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.

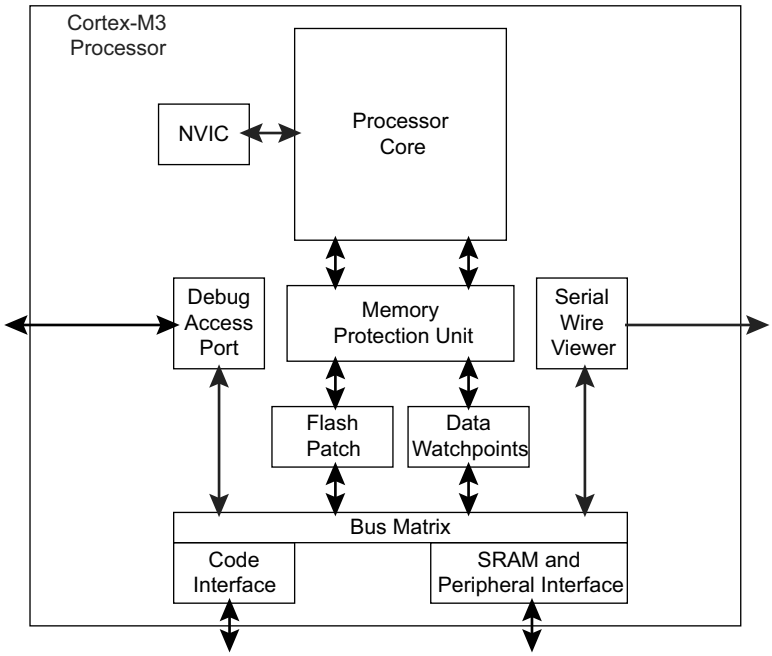
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	84MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	63
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam3x4ca-au

1. Features

- Core
  - ARM Cortex-M3 revision 2.0 running at up to 84 MHz
  - Memory Protection Unit (MPU)
  - Thumb®-2 instruction set
  - 24-bit SysTick Counter
  - Nested Vector Interrupt Controller
- Memories
  - 256 to 512 Kbytes embedded Flash, 128-bit wide access, memory accelerator, dual bank
  - 32 to 100 Kbytes embedded SRAM with dual banks
  - 16 Kbytes ROM with embedded bootloader routines (UART, USB) and IAP routines
  - Static Memory Controller (SMC): SRAM, NOR, NAND support. NFC with 4 Kbyte RAM buffer and ECC
- System
  - Embedded voltage regulator for single supply operation
  - Power-on-Reset (POR), Brown-out Detector (BOD) and Watchdog for safe reset
  - Quartz or ceramic resonator oscillators: 3 to 20 MHz main and optional low power 32.768 kHz for RTC or device clock
  - High precision 8/12 MHz factory trimmed internal RC oscillator with 4 MHz default frequency for fast device startup
  - Slow Clock Internal RC oscillator as permanent clock for device clock in low-power mode
  - One PLL for device clock and one dedicated PLL for USB 2.0 High Speed Mini Host/Device
  - Temperature Sensor
  - Up to 17 peripheral DMA (PDC) channels and 6-channel central DMA plus dedicated DMA for High-Speed USB Mini Host/Device and Ethernet MAC
- Low-power Modes
  - Sleep, Wait and Backup modes, down to 2.5 µA in Backup mode with RTC, RTT, and GPBR
- Peripherals
  - USB 2.0 Device/Mini Host: 480 Mbps, 4 Kbyte FIFO, up to 10 bidirectional Endpoints, dedicated DMA
  - Up to 4 USARTs (ISO7816, IrDA®, Flow Control, SPI, Manchester and LIN support) and one UART
  - 2 TWI (I2C compatible), up to 6 SPIs, 1 SSC (I2S), 1 HSMCI (SDIO/SD/MMC) with up to 2 slots
  - 9-channel 32-bit Timer Counter (TC) for capture, compare and PWM mode, Quadrature Decoder Logic and 2-bit Gray Up/Down Counter for Stepper Motor
  - Up to 8-channel 16-bit PWM (PWMC) with Complementary Output, Fault Input, 12-bit Dead Time Generator Counter for Motor Control
  - 32-bit low-power Real-time Timer (RTT) and low-power Real-time Clock (RTC) with calendar and alarm features
  - 256-bit General Purpose Backup Registers (GPBR)
  - 16-channel 12-bit 1 msps ADC with differential input mode and programmable gain stage
  - 2-channel 12-bit 1 msps DAC
  - Ethernet MAC 10/100 (EMAC) with dedicated DMA
  - 2 CAN Controllers with 8 Mailboxes
  - True Random Number Generator (TRNG)
  - Register Write Protection
- I/O
  - Up to 103 I/O lines with external interrupt capability (edge or level sensitivity), debouncing, glitch filtering and on-die Series Resistor Termination
  - Up to six 32-bit Parallel Input/Outputs (PIO)

Figure 10-1. Typical Cortex-M3 implementation



The Cortex-M3 processor is built on a high-performance processor core, with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including single-cycle 32x32 multiplication and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M3 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M3 processor implements a version of the Thumb® instruction set, ensuring high code density and reduced program memory requirements. The Cortex-M3 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M3 processor closely integrates a configurable *nested interrupt controlle*r (NVIC), to deliver industry-leading interrupt performance. The NVIC provides up to 16 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of *interrupt service routines* (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require any assembler stubs, removing any code overhead from the ISRs. Tail-chaining optimization also significantly reduces the overhead when switching from one ISR to another.

To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down.

10.3.1 System level interface

The Cortex-M3 processor provides multiple interfaces using AMBA® technology to provide high speed, low latency memory accesses. It supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks and thread-safe Boolean data handling.

The Cortex-M3 processor has a *memory protection unit* (MPU) that provides fine grain memory control, enabling applications to implement security privilege levels, separating code, data and stack on a task-by-task basis. Such requirements are becoming critical in many embedded applications.

10.17.3 IT

If-Then condition instruction.

10.17.3.1 Syntax

```
IT{x{y{z}}} cond
```

where:

x specifies the condition switch for the second instruction in the IT block.

y specifies the condition switch for the third instruction in the IT block.

z specifies the condition switch for the fourth instruction in the IT block.

cond specifies the condition for the first instruction in the IT block.

The condition switch for the second, third and fourth instruction in the IT block can be either:

T Then. Applies the condition *cond* to the instruction.

E Else. Applies the inverse condition of *cond* to the instruction.

It is possible to use AL (the *always* condition) for *cond* in an IT instruction. If this is done, all of the instructions in the IT block must be unconditional, and each of *x*, *y*, and *z* must be T or omitted but not E.

10.17.3.2 Operation

The IT instruction makes up to four following instructions conditional. The conditions can be all the same, or some of them can be the logical inverse of the others. The conditional instructions following the IT instruction form the *IT block*.

The instructions in the IT block, including any branches, must specify the condition in the {*cond*} part of their syntax.

Your assembler might be able to generate the required IT instructions for conditional instructions automatically, so that you do not need to write them yourself. See your assembler documentation for details.

A BKPT instruction in an IT block is always executed, even if its condition fails.

Exceptions can be taken between an IT instruction and the corresponding IT block, or within an IT block. Such an exception results in entry to the appropriate exception handler, with suitable return information in LR and stacked PSR.

Instructions designed for use for exception returns can be used as normal to return from the exception, and execution of the IT block resumes correctly. This is the only way that a PC-modifying instruction is permitted to branch to an instruction in an IT block.

10.17.3.3 Restrictions

The following instructions are not permitted in an IT block:

- IT
- CBZ and CBNZ
- CPSID and CPSIE.

Other restrictions when using an IT block are:

- a branch or any instruction that modifies the PC must either be outside an IT block or must be the last instruction inside the IT block. These are:
  - ADD PC, PC, Rm
  - MOV PC, Rm
  - B, BL, BX, BLX
  - any LDM, LDR, or POP instruction that writes to the PC
  - TBB and TBH

14.6.13 RTC Write Protect Mode Register

Name: RTC_WPMR

Address: 0x400E1B44

Access: Read-write

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 0x525443 (“RTC” in ASCII).

1: Enables the Write Protect if WPKEY corresponds to 0x525443 (“RTC” in ASCII).

Protects the registers:

“RTC Mode Register” on page 248

“RTC Time Alarm Register” on page 251

“RTC Calendar Alarm Register” on page 252

One error can be detected in the EEFC_FSR register after a programming sequence:

- a Command Error: a bad keyword has been written in the EEFC_FCR register.

It is possible to clear lock bits previously set. Then the locked region can be erased or programmed. The unlock sequence is:

- The Clear Lock command (CLB) and a page number to be unprotected are written in the Flash Command Register.
- When the unlock completes, the FRDY bit in the Flash Programming Status Register (EEFC_FSR) rises. If an interrupt has been enabled by setting the FRDY bit in EEFC_FMR, the interrupt line of the NVIC is activated.
- If the lock bit number is greater than the total number of lock bits, then the command has no effect.

One error can be detected in the EEFC_FSR register after a programming sequence:

- a Command Error: a bad keyword has been written in the EEFC_FCR register.

The status of lock bits can be returned by the Enhanced Embedded Flash Controller (EEFC). The Get Lock Bit status sequence is:

- The Get Lock Bit command (GLB) is written in the Flash Command Register, FARG field is meaningless.
- Lock bits can be read by the software application in the EEFC_FRR register. The first word read corresponds to the 32 first lock bits, next reads providing the next 32 lock bits as long as it is meaningful. Extra reads to the EEFC_FRR register return 0.

For example, if the third bit of the first word read in the EEFC_FRR is set, then the third lock region is locked.

One error can be detected in the EEFC_FSR register after a programming sequence:

- a Command Error: a bad keyword has been written in the EEFC_FCR register.

Note: Access to the Flash in read is permitted when a set, clear or get lock bit command is performed.

18.4.3.5 GPNVM Bit

GPNVM bits do not interfere with the embedded Flash memory plane. Refer to the product definition section for information on the GPNVM Bit Action.

The set GPNVM bit sequence is:

- Start the Set GPNVM Bit command (SGPB) by writing the Flash Command Register with the SGPB command and the number of the GPNVM bit to be set.
- When the GPVNM bit is set, the bit FRDY in the Flash Programming Status Register (EEFC_FSR) rises. If an interrupt was enabled by setting the FRDY bit in EEFC_FMR, the interrupt line of the NVIC is activated.
- If the GPNVM bit number is greater than the total number of GPNVM bits, then the command has no effect. The result of the SGPB command can be checked by running a GGPB (Get GPNVM Bit) command.

One error can be detected in the EEFC_FSR register after a programming sequence:

- A Command Error: a bad keyword has been written in the EEFC_FCR register.

It is possible to clear GPNVM bits previously set. The clear GPNVM bit sequence is:

- Start the Clear GPNVM Bit command (CGPB) by writing the Flash Command Register with CGPB and the number of the GPNVM bit to be cleared.
- When the clear completes, the FRDY bit in the Flash Programming Status Register (EEFC_FSR) rises. If an interrupt has been enabled by setting the FRDY bit in EEFC_FMR, the interrupt line of the NVIC is activated.
- If the GPNVM bit number is greater than the total number of GPNVM bits, then the command has no effect.

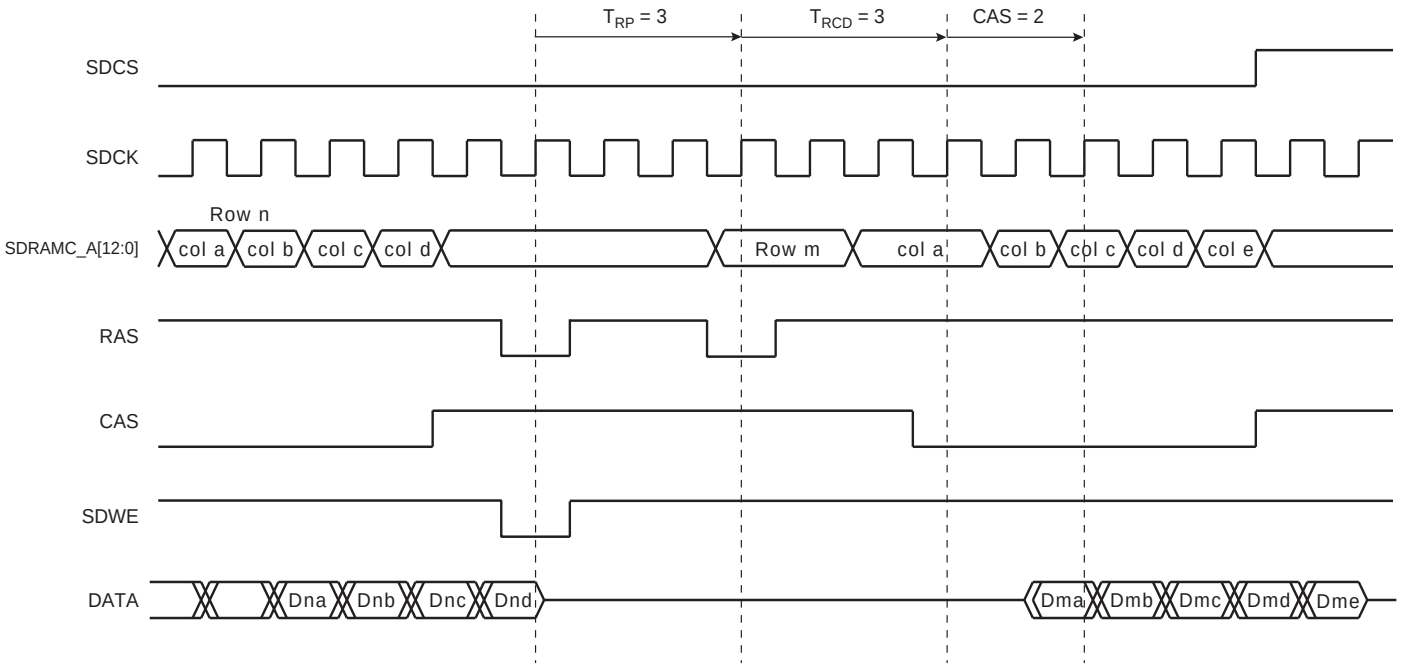
One error can be detected in the EEFC_FSR register after a programming sequence:

- A Command Error: a bad keyword has been written in the EEFC_FCR register.

24.6.3 Border Management

When the memory row boundary has been reached, an automatic page break is inserted. In this case, the SDRAM controller generates a precharge command, activates the new row and initiates a read or write command. To comply with SDRAM timing parameters, an additional clock cycle is inserted between the precharge/active (t_{RP}) command and the active/read (t_{RCD}) command. This is described in Figure 24-4 below.

Figure 24-4. Read Burst with Boundary Row Access



25.16.2.1 Building NFC Address Command Example.

The base address is made of address 0x60000000 + NFCCMD bit set = 0x68000000.

Page read operation example:

```
// Build the Address Command (NFCADDR_CMD)
AddressCommand = (0x60000000  |
                  NFCCMD=1 | // NFC Command Enable
                  NFCWR=0   |// NFC Read Data from NAND Flash
                  NFCEN=1    |    // NFC Enable.
                  CSID=1        |        // Chip Select ID = 1
                 ACYCLE= 5 |     // Number of address cycle.
                 VCMD2=1    |               // CMD2 is sent after Address Cycles
                 CMD2=0x30  |                // CMD2 = 30h
                 CMD1=0x0)    // CMD1 = Read Command = 00h

  // Set the Address for Cycle 0
  SMC_ADDR = Col. Add1

  // Write command with the Address Command built above
*AddressCommand = (Col. Add2   |// ADDR_CYCLE1
           Row Add1  | // ADDR_CYCLE2
           Row Add2  |// ADDR_CYCLE3
           Row Add3  )// ADDR_CYCLE4
```

25.18.16 SMC ECC Parity Registers for 1 ECC per 256 Bytes for a Page of 512/2048/4096 Bytes, 8-bit Word

Name: SMC_ECC_PRx [x=0..15] (W8BIT)

Address: 0x400E0038 [2] .. 0x400E006C [15]

Access: Read-only

Reset: 0x00000000

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

Once the entire main area of a page is written with data, the register content must be stored at any free location of the spare area.

- **BITADDR: Corrupted Bit Address in the page between (i x 256) and ((i + 1) x 512) - 1) Bytes**
- **During a page read, this value contains the corrupted bit offset where an error occurred, if a single error was detected. If multiple errors were detected, this value is meaningless.**
- **WORDADDR: Corrupted Word Address in the page between (i x 256) and ((i + 1) x 512) - 1) Bytes**

During a page read, this value contains the word address (8-bit word) where an error occurred, if a single error was detected. If multiple errors were detected, this value is meaningless.

- **NPARITY: Parity N**

30.9.9 SSC Receive Synchronization Holding Register

Name: SSC_RSHR

Address: 0x40004030

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
RSDAT							

7	6	5	4	3	2	1	0
RSDAT							

- **RSDAT: Receive Synchronization Data**

30.9.17 SSC Write Protect Mode Register

Name: SSC_WPMR

Address: 0x400040E4

Access: Read-write

Reset: See Table 30-6

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 0x535343 (“SSC” in ASCII).

1 = Enables the Write Protect if WPKEY corresponds to 0x535343 (“SSC” in ASCII).

Protects the registers:

- “SSC Clock Mode Register” on page 593
- “SSC Receive Clock Mode Register” on page 594
- “SSC Receive Frame Mode Register” on page 596
- “SSC Transmit Clock Mode Register” on page 598
- “SSC Transmit Frame Mode Register” on page 600
- “SSC Receive Compare 0 Register” on page 606
- “SSC Receive Compare 1 Register” on page 607

- **WPKEY: Write Protect KEY**

Should be written at value 0x535343 (“SSC” in ASCII). Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

33.11.3 TWI Slave Mode Register

Name: TWI_SMR

Address: 0x4008C008 (0), 0x40090008 (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.

35.2 Embedded Characteristics

- Programmable Baud Rate Generator
- 5- to 9-bit Full-duplex Synchronous or Asynchronous Serial Communications
  - 1, 1.5 or 2 Stop Bits in Asynchronous Mode or 1 or 2 Stop Bits in Synchronous Mode
  - Parity Generation and Error Detection
  - Framing Error Detection, Overrun Error Detection
  - MSB- or LSB-first
  - Optional Break Generation and Detection
  - By 8 or by 16 Over-sampling Receiver Frequency
  - Optional Hardware Handshaking RTS-CTS
  - Receiver Time-out and Transmitter Timeguard
  - Optional Multidrop Mode with Address Generation and Detection
- RS485 with Driver Control Signal
- ISO7816, T = 0 or T = 1 Protocols for Interfacing with Smart Cards
  - NACK Handling, Error Counter with Repetition and Iteration Limit
- IrDA Modulation and Demodulation
  - Communication at up to 115.2 Kbps
- SPI Mode
  - Master or Slave
  - Serial Clock Programmable Phase and Polarity
  - SPI Serial Clock (SCK) Frequency up to Internal Clock Frequency MCK/6
- LIN Mode (USART0 only)
  - Compliant with LIN 1.3 and LIN 2.0 specifications
  - Master or Slave
  - Processing of frames with up to 256 data bytes
  - Response Data length can be configurable or defined automatically by the Identifier
  - Self synchronization in Slave node configuration
  - Automatic processing and verification of the “Synch Break” and the “Synch Field”
  - The “Synch Break” is detected even if it is partially superimposed with a data byte
  - Automatic Identifier parity calculation/sending and verification
  - Parity sending and verification can be disabled
  - Automatic Checksum calculation/sending and verification
  - Checksum sending and verification can be disabled
  - Support both “Classic” and “Enhanced” checksum types
  - Full LIN error checking and reporting
  - Frame Slot Mode: the Master allocates slots to the scheduled frames automatically.
  - Generation of the Wakeup signal
- Test Modes
  - Remote Loopback, Local Loopback, Automatic Echo
- Supports Connection of Two Peripheral DMA Controller Channels (PDC)
  - Offers Buffer Transfer without Processor Intervention

- **ACPC: RC Compare Effect on TIOA**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **AEEVT: External Event Effect on TIOA**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **ASWTRG: Software Trigger Effect on TIOA**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BCPB: RB Compare Effect on TIOB**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BCPC: RC Compare Effect on TIOB**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

- **BEEVT: External Event Effect on TIOB**

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

38.5 Product Dependencies

38.5.1 I/O Lines

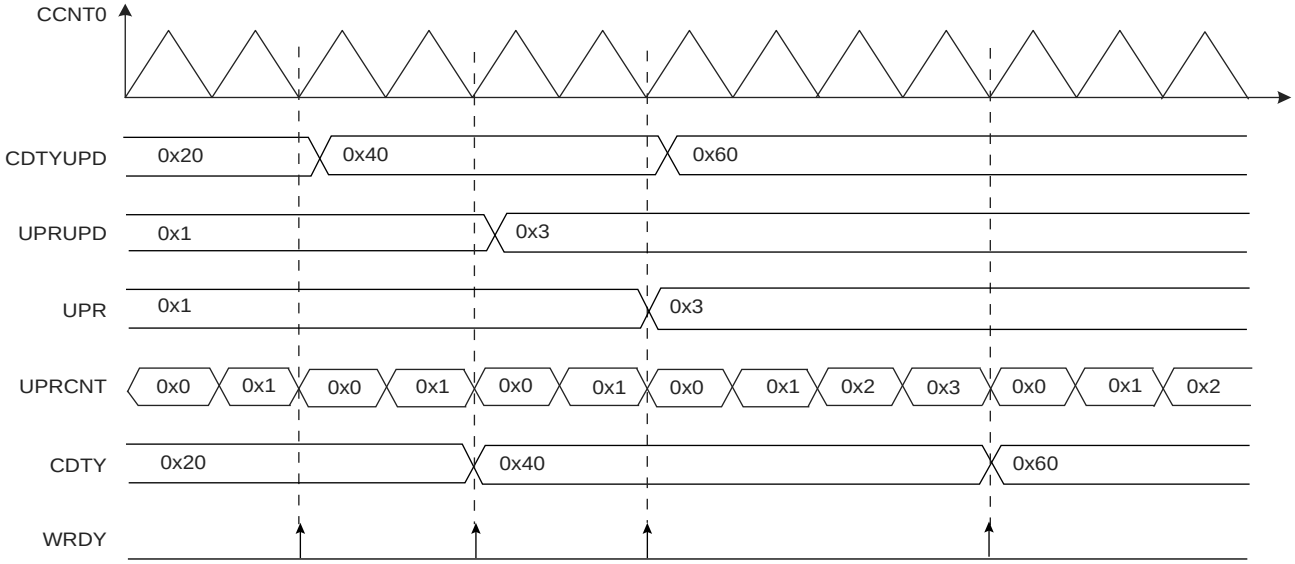
The pins used for interfacing the PWM are multiplexed with PIO lines. The programmer must first program the PIO controller to assign the desired PWM pins to their peripheral function. If I/O lines of the PWM are not used by the application, they can be used for other purposes by the PIO controller.

All of the PWM outputs may or may not be enabled. If an application requires only four channels, then only four PIO lines will be assigned to PWM outputs.

Table 38-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
PWM	PWMFI0	PA5	B
PWM	PWMFI1	PA3	B
PWM	PWMFI2	PD6	B
PWM	PWMH0	PA8	B
PWM	PWMH0	PB12	B
PWM	PWMH0	PC3	B
PWM	PWMH0	PE15	A
PWM	PWMH1	PA19	B
PWM	PWMH1	PB13	B
PWM	PWMH1	PC5	B
PWM	PWMH1	PE16	A
PWM	PWMH2	PA13	B
PWM	PWMH2	PB14	B
PWM	PWMH2	PC7	B
PWM	PWMH3	PA9	B
PWM	PWMH3	PB15	B
PWM	PWMH3	PC9	B
PWM	PWMH3	PF3	A
PWM	PWMH4	PC20	B
PWM	PWMH4	PE20	A
PWM	PWMH5	PC19	B
PWM	PWMH5	PE22	A
PWM	PWMH6	PC18	B
PWM	PWMH6	PE24	A
PWM	PWMH7	PE26	A
PWM	PWML0	PA21	B
PWM	PWML0	PB16	B
PWM	PWML0	PC2	B
PWM	PWML0	PE18	A
PWM	PWML1	PA12	B

Figure 38-11. Method 2 (UPDM=1)



38.6.5 PWM Controller Operations

38.6.5.1 Initialization

Before enabling the channels, they must have been configured by the software application:

- Unlock User Interface by writing the WPCMD field in the PWM_WPCR Register.
- Configuration of the clock generator (DIVA, PREA, DIVB, PREB in the PWM_CLK register if required).
- Selection of the clock for each channel (CPRE field in the PWM_CMRx register)
- Configuration of the waveform alignment for each channel (CALG field in the PWM_CMRx register)
- Selection of the counter event selection (if CALG = 1) for each channel (CES field in the PWM_CMRx register)
- Configuration of the output waveform polarity for each channel (CPOL in the PWM_CMRx register)
- Configuration of the period for each channel (CPRD in the PWM_CPRDx register). Writing in PWM_CPRDx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CPRDUPDx register to update PWM_CPRDx as explained below.
- Configuration of the duty-cycle for each channel (CDTY in the PWM_CDTYx register). Writing in PWM_CDTYx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CDTYUPDx register to update PWM_CDTYx as explained below.
- Configuration of the dead-time generator for each channel (DTH and DTL in PWM_DTx) if enabled (DTE bit in the PWM_CMRx register). Writing in the PWM_DTx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_DTUPDx register to update PWM_DTx
- Selection of the synchronous channels (SYNCx in the PWM_SCM register)
- Selection of the moment when the WRDY flag and the corresponding PDC transfer request are set (PTRM and PTRCS in the PWM_SCM register)
- Configuration of the update mode (UPDM in the PWM_SCM register)
- Configuration of the update period (UPR in the PWM_SCUP register) if needed.
- Configuration of the comparisons (PWM_CMPVx and PWM_CMPMx).
- Configuration of the event lines (PWM_ELMRx).
- Configuration of the fault inputs polarity (FPOL in PWM_FMR)
- Configuration of the fault protection (FMOD and FFIL in PWM_FMR, PWM_FPV and PWM_FPE/2)
- Enable of the Interrupts (writing CHIDx and FCHIDx in PWM_IER1 register, and writing WRDYE, ENDTXE, TXBUFE, UNRE, CMPMx and CMPUx in PWM_IER2 register)
- Enable of the PWM channels (writing CHIDx in the PWM_ENA register)

39.6.2.10 Device Endpoint x Configuration Register

Name: UOTGHS_DEVEPTCFGx [x=0..9]

Address: 0x400AC100

Access: Read-write

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

- **NBTRANS: Number of transaction per microframe for isochronous endpoint**

This field shall be written to the number of transaction per microframe to perform high-bandwidth isochronous transfer

This field can be written only for endpoint that have this capability (see UOTGHS_FEATURES.ENHBISOx bit). This field is 0 otherwise.

This field is irrelevant for non-isochronous endpoint.

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 micro-frame.
2	2_TRANS	2 transactions per micro-frame. This endpoint should be configured as double-bank.
3	3_TRANS	3 transactions per micro-frame. This endpoint should be configured as triple-bank.

- **EPTYPE: Endpoint Type**

This field shall be written to select the endpoint type:

Value	Name	Description
0	CTRL	Control
1	ISO	Isochronous
2	BLK	Bulk
3	INTRPT	Interrupt

This field is cleared upon receiving a USB reset.

- **AUTOSW: Automatic Switch**

This bit is cleared upon receiving a USB reset.

0: The automatic bank switching is disabled.

1: The automatic bank switching is enabled.

- **EPDIR: Endpoint Direction**

This bit is cleared upon receiving a USB reset.

0 (OUT): The endpoint direction is OUT.

1 (IN): The endpoint direction is IN (nor for control endpoints).

39.6.3.12 Host Pipe Register

Name: UOTGHS_HSTPIP

Address: 0x400AC41C

Access: Read-write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	PRST8
23	22	21	20	19	18	17	16
PRST7	PRST6	PRST5	PRST4	PRST3	PRST2	PRST1	PRST0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	PEN8
7	6	5	4	3	2	1	0
PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0

- **PRSTx: Pipe x Reset**

Writing a one to this bit will reset the Pipe x FIFO.

This resets the endpoint x registers (UOTGHS_HSTPIPCFGx, UOTGHS_HSTPIPISRx, UOTGHS_HSTPIPIMRx) but not the endpoint configuration (ALLOC, PBK, PSIZE, PTOKEN, PTYPE, PEPNUM, INTFRQ).

All the endpoint mechanism (FIFO counter, reception, transmission, etc.) is reset apart from the Data Toggle management.

The endpoint configuration remains active and the endpoint is still enabled.

Writing a zero to this bit will complete the reset operation and allow to start using the FIFO.

- **PENx: Pipe x Enable**

Writing a one to this bit will enable the Pipe x.

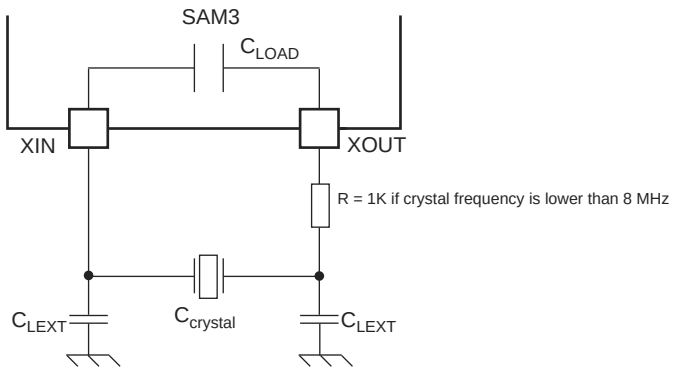
Writing a zero to this bit will disable the Pipe x, which forces the Pipe x state to inactive and resets the pipe x registers (UOTGHS_HSTPIPCFGx, UOTGHS_HSTPIPISRx, UOTGHS_HSTPIPIMRx) but not the pipe configuration (UOTGHS_HSTPIPCFGx.ALLOC, UOTGHS_HSTPIPCFGx.PBK, UOTGHS_HSTPIPCFGx.PSIZE).

45.4.6 3 to 20 MHz Crystal Oscillator Characteristics

Table 45-20. 3 to 20 MHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	Operating Frequency	Normal mode with crystal	3	16	20	MHz
$f_{OSC(bypass)}$	Operating Frequency In Bypass Mode	External Clock on XIN			50	MHz
$V_{rip(VDDPLL)}$	Supply Ripple Voltage (on VDDPLL)	RMS value, 10 kHz to 10 MHz			30	mV
	Duty Cycle		40	50	60	%
t_{START}	Startup Time	3 MHz, C_{SHUNT} = 3 pF 8 MHz, C_{SHUNT} = 7 pF 12 to 16 MHz, C_{SHUNT} = 7 pF 20 MHz, C_{SHUNT} = 7 pF			14.5 4 1.4 1	ms
I_{DDON}	Current Consumption	3 MHz 8 MHz 12 to 16 MHz 20 MHz		150 150 300 400	250 250 450 550	µA
$I_{DD(standby)}$	Standby Current Consumption	Standby mode @ 3.6V			5	nA
P_{ON}	Drive Level	3 MHz 8 MHz 12 MHz, 16 MHz, 20 MHz			15 30 50	µW
R_f	Internal Resistor	Between XIN and XOUT		1		MΩ
C_{LEXT}	Maximum external capacitor on XIN and XOUT				10	pF
$C_{crystal}$	Allowed Crystal Capacitance Load	From crystal specification	12.5		17.5	pF
C_{LOAD}	Internal Equivalent Load Capacitance	Integrated Load Capacitance (XIN and XOUT in series)	7.5	9.5	12	pF

Figure 45-14. 3 to 20 MHz Crystal Oscillator Schematic



$C_{LEXT} = 2 \times (C_{crystal} - C_{LOAD} - C_{PCB})$

where:

C_{PCB} is the capacitance of the printed circuit board (PCB) track layout from the crystal to the SAM3 pin.

Table 50-1. SAM3X/SAM3A Datasheet Rev. 11057C Revision History (Continued)

Issue Date	Comments
23-Mar-15	Section 7. “Memories” Inserted Section 7.1 “Product Mapping” (was previously section 8. “Product Mapping”) Updated Section 7.2.1 “Internal SRAM” Section 7.2.2 “Internal ROM”: changed “At any time, the ROM is mapped at address 0x0018 0000” to “At any time, the ROM is mapped at address 0x0010 0000” Updated Section 7.2.3.10 “GPNVM Bits” Section 7.2.4 “Boot Strategies”: deleted sentence “The GPNVM bit can be cleared or set respectively through the “Clear General-purpose NVM Bit” and “Set General-purpose NVM Bit” commands of the EEFC User Interface” (this information is already provided in Section 7.2.3.10 “GPNVM Bits”)
	Section 8. “System Controller” Removed Figure 10-1. “System Controller Block Diagram” (redundant with Figure 16-1 “Supply Controller Block Diagram”)
	Section 9. “Peripherals” Table 9-2 ”Multiplexing on PIO Controller A (PIOA)”: added footnotes providing information on selecting extra functions Table 9-3 ”Multiplexing on PIO Controller B (PIOB)”: deleted “See the notes” from the ‘Comments’ column for rows PB0–PB9; added footnotes providing information on selecting extra functions Table 9-4 ”Multiplexing on PIO Controller C (PIOC)”: added footnote providing information on selecting ERASE as extra function
	Section 11. “Debug and Test Features” Section 11.5.2 “Debug Architecture”: corrected “Cortex-M3 embeds four functional units” to “Cortex-M3 embeds five functional units”
	Section 20. “SAM3X/A Boot Program” Section 20.4.3 “USB Device Port”: - in first paragraph, replaced “from Windows 98SE to Windows XP” with “beginning with Windows 98SE” - updated link to www.usb.org - deleted sentence “Unauthorized use of assigned or unassigned USB Vendor ID Numbers and associated Product ID Numbers is strictly prohibited” Section 20.4.4 “In Application Programming (IAP) Feature”: replaced two instances of “MC_FSR register” with “EEFC_FSR”; changed “by reading the NMI vector in ROM (0x00800008)” to “by reading the NMI vector in ROM (0x00100008)”
	Section 23. “External Memory Bus” Table 23-3 ”EBI Pins and External Static Device Connections”: edited and reorganized column headers Section 23.8 “Implementation Examples”: reorganized contents
	Section 36. “Timer Counter (TC)” Throughout, replaced instances of “Master clock” or “MCK” d with “peripheral clock” Throughout, replaced instances of ‘quadrature decoder logic’ with ‘quadrature decoder’ or ‘QDEC’ Section 36.1 “Description”: rewrote text; in second paragraph, TIOA1 replaced with TIOB1; removed Table 37-1. “Timer Counter Clock Assignment” Section 36.2 “Embedded Characteristics”: - corrected number of TC channels from “Three” to “9” - changed “Two Multi-purpose Input/Output Signals” to “Two multi-purpose input/output signals acting as trigger event” - deleted “Two global registers that act on all TC channels” - changed “Configuration Registers can be write protected” to “Register Write Protection” Section 36.3 “Block Diagram”: inserted Table 36-1 ”Timer Counter Clock Assignment” (was previously in Section 36.1 “Description”) and updated table footnote