# Microchip Technology - AT91SAM7X128-CU-999 Datasheet

**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 | ARM7® |
| Core Size | 16/32-Bit |
| Speed | 55MHz |
| Connectivity | CANbus, Ethernet, I²C, SPI, SSC, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, POR, PWM, WDT |
| Number of I/O | 62 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 32K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.65V ~ 1.95V |
| Data Converters | A/D 8x10b |
| 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/microchip-technology/at91sam7x128-cu-999 |

- Individually Programmable Open-drain, Pull-up Resistor and Synchronous Output
- Thirteen Peripheral DMA Controller (PDC) Channels
- One USB 2.0 Full Speed (12 Mbits per second) Device Port
  - On-chip Transceiver, 1352-byte Configurable Integrated FIFOs
- One Ethernet MAC 10/100 base-T
  - Media Independent Interface (MII) or Reduced Media Independent Interface (RMII)
  - Integrated 28-byte FIFOs and Dedicated DMA Channels for Transmit and Receive
- One Part 2.0A and Part 2.0B Compliant CAN Controller
  - Eight Fully-programmable Message Object Mailboxes, 16-bit Time Stamp Counter
- One Synchronous Serial Controller (SSC)
  - Independent Clock and Frame Sync Signals for Each Receiver and Transmitter
  - I²S Analog Interface Support, Time Division Multiplex Support
  - High-speed Continuous Data Stream Capabilities with 32-bit Data Transfer
- Two Universal Synchronous/Asynchronous Receiver Transmitters (USART)
  - Individual Baud Rate Generator, IrDA® Infrared Modulation/Demodulation
  - Support for ISO7816 T0/T1 Smart Card, Hardware Handshaking, RS485 Support
  - Full Modem Line Support on USART1
- Two Master/Slave Serial Peripheral Interfaces (SPI)
  - 8- to 16-bit Programmable Data Length, Four External Peripheral Chip Selects
- One Three-channel 16-bit Timer/Counter (TC)
  - Three External Clock Inputs, Two Multi-purpose I/O Pins per Channel
  - Double PWM Generation, Capture/Waveform Mode, Up/Down Capability
- One Four-channel 16-bit Power Width Modulation Controller (PWMC)
- One Two-wire Interface (TWI)
  - Master Mode Support Only, All Two-wire Atmel EEPROMs and $I^2C$ Compatible Devices Supported
- One 8-channel 10-bit Analog-to-Digital Converter, Four Channels Multiplexed with Digital I/Os
- SAM-BA® Boot Assistance
  - Default Boot program
  - Interface with SAM-BA Graphic User Interface
- IEEE® 1149.1 JTAG Boundary Scan on All Digital Pins
- 5V-tolerant I/Os, Including Four High-current Drive I/O lines, Up to 16 mA Each
- Power Supplies
  - Embedded 1.8V Regulator, Drawing up to 100 mA for the Core and External Components
  - 3.3V VDDIO I/O Lines Power Supply, Independent 3.3V VDDFLASH Flash Power Supply
  - 1.8V VDDCORE Core Power Supply with Brownout Detector
- Fully Static Operation: Up to 55 MHz at 1.65V and 85°C Worst Case Conditions
- Available in 100-lead LQFP Green and 100-ball TFBGA Green Packages

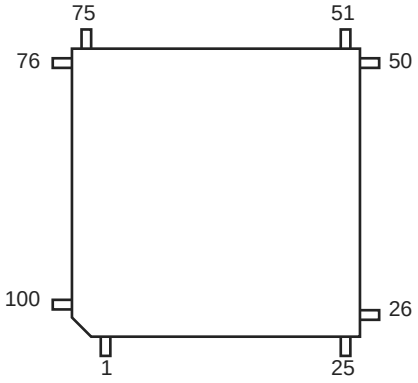
# 4. Package

The SAM7X512/256/128 is available in 100-lead LQFP Green and 100-ball TFBGA RoHS-compliant packages.

## 4.1 100-lead LQFP Package Outline

Figure 4-1 shows the orientation of the 100-lead LQFP package. A detailed mechanical description is given in the Mechanical Characteristics section.

**Figure 4-1. 100-lead LQFP Package Outline (Top View)**




## 4.2 100-lead LQFP Pinout

**Table 4-1. Pinout in 100-lead LQFP Package**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | ADVREF | 26 | PA18/PGMD6 | 51 | TDI | 76 | TDO |
| 2 | GND | 27 | PB9 | 52 | GND | 77 | JTAGSEL |
| 3 | AD4 | 28 | PB8 | 53 | PB16 | 78 | TMS |
| 4 | AD5 | 29 | PB14 | 54 | PB4 | 79 | TCK |
| 5 | AD6 | 30 | PB13 | 55 | PA23/PGMD11 | 80 | PA30 |
| 6 | AD7 | 31 | PB6 | 56 | PA24/PGMD12 | 81 | PA0/PGMEN0 |
| 7 | VDDOUT | 32 | GND | 57 | NRST | 82 | PA1/PGMEN1 |
| 8 | VDDIN | 33 | VDDIO | 58 | TST | 83 | GND |
| 9 | PB27/AD0 | 34 | PB5 | 59 | PA25/PGMD13 | 84 | VDDIO |
| 10 | PB28/AD1 | 35 | PB15 | 60 | PA26/PGMD14 | 85 | PA3 |
| 11 | PB29/AD2 | 36 | PB17 | 61 | VDDIO | 86 | PA2 |
| 12 | PB30/AD3 | 37 | VDDCORE | 62 | VDDCORE | 87 | VDDCORE |
| 13 | PA8/PGMM0 | 38 | PB7 | 63 | PB18 | 88 | PA4/PGMNCMD |
| 14 | PA9/PGMM1 | 39 | PB12 | 64 | PB19 | 89 | PA5/PGMRDY |
| 15 | VDDCORE | 40 | PB0 | 65 | PB20 | 90 | PA6/PGMNOE |
| 16 | GND | 41 | PB1 | 66 | PB21 | 91 | PA7/PGMNVALID |
| 17 | VDDIO | 42 | PB2 | 67 | PB22 | 92 | ERASE |
| 18 | PA10/PGMM2 | 43 | PB3 | 68 | GND | 93 | DDM |
| 19 | PA11/PGMM3 | 44 | PB10 | 69 | PB23 | 94 | DDP |
| 20 | PA12/PGMD0 | 45 | PB11 | 70 | PB24 | 95 | VDDFLASH |
| 21 | PA13/PGMD1 | 46 | PA19/PGMD7 | 71 | PB25 | 96 | GND |
| 22 | PA14/PGMD2 | 47 | PA20/PGMD8 | 72 | PB26 | 97 | XIN/PGMCK |
| 23 | PA15/PGMD3 | 48 | VDDIO | 73 | PA27/PGMD15 | 98 | XOUT |
| 24 | PA16/PGMD4 | 49 | PA21/PGMD9 | 74 | PA28 | 99 | PLLRC |
| 25 | PA17/PGMD5 | 50 | PA22/PGMD10 | 75 | PA29 | 100 | VDDPLL |

# 5. Power Considerations

## 5.1 Power Supplies

The SAM7X512/256/128 has six types of power supply pins and integrates a voltage regulator, allowing the device to be supplied with only one voltage. The six power supply pin types are:

- VDDIN pin. It powers the voltage regulator and the ADC; voltage ranges from 3.0V to 3.6V, 3.3V nominal. In order to decrease current consumption, if the voltage regulator and the ADC are not used, VDDIN, ADVREF, AD4, AD5, AD6 and AD7 should be connected to GND. In this case, VDDOUT should be left unconnected.
- VDDOUT pin. It is the output of the 1.8V voltage regulator.
- VDDIO pin. It powers the I/O lines; voltage ranges from 3.0V to 3.6V, 3.3V nominal.
- VDDFLASH pin. It powers the USB transceivers and a part of the Flash and is required for the Flash to operate correctly; voltage ranges from 3.0V to 3.6V, 3.3V nominal.
- VDDCORE pins. They power the logic of the device; voltage ranges from 1.65V to 1.95V, 1.8V typical. It can be connected to the VDDOUT pin with decoupling capacitor. VDDCORE is required for the device, including its embedded Flash, to operate correctly.
- VDDPLL pin. It powers the oscillator and the PLL. It can be connected directly to the VDDOUT pin.

No separate ground pins are provided for the different power supplies. Only GND pins are provided and should be connected as shortly as possible to the system ground plane.

## 5.2 Power Consumption

The SAM7X512/256/128 has a static current of less than 60 µA on VDDCORE at 25°C, including the RC oscillator, the voltage regulator and the power-on reset when the brownout detector is deactivated. Activating the brownout detector adds 28 µA static current.

The dynamic power consumption on VDDCORE is less than 90 mA at full speed when running out of the Flash. Under the same conditions, the power consumption on VDDFLASH does not exceed 10 mA.

## 5.3 Voltage Regulator

The SAM7X512/256/128 embeds a voltage regulator that is managed by the System Controller.

In Normal Mode, the voltage regulator consumes less than 100 µA static current and draws 100 mA of output current.

The voltage regulator also has a Low-power Mode. In this mode, it consumes less than 25 µA static current and draws 1 mA of output current.

Adequate output supply decoupling is mandatory for VDDOUT to reduce ripple and avoid oscillations. The best way to achieve this is to use two capacitors in parallel: one external 470 pF (or 1 nF) NPO capacitor should be connected between VDDOUT and GND as close to the chip as possible. One external 2.2 µF (or 3.3 µF) X7R capacitor should be connected between VDDOUT and GND.

Adequate input supply decoupling is mandatory for VDDIN in order to improve startup stability and reduce source voltage drop. The input decoupling capacitor should be placed close to the chip. For example, two capacitors can be used in parallel: 100 nF NPO and 4.7 µF X7R.

## 5.4 Typical Powering Schematics

The SAM7X512/256/128 supports a 3.3V single supply mode. The internal regulator input connected to the 3.3V source and its output feeds VDDCORE and the VDDPLL. Figure 5-1 shows the power schematics to be used for USB bus-powered systems.

## 16.3 Functional Description

The Watchdog Timer can be used to prevent system lock-up if the software becomes trapped in a deadlock. It is supplied with VDDCORE. It restarts with initial values on processor reset.

The Watchdog is built around a 12-bit down counter, which is loaded with the value defined in the field WV of the Mode Register (WDT_MR). The Watchdog Timer uses the Slow Clock divided by 128 to establish the maximum Watchdog period to be 16 seconds (with a typical Slow Clock of 32.768 kHz).

After a Processor Reset, the value of WV is 0xFFF, corresponding to the maximum value of the counter with the external reset generation enabled (field WDRSTEN at 1 after a Backup Reset). This means that a default Watchdog is running at reset, i.e., at power-up. The user must either disable it (by setting the WDDIS bit in WDT_MR) if he does not expect to use it or must reprogram it to meet the maximum Watchdog period the application requires.

The Watchdog Mode Register (WDT_MR) can be written only once. Only a processor reset resets it. Writing the WDT_MR register reloads the timer with the newly programmed mode parameters.

If the watchdog is restarted by writing into the WDT_CR register, the WDT_MR register must not be programmed during a period of time of 3 slow clock period following the WDT_CR write access. In any case, programming a new value in WDT_MR automatically initiates a restart instruction.

In normal operation, the user reloads the Watchdog at regular intervals before the timer underflow occurs, by writing the Control Register (WDT_CR) with the bit WDRSTT to 1. The Watchdog counter is then immediately reloaded from WDT_MR and restarted, and the Slow Clock 128 divider is reset and restarted. The WDT_CR register is write-protected. As a result, writing WDT_CR without the correct hard-coded key has no effect. If an underflow does occur, the “wdt_fault” signal to the Reset Controller is asserted if the bit WDRSTEN is set in the Mode Register (WDT_MR). Moreover, the bit WDUNF is set in the Watchdog Status Register (WDT_SR).

To prevent a software deadlock that continuously triggers the Watchdog, the reload of the Watchdog must occur while the Watchdog counter is within a window between 0 and WDD, WDD is defined in the WatchDog Mode Register WDT_MR.

Any attempt to restart the Watchdog while the Watchdog counter is between WDV and WDD results in a Watchdog error, even if the Watchdog is disabled. The bit WDERR is updated in the WDT_SR and the “wdt_fault” signal to the Reset Controller is asserted.

Note that this feature can be disabled by programming a WDD value greater than or equal to the WDV value. In such a configuration, restarting the Watchdog Timer is permitted in the whole range [0; WDV] and does not generate an error. This is the default configuration on reset (the WDD and WDV values are equal).

The status bits WDUNF (Watchdog Underflow) and WDERR (Watchdog Error) trigger an interrupt, provided the bit WDFIEN is set in the mode register. The signal “wdt_fault” to the reset controller causes a Watchdog reset if the WDRSTEN bit is set as already explained in the reset controller programmer Datasheet. In that case, the processor and the Watchdog Timer are reset, and the WDERR and WDUNF flags are reset.

If a reset is generated or if WDT_SR is read, the status bits are reset, the interrupt is cleared, and the “wdt_fault” signal to the reset controller is deasserted.

Writing the WDT_MR reloads and restarts the down counter.

While the processor is in debug state or in idle mode, the counter may be stopped depending on the value programmed for the bits WDIDLEHLT and WDDBGHLT in the WDT_MR.

#### 20.2.5.9 Get Version Command

The **Get Version** (GVE) command retrieves the version of the FFPI interface.

**Table 20-17. Get Version Command**

| Step | Handshake Sequence | MODE[3:0] | DATA[15:0] |
|---|---|---|---|
| 1 | Write handshaking | CMDE | GVE |
| 2 | Write handshaking | DATA | Version |

# 22. Peripheral DMA Controller (PDC)

## 22.1 Overview

The Peripheral DMA Controller (PDC) transfers data between on-chip serial peripherals such as the UART, USART, SSC, SPI, MCI and the on- and off-chip memories. Using the Peripheral DMA Controller avoids processor intervention and removes the processor interrupt-handling overhead. This significantly reduces the number of clock cycles required for a data transfer and, as a result, improves the performance of the microcontroller and makes it more power efficient.

The PDC channels are implemented in pairs, each pair being dedicated to a particular peripheral. One channel in the pair is dedicated to the receiving channel and one to the transmitting channel of each UART, USART, SSC and SPI.

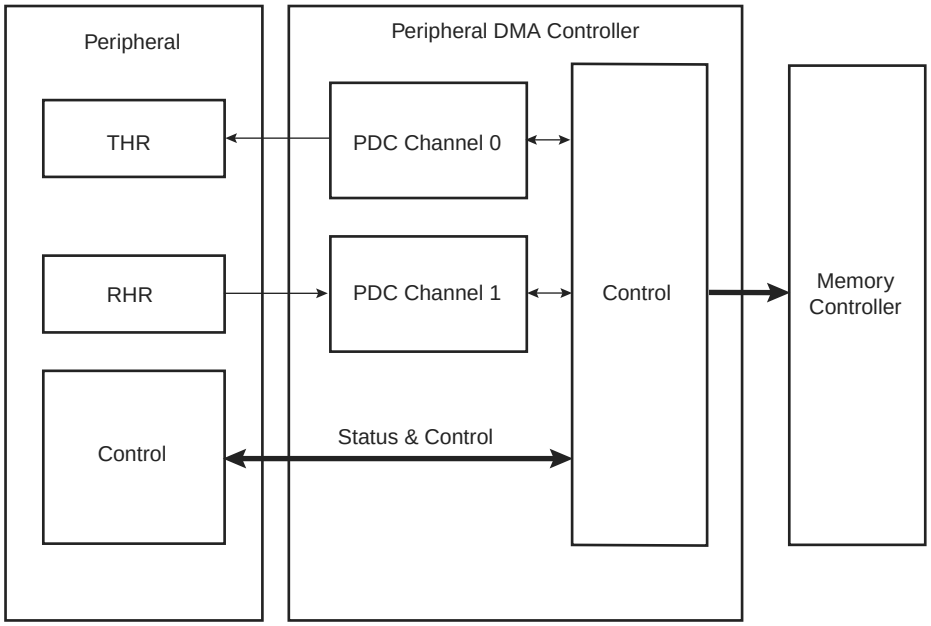
The user interface of a PDC channel is integrated in the memory space of each peripheral. It contains:

- A 32-bit memory pointer register
- A 16-bit transfer count register
- A 32-bit register for next memory pointer
- A 16-bit register for next transfer count

The peripheral triggers PDC transfers using transmit and receive signals. When the programmed data is transferred, an end of transfer interrupt is generated by the corresponding peripheral.

## 22.2 Block Diagram

**Figure 22-1. Block Diagram**




## 22.3 Functional Description

### 22.3.1 Configuration

The PDC channels user interface enables the user to configure and control the data transfers for each channel. The user interface of a PDC channel is integrated into the user interface of the peripheral (offset 0x100), which it is related to.

Per peripheral, it contains four 32-bit Pointer Registers (RPR, RNPR, TPR, and TNPR) and four 16-bit Counter Registers (RCR, RNCR, TCR, and TNCR).

## 25.9 Power Management Controller (PMC) User Interface

**Table 25-2. Register Mapping**

| Offset | Register | Name | Access | Reset |
|---|---|---|---|---|
| 0x0000 | System Clock Enable Register | PMC_SCER | Write-only | – |
| 0x0004 | System Clock Disable Register | PMC_SCDR | Write-only | – |
| 0x0008 | System Clock Status Register | PMC _SCSR | Read-only | 0x01 |
| 0x000C | Reserved | – | – | – |
| 0x0010 | Peripheral Clock Enable Register | PMC _PCER | Write-only | – |
| 0x0014 | Peripheral Clock Disable Register | PMC_PCDR | Write-only | – |
| 0x0018 | Peripheral Clock Status Register | PMC_PCSR | Read-only | 0x0 |
| 0x001C | Reserved | – | – | – |
| 0x0020 | Main Oscillator Register | CKGR_MOR | Read-write | 0x0 |
| 0x0024 | Main Clock Frequency Register | CKGR_MCFR | Read-only | 0x0 |
| 0x0028 | Reserved | – | – | – |
| 0x002C | PLL Register | CKGR_PLLR | Read-write | 0x3F00 |
| 0x0030 | Master Clock Register | PMC_MCKR | Read-write | 0x0 |
| 0x0038 | Reserved | – | – | – |
| 0x003C | Reserved | – | – | – |
| 0x0040 | Programmable Clock 0 Register | PMC_PCK0 | Read-write | 0x0 |
| 0x0044 | Programmable Clock 1 Register | PMC_PCK1 | Read-write | 0x0 |
| ... | ... | ... | ... | ... |
| 0x0060 | Interrupt Enable Register | PMC_IER | Write-only | -- |
| 0x0064 | Interrupt Disable Register | PMC_IDR | Write-only | -- |
| 0x0068 | Status Register | PMC_SR | Read-only | 0x08 |
| 0x006C | Interrupt Mask Register | PMC_IMR | Read-only | 0x0 |
| 0x0070 - 0x007C | Reserved | – | – | – |

### 26.5.3 Debug Unit Interrupt Enable Register

**Name:** DBGU_IER

**Access Type:** Write-only

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| COMMRX | COMMTX | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | RXBUFF | TXBUFE | – | TXEMPTY | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PARE | FRAME | OVRE | ENDTX | ENDRX | – | TXRDY | RXRDY |

- **RXRDY: Enable RXRDY Interrupt**
- **TXRDY: Enable TXRDY Interrupt**
- **ENDRX: Enable End of Receive Transfer Interrupt**
- **ENDTX: Enable End of Transmit Interrupt**
- **OVRE: Enable Overrun Error Interrupt**
- **FRAME: Enable Framing Error Interrupt**
- **PARE: Enable Parity Error Interrupt**
- **TXEMPTY: Enable TXEMPTY Interrupt**
- **TXBUFE: Enable Buffer Empty Interrupt**
- **RXBUFF: Enable Buffer Full Interrupt**
- **COMMTX: Enable COMMTX (from ARM) Interrupt**
- **COMMRX: Enable COMMRX (from ARM) Interrupt**

0 = No effect.

1 = Enables the corresponding interrupt.

### 30.7.10 USART Receiver Time-out Register

**Name:** US_RTOR

**Access Type:** 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 |
|---|---|---|---|---|---|---|---|
| TO | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TO | | | | | | | |

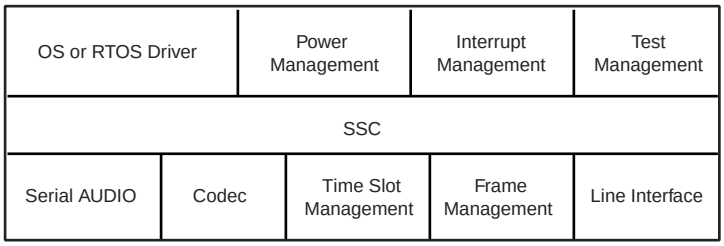
- **TO: Time-out Value**

0: The Receiver Time-out is disabled.

1 - 65535: The Receiver Time-out is enabled and the Time-out delay is TO x Bit Period.

## 31.3 Application Block Diagram

**Figure 31-2. Application Block Diagram**




## 31.4 Pin Name List

**Table 31-1. 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 |

## 31.5 Product Dependencies

### 31.5.1 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.

### 31.5.2 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.

### 31.5.3 Interrupt

The SSC interface has an interrupt line connected to the Advanced Interrupt Controller (AIC). Handling interrupts requires programming the AIC 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.

## 31.6 Functional Description

This chapter contains the functional description of the following: SSC Functional Block, Clock Management, Data format, Start, Transmitter, Receiver and Frame Sync.

**Figure 32-2. Clock Chaining Selection**

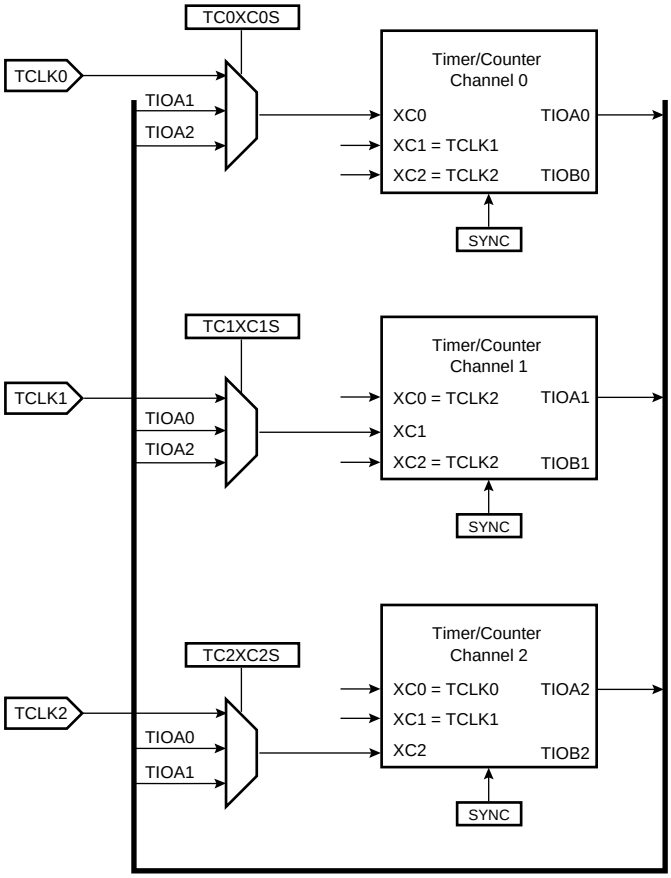
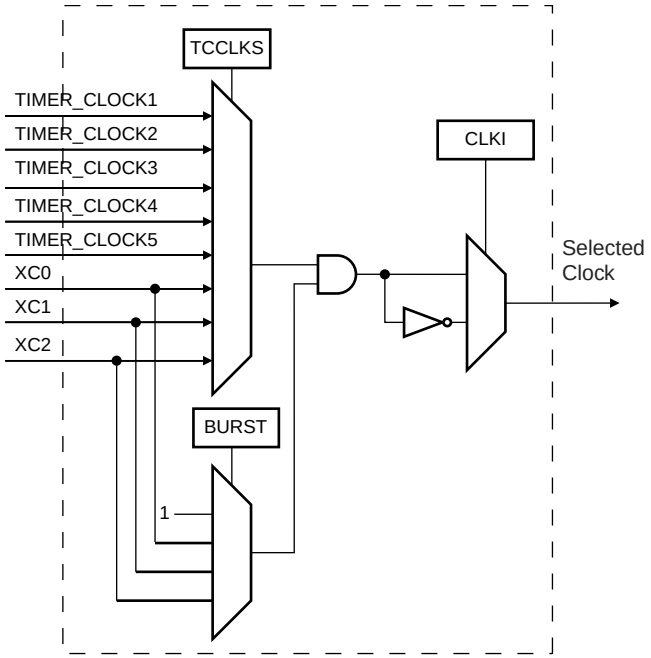



**Figure 32-3. Clock Selection**



# 36. Controller Area Network (CAN)

## 36.1 Overview

The CAN controller provides all the features required to implement the serial communication protocol CAN defined by Robert Bosch GmbH, the CAN specification as referred to by ISO/11898A (2.0 Part A and 2.0 Part B) for high speeds and ISO/11519-2 for low speeds. The CAN Controller is able to handle all types of frames (Data, Remote, Error and Overload) and achieves a bitrate of 1 Mbit/sec.

CAN controller accesses are made through configuration registers. 8 independent message objects (mailboxes) are implemented.

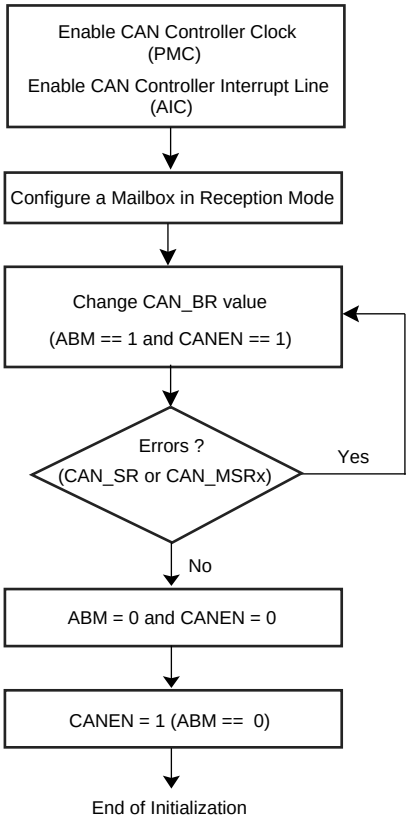
Any mailbox can be programmed as a reception buffer block (even non-consecutive buffers). For the reception of defined messages, one or several message objects can be masked without participating in the buffer feature. An interrupt is generated when the buffer is full. According to the mailbox configuration, the first message received can be locked in the CAN controller registers until the application acknowledges it, or this message can be discarded by new received messages.

Any mailbox can be programmed for transmission. Several transmission mailboxes can be enabled in the same time. A priority can be defined for each mailbox independently.

An internal 16-bit timer is used to stamp each received and sent message. This timer starts counting as soon as the CAN controller is enabled. This counter can be reset by the application or automatically after a reception in the last mailbox in Time Triggered Mode.

The CAN controller offers optimized features to support the Time Triggered Communication (TTC) protocol.

**Figure 36-10. Possible Initialization Procedure**




### 36.7.2 CAN Controller Interrupt Handling

There are two different types of interrupts. One type of interrupt is a message-object related interrupt, the other is a system interrupt that handles errors or system-related interrupt sources.

All interrupt sources can be masked by writing the corresponding field in the CAN_IDR register. They can be unmasked by writing to the CAN_IER register. After a power-up reset, all interrupt sources are disabled (masked). The current mask status can be checked by reading the CAN_IMR register.

The CAN_SR register gives all interrupt source states.

The following events may initiate one of the two interrupts:

- Message object interrupt
  - Data registers in the mailbox object are available to the application. In Receive Mode, a new message was received. In Transmit Mode, a message was transmitted successfully.
  - A sent transmission was aborted.
- System interrupts
  - Bus off interrupt: The CAN module enters the bus off state.
  - Error passive interrupt: The CAN module enters Error Passive Mode.
  - Error Active Mode: The CAN module is neither in Error Passive Mode nor in Bus Off mode.
  - Warn Limit interrupt: The CAN module is in Error-active Mode, but at least one of its error counter value exceeds 96.
  - Wake-up interrupt: This interrupt is generated after a wake-up and a bus synchronization.
  - Sleep interrupt: This interrupt is generated after a Low-power Mode enable once all pending messages in transmission have been sent.
  - Internal timer counter overflow interrupt: This interrupt is generated when the internal timer rolls over.

### 36.8.3 CAN Interrupt Disable Register

**Name:** CAN_IDR

**Access Type:** Write-only

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | BERR | FERR | AERR | SERR | CERR |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TSTP | TOVF | WAKEUP | SLEEP | BOFF | ERRP | WARN | ERRA |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MB7 | MB6 | MB5 | MB4 | MB3 | MB2 | MB1 | MB0 |

- **MBx: Mailbox x Interrupt Disable**

0 = No effect.

1 = Disable Mailbox x interrupt.

- **ERRA: Error Active Mode Interrupt Disable**

0 = No effect.

1 = Disable ERRA interrupt.

- **WARN: Warning Limit Interrupt Disable**

0 = No effect.

1 = Disable WARN interrupt.

- **ERRP: Error Passive Mode Interrupt Disable**

0 = No effect.

1 = Disable ERRP interrupt.

- **BOFF: Bus Off Mode Interrupt Disable**

0 = No effect.

1 = Disable BOFF interrupt.

- **SLEEP: Sleep Interrupt Disable**

0 = No effect.

1 = Disable SLEEP interrupt.

- **WAKEUP: Wakeup Interrupt Disable**

0 = No effect.

1 = Disable WAKEUP interrupt.

- **TOVF: Timer Overflow Interrupt**

0 = No effect.

1 = Disable TOVF interrupt.

### 36.8.12 CAN Message Mode Register

**Name:** CAN_MMRx

**Access Type:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | MOT | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | PRIOR | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MTIMEMARK15 | MTIMEMARK14 | MTIMEMARK13 | MTIMEMARK12 | MTIMEMARK11 | MTIMEMARK10 | MTIMEMARK9 | MTIMEMARK8 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MTIMEMARK7 | MTIMEMARK6 | MTIMEMARK5 | MTIMEMARK4 | MTIMEMARK3 | MTIMEMARK2 | MTIMEMARK1 | MTIMEMARK0 |

- **MTIMEMARK: Mailbox Timemark**

This field is active in Time Triggered Mode. Transmit operations are allowed when the internal timer counter reaches the Mailbox Timemark. See “Transmitting within a Time Window” on page 504.

In Timestamp Mode, MTIMEMARK is set to 0.

- **PRIOR: Mailbox Priority**

This field has no effect in receive and receive with overwrite modes. In these modes, the mailbox with the lowest number is serviced first.

When several mailboxes try to transmit a message at the same time, the mailbox with the highest priority is serviced first. If several mailboxes have the same priority, the mailbox with the lowest number is serviced first (i.e., MBx0 is serviced before MBx 15 if they have the same priority).

- **MOT: Mailbox Object Type**

This field allows the user to define the type of the mailbox. All mailboxes are independently configurable. Five different types are possible for each mailbox:

| MOT | | | Mailbox Object Type |
|---|---|---|---|
| 0 | 0 | 0 | Mailbox is disabled. This prevents receiving or transmitting any messages with this mailbox. |
| 0 | 0 | 1 | Reception Mailbox. Mailbox is configured for reception. If a message is received while the mailbox data register is full, it is discarded. |
| 0 | 1 | 0 | Reception mailbox with overwrite. Mailbox is configured for reception. If a message is received while the mailbox is full, it overwrites the previous message. |
| 0 | 1 | 1 | Transmit mailbox. Mailbox is configured for transmission. |
| 1 | 0 | 0 | Consumer Mailbox. Mailbox is configured in reception but behaves as a Transmit Mailbox, i.e., it sends a remote frame and waits for an answer. |
| 1 | 0 | 1 | Producer Mailbox. Mailbox is configured in transmission but also behaves like a reception mailbox, i.e., it waits to receive a Remote Frame before sending its contents. |
| 1 | 1 | X | Reserved |

### 36.8.18 CAN Message Data High Register

**Name:** CAN_MDHx

**Access Type:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MDH | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| MDH | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MDH | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MDH | | | | | | | |

- **MDH: Message Data High Value**

When MRDY field is set in the CAN_MSRx register, the upper 32 bits of a received message are read or written by the software application. Otherwise, the MDH value is locked by the CAN controller to send/receive a new message.

In Receive with overwrite, the CAN controller may modify MDH value while the software application reads MDH and MDL registers. To check that MDH and MDL do not belong to different messages, the application has to check the MMI field in the CAN_MSRx register. In this mode, the software application must re-read CAN_MDH and CAN_MDL, while the MMI bit in the CAN_MSRx register is set.

Bytes are received/sent on the bus in the following order:

1. CAN_MDL[7:0]
2. CAN_MDL[15:8]
3. CAN_MDL[23:16]
4. CAN_MDL[31:24]
5. CAN_MDH[7:0]
6. CAN_MDH[15:8]
7. CAN_MDH[23:16]
8. CAN_MDH[31:24]

### 37.5.6 Transmit Buffer Queue Pointer Register

**Register Name:** EMAC_TBQP

**Access Type:** Read-write

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | – | – |

This register points to the entry in the transmit buffer queue (descriptor list) currently being used. It is written with the start location of the transmit buffer descriptor list. The lower order bits increment as buffers are used up and wrap to their original values after either 1024 buffers or when the wrap bit of the entry is set. This register can only be written when bit 3 in the transmit status register is low.

As transmit buffer reads consist of bursts of two words, it is recommended that bit 2 is always written with zero to prevent a burst crossing a 1K boundary, in violation of section 3.6 of the AMBA specification.

- **ADDR: Transmit buffer queue pointer address**

Written with the address of the start of the transmit queue, reads as a pointer to the first buffer of the frame being transmitted or about to be transmitted.

# 40. AT91SAM7X Ordering Information

**Table 40-1. Ordering Information**

| MRL A Ordering Code | MRL B Ordering Code | MRL C Ordering Code | Package | Package Type | Temperature Operating Range |
|---|---|---|---|---|---|
| AT91SAM7X512-AU | AT91SAM7X512B-AU | – | LQFP 100 | Green | Industrial (-40°C to 85°C) |
| – | AT91SAM7X512B-AUR | | LQFP 100(1) | | |
| AT91SAM7X512-CU | AT91SAM7X512B-CU | | TFBGA 100 | | |
| AT91SAM7X256-AU | AT91SAM7X256B-AU | AT91SAM7X256C-AU | LQFP 100 | Green | Industrial (-40°C to 85°C) |
| AT91SAM7X256-CU | AT91SAM7X256B-CU | AT91SAM7X256C-CU | TFBGA 100 | | |
| AT91SAM7X128-AU | AT91SAM7X128B-AU | AT91SAM7X128C-AU | LQFP 100 | Green | Industrial (-40°C to 85°C) |
| AT91SAM7X128-CU | AT91SAM7X128B-CU | AT91SAM7X128C-CU | TFBGA 100 | | |

Note: 1. Packing type is tape and reel.

| Version 6120I | Comments | Change Request Ref. |
|---|---|---|
| | **Ordering Information:**<br>Table 40-1, “Ordering Information”<br>The following ordering codes added to the table for MRL C.<br>AT91SAM7X256C-AU<br>AT91SAM7X256C-CU<br>AT91SAM7X128C-AU<br>AT91SAM7X128C-CU | 7371 |
| | **Overview:**<br>Section 9.5 ”Debug Unit”<br>“Chip ID Registers” , Chip IDs updated with reference to MRL A, B or C. | rfo |
| | **Product Series Naming Convention:**<br>Except for part ordering and library references, AT91 prefix dropped from most nomenclature.<br>AT91SAM7X becomes SAM7X. | rfo |
| | **Errata:**<br>Table 41-1, “Errata Summary Table”, added.<br>Section 41.7 ”AT91SAM7X256/128 Errata - Rev. C Parts”, added.<br>Section 41.3 ”AT91SAM7X256/128 Errata - Rev. A Parts”, added note specific to Rev A chip IDs.<br>Section 41.4 ”AT91SAM7X512 Errata - Rev. A Parts”, added note specific to Rev A chip ID.<br>Section 41.5 ”AT91SAM7X256/128 Errata - Rev. B Parts”, added note specific to Rev B chip IDs.<br>Section 41.4.3.1 ”EFC: Embedded Flash Access Time” Problem Fix/Workaround, revised.<br>Section 41.3.10.3 ”USART: RXBRK Flag Error in Asynchronous Mode”, Revised.<br>Section 41.4.11.3 ”USART: RXBRK Flag Error in Asynchronous Mode”, Revised.<br>Section 41.5.10.3 ”USART: RXBRK Flag Error in Asynchronous Mode”, Revised. | 7371<br>rfo<br>6624 |
| | **Electrical Characteristics:**<br>Table 38-2, “DC Characteristics” $V_{OL}$ and $V_{OH}$ rows revised (removed 1.65 t0 1.95V $V_{VDDIO}$ values).<br>Table 38-9, Table 38-10, Table 38-10, fixed typos in Units column: µW or W => µΩ or Ω. | 7211<br>6484 |
| | **EFC:**<br>Section 19.2.4.4 ”General-purpose NVM Bits”, updated the last paragraph.<br>Figure 19-6,”Example of Partial Page Programming” Text added befow figure | 6233<br>6825 |
| | **Debug and Test Features:**<br>“MANUFACTURER IDENTITY[11:1]” , AT91SAM7X128: JTAG ID Code value is 05B1_603F. | 7354 |