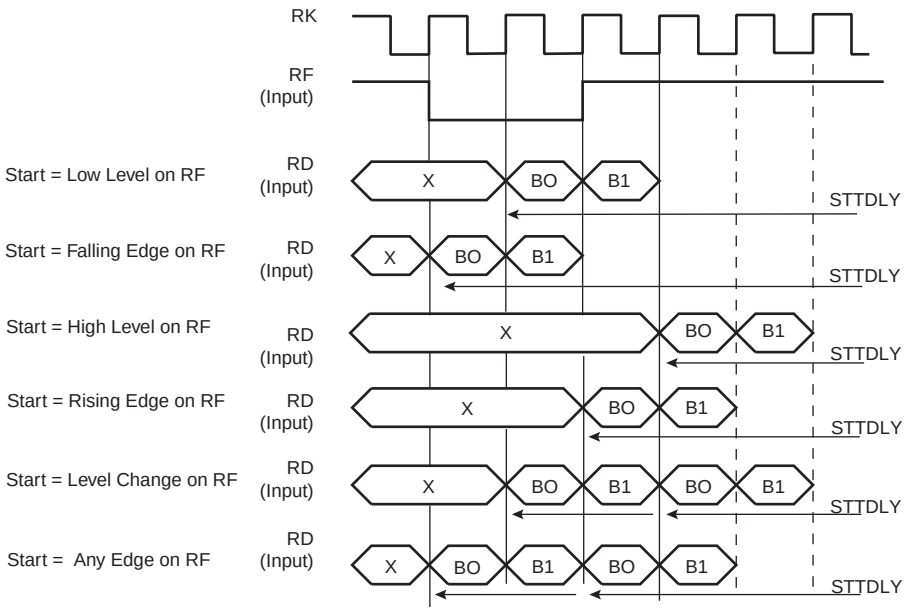
The SSC interface has an interrupt line connected to the Nested Vector Interrupt Controller (NVIC). Handling interrupts requires programming the NVIC before configuring the SSC.

All SSC interrupts can be enabled/disabled configuring the SSC Interrupt mask register. Each pending and unmasked SSC interrupt will assert the SSC interrupt line. The SSC interrupt service routine can get the interrupt origin by reading the SSC interrupt status register.

Table 30-3. Peripheral IDs

Instance	ID
SSC	21

Figure 30-11. Receive Pulse/Edge Start Modes



30.7.5 Frame Sync

The Transmitter and Receiver Frame Sync pins, TF and RF, can be programmed to generate different kinds of frame synchronization signals. The Frame Sync Output Selection (FSOS) field in the Receive Frame Mode Register (SSC_RFMR) and in the Transmit Frame Mode Register (SSC_TFMR) are used to select the required waveform.

- Programmable low or high levels during data transfer are supported.
- Programmable high levels before the start of data transfers or toggling are also supported.

If a pulse waveform is selected, the Frame Sync Length (FSLEN) field in SSC_RFMR and SSC_TFMR programs the length of the pulse, from 1 bit time up to 256 bit time.

The periodicity of the Receive and Transmit Frame Sync pulse output can be programmed through the Period Divider Selection (PERIOD) field in SSC_RCMR and SSC_TCMR.

30.7.5.1 Frame Sync Data

Frame Sync Data transmits or receives a specific tag during the Frame Sync signal.

During the Frame Sync signal, the Receiver can sample the RD line and store the data in the Receive Sync Holding Register and the transmitter can transfer Transmit Sync Holding Register in the Shifter Register. The data length to be sampled/shifted out during the Frame Sync signal is programmed by the FSLEN field in SSC_RFMR/SSC_TFMR and has a maximum value of 16.

Concerning the Receive Frame Sync Data operation, if the Frame Sync Length is equal to or lower than the delay between the start event and the actual data reception, the data sampling operation is performed in the Receive Sync Holding Register through the Receive Shift Register.

The Transmit Frame Sync Operation is performed by the transmitter only if the bit Frame Sync Data Enable (FSDEN) in SSC_TFMR is set. If the Frame Sync length is equal to or lower than the delay between the start event and the actual data transmission, the normal transmission has priority and the data contained in the Transmit Sync Holding Register is transferred in the Transmit Register, then shifted out.

32.11.3 TWI Slave Mode Register

Name: TWI_SMR

Address: 0x40084008 (0), 0x40088008 (1)

Access: Read-write

Reset: 0x00000000

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	SADR						
15	14	13	12	11	10	9	8
–	–	–	–	–	–		
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **SADR: Slave Address**

The slave device address is used in Slave mode in order to be accessed by master devices in read or write mode.

SADR must be programmed before enabling the Slave mode or after a general call. Writes at other times have no effect.

- **MODE9: 9-bit Character Length**

0: CHRL defines character length.

1: 9-bit character length.

- **CLKO: Clock Output Select**

0: The USART does not drive the SCK pin.

1: The USART drives the SCK pin if USCLKS does not select the external clock SCK.

- **OVER: Oversampling Mode**

0: 16x Oversampling.

1: 8x Oversampling.

- **INACK: Inhibit Non Acknowledge**

0: The NACK is generated.

1: The NACK is not generated.

Note: In SPI master mode, if INACK = 0 the character transmission starts as soon as a character is written into US_THR register (assuming TXRDY was set). When INACK is 1, an additional condition must be met. The character transmission starts when a character is written and only if RXRDY flag is cleared (Receiver Holding Register has been read).

- **DSNACK: Disable Successive NACK**

0: NACK is sent on the ISO line as soon as a parity error occurs in the received character (unless INACK is set).

1: Successive parity errors are counted up to the value specified in the MAX_ITERATION field. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line. The flag ITERATION is asserted.

- **INVDATA: INverted Data**

0: The data field transmitted on TXD line is the same as the one written in US_THR register or the content read in US_RHR is the same as RXD line. Normal mode of operation.

1: The data field transmitted on TXD line is inverted (voltage polarity only) compared to the value written on US_THR register or the content read in US_RHR is inverted compared to what is received on RXD line (or ISO7816 IO line). Inverted Mode of operation, useful for contactless card application. To be used with configuration bit MSBF.

- **VAR_SYNC: Variable Synchronization of Command/Data Sync Start Frame Delimiter**

0: User defined configuration of command or data sync field depending on MODSYNC value.

1: The sync field is updated when a character is written into US_THR register.

- **MAX_ITERATION**

Defines the maximum number of iterations in mode ISO7816, protocol T= 0.

- **FILTER: Infrared Receive Line Filter**

0: The USART does not filter the receive line.

1: The USART filters the receive line using a three-sample filter (1/16-bit clock) (2 over 3 majority).

- **MAN: Manchester Encoder/Decoder Enable**

0: Manchester Encoder/Decoder are disabled.

1: Manchester Encoder/Decoder are enabled.

34.8.16 USART Write Protect Mode Register

Name: US_WPMR

Address: 0x400900E4 (0), 0x400940E4 (1), 0x400980E4 (2), 0x4009C0E4 (3)

Access: Read-write

Reset: See Table 34-15

31	30	29	28	27	26	25	24
WPKEY							
23	22	21	20	19	18	17	16
WPKEY							
15	14	13	12	11	10	9	8
WPKEY							
7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	WPEN

- **WPEN: Write Protect Enable**

0 = Disables the Write Protect if WPKEY corresponds to 0x555341 (“USA” in ASCII).

1 = Enables the Write Protect if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Protects the registers:

- “USART Mode Register” on page 730
- “USART Baud Rate Generator Register” on page 745
- “USART Receiver Time-out Register” on page 746
- “USART Transmitter Timeguard Register” on page 747
- “USART FI DI RATIO Register” on page 748
- “USART IrDA FILTER Register” on page 750
- “USART Manchester Configuration Register” on page 751

- **WPKEY: Write Protect KEY**

Should be written at value 0x555341 (“USA” in ASCII). Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

36.8.7 WRITE_MULTIPLE_BLOCK

36.8.7.1 One Block per Descriptor

1. Wait until the current command execution has successfully terminated.
   a. Check that CMDRDY and NOTBUSY are asserted in HSMCI_SR.
2. Program the block length in the card. This value defines the value *block_length.*
3. Program the block length in the HSMCI configuration register with *block_length* value.
4. Program HSMCI_DMA register with the following fields:
   – OFFSET field with *dma_offset.*
   – CHKSIZE is user defined.
   – DMAEN is set to true to enable DMAC hardware handshaking in the HSMCI. This bit was previously set to false.
5. Issue a WRITE_MULTIPLE_BLOCK command.
6. Program the DMA Controller to use a list of descriptors. Each descriptor transfers one block of data. Block *n* of data is transferred with descriptor LLI(n).
   a. Read the channel Register to choose an available (disabled) channel.
   b. Clear any pending interrupts on the channel from the previous DMAC transfer by reading the DMAC_EBCISR register.
   c. Program a List of descriptors.
   d. The LLI(n).DMAC_SADDRx memory location for channel x must be set to the location of the source data. When the first data location is not word aligned, the two LSB bits define the temporary value called *dma_offset.* The two LSB bits of LLI(n).DMAC_SADDRx must be set to 0.
   e. The LLI(n).DMAC_DADDRx register for channel x must be set with the starting address of the HSMCI_FIFO address.
   f. Program LLI(n).DMAC_CTRLAx register of channel x with the following field's values:
      –DST_WIDTH is set to WORD.
      –SRC_WIDTH is set to WORD.
      –DCSIZE must be set according to the value of HSMCI_DMA, CHKSIZE field.
      –BTSIZE is programmed with *CEILING((block_length + dma_offset)/4).*
   g. Program LLI(n).DMAC_CTRLBx register for channel x with the following field's values:
      –DST_INCR is set to INCR.
      –SRC_INCR is set to INCR.
      –DST_DSCR is set to 0 (fetch operation is enabled for the destination).
      –SRC_DSCR is set to 1 (source address is contiguous).
      –FC field is programmed with memory to peripheral flow control mode.
      –Both DST_DSCR and SRC_DSCR are set to 1 (descriptor fetch is disabled).
      –DIF and SIF are set with their respective layer ID. If SIF is different from DIF, DMA Controller is able to prefetch data and write HSMCI simultaneously.
   h. Program LLI(n).DMAC_CFGx register for channel x with the following field's values:
      –FIFOCFG defines the watermark of the DMA channel FIFO.
      –DST_H2SEL is set to true to enable hardware handshaking on the destination.
      –SRC_REP is set to 0. (contiguous memory access at block boundary)
      –DST_PER is programmed with the hardware handshaking ID of the targeted HSMCI Host Controller.

9. Wait for XFRDONE in HSMCI_SR register.

36.9 SD/SDIO Card Operation

The High Speed MultiMedia Card Interface allows processing of SD Memory (Secure Digital Memory Card) and SDIO (SD Input Output) Card commands.

SD/SDIO cards are based on the Multi Media Card (MMC) format, but are physically slightly thicker and feature higher data transfer rates, a lock switch on the side to prevent accidental overwriting and security features. The physical form factor, pin assignment and data transfer protocol are forward-compatible with the High Speed MultiMedia Card with some additions. SD slots can actually be used for more than flash memory cards. Devices that support SDIO can use small devices designed for the SD form factor, such as GPS receivers, Wi-Fi or Bluetooth adapters, modems, barcode readers, IrDA adapters, FM radio tuners, RFID readers, digital cameras and more.

SD/SDIO is covered by numerous patents and trademarks, and licensing is only available through the Secure Digital Card Association.

The SD/SDIO Card communication is based on a 9-pin interface (Clock, Command, 4 x Data and 3 x Power lines). The communication protocol is defined as a part of this specification. The main difference between the SD/SDIO Card and the High Speed MultiMedia Card is the initialization process.

The SD/SDIO Card Register (HSMCI_SDCR) allows selection of the Card Slot and the data bus width.

The SD/SDIO Card bus allows dynamic configuration of the number of data lines. After power up, by default, the SD/SDIO Card uses only DAT0 for data transfer. After initialization, the host can change the bus width (number of active data lines).

36.9.1 SDIO Data Transfer Type

SDIO cards may transfer data in either a multi-byte (1 to 512 bytes) or an optional block format (1 to 511 blocks), while the SD memory cards are fixed in the block transfer mode. The TRTYP field in the HSMCI Command Register (HSMCI_CMDR) allows to choose between SDIO Byte or SDIO Block transfer.

The number of bytes/blocks to transfer is set through the BCNT field in the HSMCI Block Register (HSMCI_BLKR). In SDIO Block mode, the field BLKLEN must be set to the data block size while this field is not used in SDIO Byte mode.

An SDIO Card can have multiple I/O or combined I/O and memory (called Combo Card). Within a multi-function SDIO or a Combo card, there are multiple devices (I/O and memory) that share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SDIO and combo cards can implement the optional concept of suspend/resume (Refer to the SDIO Specification for more details). To send a suspend or a resume command, the host must set the SDIO Special Command field (IOSPCMD) in the HSMCI Command Register.

36.9.2 SDIO Interrupts

Each function within an SDIO or Combo card may implement interrupts (Refer to the SDIO Specification for more details). In order to allow the SDIO card to interrupt the host, an interrupt function is added to a pin on the DAT[1] line to signal the card's interrupt to the host. An SDIO interrupt on each slot can be enabled through the HSMCI Interrupt Enable Register. The SDIO interrupt is sampled regardless of the currently selected slot.

37.7.9 PWM Sync Channels Mode Register

Name: PWM_SCM

Address: 0x4008C020

Access: Read-write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
PTRCS			PTRM	–	–	UPDM	

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	SYNC3	SYNC2	SYNC1	SYNC0

This register can only be written if the bits WPSWS2 and WPHWS2 are cleared in “PWM Write Protect Status Register” on page 926.

- **SYNCx: Synchronous Channel x**

0 = Channel x is not a synchronous channel.

1 = Channel x is a synchronous channel.

- **UPDM: Synchronous Channels Update Mode**

Value	Name	Description
0	MODE0	Manual write of double buffer registers and manual update of synchronous channels(1)
1	MODE1	Manual write of double buffer registers and automatic update of synchronous channels(2)
2	MODE2	Automatic write of duty-cycle update registers by the PDC and automatic update of synchronous channels(2)
3	–	Reserved

Notes: 1. The update occurs at the beginning of the next PWM period, when the UPDULOCK bit in “PWM Sync Channels Update Control Register” is set.

2. The update occurs when the Update Period is elapsed.

- **PTRM: PDC Transfer Request Mode**

UPDM	PTRM	WRDY Flag and PDC Transfer Request
0	x	The WRDY flag in “PWM Interrupt Status Register 2” on page 911 and the PDC transfer request are never set to 1.
1	x	The WRDY flag in “PWM Interrupt Status Register 2” on page 911 is set to 1 as soon as the update period is elapsed, the PDC transfer request is never set to 1.
2	0	The WRDY flag in “PWM Interrupt Status Register 2” on page 911 and the PDC transfer request are set to 1 as soon as the update period is elapsed.
	1	The WRDY flag in “PWM Interrupt Status Register 2” on page 911 and the PDC transfer request are set to 1 as soon as the selected comparison matches.

- **PTRCS: PDC Transfer Request Comparison Selection**

Selection of the comparison used to set the flag WRDY and the corresponding PDC transfer request.

39.5.5 DMAC Software Last Transfer Flag Register

Name: DMAC_LAST

Address: 0x400B0010

Access: Read-write

Reset: 0x00000000

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
DLAST3	SLAST3	DLAST2	SLAST2	DLAST1	SLAST1	DLAST0	SLAST0

- **DLASTx**

Writing one to DLASTx prior to writing one to DSREQx or DCREQx indicates that this destination request is the last transfer of the buffer.

- **SLASTx**

Writing one to SLASTx prior to writing one to SSREQx or SCREQx indicates that this source request is the last transfer of the buffer.

Warning: No input buffer amplifier to isolate the source is included in the ADC12B. This must be taken into consideration to program a precise value in the SHTIM field. See the section, ADC12B Characteristics in the product datasheet.

41.6.7 ADC Last Converted Data Register

Name: ADC_LCDR

Address: 0x400AC020

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	LDATA	
7	6	5	4	3	2	1	0
LDATA							

- **LDATA: Last Data Converted**

The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.

42.2 DC Characteristics

The following characteristics are applicable to the operating temperature range: T_A = -40°C to 85°C, unless otherwise specified.

Table 42-2. DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDCORE}	DC Supply Core		1.62	1.8	1.95	V
V_{DDIO}	DC Supply I/Os		1.62	3.3	3.6	V
V_{DDBU}	Backup I/O Lines Power Supply		1.62		3.6	V
V_{DDUTMI}	USB UTMI+ Interface Power Supply		3.0		3.6	V
V_{DDPLL}	PLL A, UPLL and Main Oscillator Supply		1.62		1.95	V
V_{DDANA}	ADC Analog Power Supply		(1)		(1)	V
V_{IL}	Input Low-level Voltage	PIOA/B/C[0–31]	-0.3		$0.3 \times V_{DDIO}$	V
V_{IH}	Input High-level Voltage	PIOA/B/C[0–31]	$0.7 \times V_{DDIO}$		V_{DDIO} + 0.3V	V
V_{OH}	Output High-level Voltage	PIOA/B/C[0–31] I_{OH} ~ 0 I_{OH} > 0 (See I_{OH} characteristics in this table)	 V_{DDIO} - 0.2V V_{DDIO} - 0.4V			V
V_{OL}	Output Low-level Voltage	PIOA/B/C[0–31] I_{OH} ~ 0 I_{OH} > 0 (See I_{OL} characteristics in this table)			 0.2 0.4	V
V_{Hys}	Hysteresis Voltage	PIOA/B/C[0–31] except PIOA[14], PB[9–16], PB[25–PB31] and PC[20–27]	150		500	mV
		ERASE, TST, FWUP, JTAGSEL	230		700	mV
I_{OH}	Source Current	1.62V < VDDIO < 1.95V; V_{OH} = V_{DDIO} - 0.4V - PA3 (SPCK), PA15 (MCCK) pins - Other pins(2)			 -8 -3	mA
		3.0V < VDDIO < 3.6V; V_{OH} = V_{DDIO} - 0.4V - PA3 (SPCK), PA15 (MCCK) pins - Other pins(2)			 -15 -3	
		1.62V < VDDIO < 3.6V; V_{OH} = V_{DDIO} - 0.4V - NRST, TDO			 -2	
		Relaxed Mode: 3.0V < VDDIO < 3.6V; V_{OH} = 2.2V - PA3 (SPCK), PA15 (MCCK) pins - Other pins(2)			 -24 -9	

42.9 AC Characteristics

42.9.1 Master Clock Characteristics

Table 42-45. Master Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPMCK})$	Master Clock Frequency	VDDCORE @ 1.62V VDDCORE @ 1.8V		84 96	MHz

42.9.2 I/O Characteristics

Criteria used to define the maximum frequency of the I/Os:

- Output duty cycle (40%–60%)
- Minimum output swing: 100 mV to VDDIO - 100 mV
- Minimum output swing: 100 mV to VDDIO - 100 mV
- Addition of rising and falling time inferior to 75% of the period

Table 42-46. I/O Characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
FreqMax1	Pin Group 1 (1) Maximum output frequency	30 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V		45 65	MHz
		45 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V		34 45	
$PulseminH_1$	Pin Group 1 (1) High Level Pulse Width	30 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	11 7.7		ns
		45 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	14.7 11		
$PulseminL_1$	Pin Group 1 (1) Low Level Pulse Width	30 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	11 7.7		ns
		45 pF	V_{DDIO} = 1.62V V_{DDIO} = 3.0V	14.7 11		
FreqMax2	Pin Group 2 (2) Maximum output frequency	25 pF	1.62V < V_{DDIO} < 3.6V		35	MHz
$PulseminH_2$	Pin Group 2 (2) High Level Pulse Width	25 pF	1.62V < V_{DDIO} < 3.6V	14.5		ns
$PulseminL_2$	Pin Group 2 (2) Low Level Pulse Width	25 pF	1.62V < V_{DDIO} < 3.6V	14.5		ns

Notes: 1. Pin Group 1 = PA3, PA15

2. Pin Group 2 = PA[0–2], PA[4–14], PA[16–31], PB[0–31], PC[0–31]

Table 42-47. NRSTB Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{NRSTB(filtered)}$	Filtered Pulse Width				1	µs
$t_{NRSTB(unfiltered)}$	Unfiltered Pulse Width		100			µs

Doc Rev 6430B	Comments (Continued)	Change Request Ref.(1)
	Section 6.6 ”NRSTB Pin”, VDDIO changed to VDDBU	6646
	Section 6. ”Input/Output Lines” replaces Section 5.8 “Programmable I/O Lines”.	6481/rfo
	Section 6.1 ”General Purpose I/O Lines (GPIO)” and Section 6.2 ”System I/O Lines” replace Section 6. “I/O Line Considerations”.	
	Figure 6-1 ”On-Die Termination schematic”, added.	
	Section 6.8 “PIO Controllers”, removed.	
	Section 8. ”Product Mapping”, title changed from “Memories”.	
	Section 9. ”Memories”; now comprises Section 9.1 ”Embedded Memories” and Section 9.2 ”External Memories”	
	Section 9.1.3.5 ”Security Bit Feature”, updated	
	Table 7-3, “SAM3U Master to Slave Access”, Slave 9, High Speed Peripheral Bridge line added.	6663
	Section 7.2 ”APB/AHB Bridges”, reference to ADC updated “10-bit ADC, 12-bit ADC (ADC12B)”.	6397
	Table 11-3, “Multiplexing on PIO Controller B (PIOB)” ADC12B2, ADC12B3 properly listed.	
	Section 12.10.1 ”12-bit High Speed ADC”, Section 12.10.2 ”10-bit Low Power ADC”, titles changed.	
	“Quadrature Decoder Logic” on page 51 properly stated in list of TC functions.	
	Section 12.10.1 ”12-bit High Speed ADC”, 2nd item on list updated.	rfo
	Section 12.10.2 ”10-bit Low Power ADC”, Ksample values updated on 2nd item of list.	
	ADC12B:	
	Section 40.6.6 ”ADC12B Analog Control Register”,IBCTL reasigned to fields 8 and 9	6649
	CORTEX-M3:	
	Table 12-31, “Priority grouping”updated.	6394:
	Section 13.19.7.1 ”IP27”, title changed to IP27, value in bitfields 0 to 7 changed to IP28.	
	Section 12.20.6 ”Application Interrupt and Reset Control Register”read/write values to VECTKEY changed	6436
	Section 12.5.2.2 ”Non Maskable Interrupt (NMI)” added to datasheet.	
	Section 12.6.2 ”Fault escalation and hard faults”, last sentence updated with NMI function.	
	Section 12.3.5 ”Data types” Condition tags sorted out.	6483
	Table 12-11, “Faults”updated footnote 1	
	Table 12-29, “CMSIS functions for NVIC control”Description in 4th row updated.	
	on page 44, “...copyright ARM Ltd., 2008 - 2009.” precise years given.	
	Table 12-4, “Memory access behavior”last row assinged to Reserved in Memory Map.	rfo
	Big Endien not used in this product.	
	Debug and Test:	
	Section 13.1 ”Overview”, SWJ-DP...also embeds a serial trace.	rfo
	Table 13-1, “Debug and Test Signal List”, reorganized.	
	FFPI:	
	Section 21.2.5.4 ”Flash Lock Commands”, last sentence removed from 2nd paragraph (ref to EA command).	6677
	HSMCI:	
	Section 37.12 ”Write Protection Registers”, Section 37.13.18 ”HSMCI Write Protect Mode Register” and	6432
	Section 37.13.19 ”HSMCI Write Protect Status Register”,added.	
	MATRIX:	6431
	Table 23-1, “Register Mapping”, added offsets for Write Protection Registers	